Novel Frameworks for Auctions and Optimization

by

Zeyuan Allen-Zhu

B.S. in Mathematics and Physics, Tsinghua University (2010)
S.M. in Electrical Engineering and Computer Science, MIT (2012)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2015

Author .....
Department of Electrical Engineering and Computer Science
August 17, 2015

Certified by .....
Jonathan A. Kelner
Associate Professor of Applied Mathematics
Thesis Supervisor

Certified by .....
Silvio Micali
Ford Professor of Engineering
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Theses

Novel Frameworks for Auctions and Optimization

by

Zeyuan Allen-Zhu

Submitted to the Department of Electrical Engineering and Computer Science
on August 17, 2015, in partial fulfillment of the
requirements for the degree of
Doctor of Science

Abstract

This thesis contains two parts.

Part I introduces novel frameworks for modeling uncertainty in auctions. This enables us to provide robust analysis to alternative specifications of preferences and information structures in Vickrey and VCG auctions.

Part II introduces novel frameworks for understanding first-order methods in optimization. This enables us to (1) break 20-year barriers on the running time used for solving positive linear programs, (2) reduce the complexity for solving positive semidefinite programs, and (3) strengthen the theory of matrix multiplicative weight updates and improve the theory of linear-sized spectral sparsification.

Thesis Supervisor: Jonathan A. Kelner
Title: Associate Professor of Applied Mathematics

Thesis Supervisor: Silvio Micali
Title: Ford Professor of Engineering

To my mum, Xiaoli Xu

Acknowledgments

暮色苍茫看劲松，乱云飞渡仍从容。天生一个仙人洞，无限风光在险峰。

— *Zedong Mao*

I would like to thank Professor Silvio Micali for his careful and close supervision in the past five academic years. Not only his inspiring and idiosyncratic talk in mechanism design inspired me to enter the field of game theory, it was also his very first choice that brings me to the theory family of MIT CSAIL. I cannot appreciate more on this. It was also my extreme fortune to be supervised by Professor Jonathan Kelner who nurtured me, supported me, and inspired me with his knowledgeable experiences and insightful comments to nearly all the research areas surrounding computer science. It was such a wonderful experience and a unique memory to be co-supervised by both Silvio and Jon during my doctoral studies, and there are lots of things that I can continue to learn from them regarding how they have become such world-class researchers.

I would like to thank Professors Shafi Goldwasser and Nir Shavit for their special and precious encouragements during many periods of my graduate program.

I would like to thank Alessandro Chiesa who introduced me all the secrets about MIT, so that I can integrate into this big family without difficulty.

I would like to thank Lorenzo Orecchia who discussed with me all of the crazy ideas so that I can bravely start a new research direction when there are only two years left in my graduate program.

Beyond this thesis, I am fortunate to have collaborated extensively with, in alphabetical order, Rati Gelashvili, Silvio Lattanzi, Zhenyu Liao, Vahab Mirrokni, Sasa Misailovic, Ilya Razenshteyn, Martin Rinard, Nir Shavit, Christian Sommer and Yang Yuan, and more or less with many others at MIT. I am also grateful for the supports that I have received beyond research, including that from our instructors, our students and our secretaries in the CSAIL Theory Group.

Last but not least, I never forget to thank my dearest mother for her unceasing support and unselfish dedication to my family, my education, throughout the past 27 years. For all of the above, and the unnamed, please allow me to express my deepest gratitude, now and always.

Financial Acknowledgements. This thesis and my graduate study at MIT are fully supported by

- 9 months of Greater China Fellowship,
- 3 months Big George Ventures Fund from Ray Sidney,
- 6 months of Simons Award from the Simons Foundation,
- 3 months of Bridge Fund from the MIT EECS department,
- 4.5 months of Teaching Assistantship from Professor David Karger,

- 9 months of Research Assistantship from Professor Jonathan Kelner, and
- 25.5 months of Research Assistantship from Professor Silvio Micali.

I would like to sincerely thank all of the supporters above that make my study at MIT possible. Besides, I would also like to thank the Akamai Foundation and the Simons Foundation that put together $22,000 travel and equipment money that has made my collaborations outside MIT more than easy and enjoyable.

Other Acknowledgements. I would like to thank a United Airline operated flight and the Simons Institute at Berkeley at which places the results of Chapter 8 of this thesis was obtained.

I would like to thank the MIT office 32-G804 and its owner at the time, Cong Yan, without its quite location and the owner's hospitality the main result of Chapter 6 of this thesis cannot be produced.

Part I of this thesis, namely, Chapters 1, 2 and 3 were obtained at my MIT CSAIL office 32-G636 with the presence of my lovely officemates. The rest of Part II, namely, Chapters 4, 5 and 7 were obtained at my MIT Math desk E17-301A as well as a bright and amazing common study room of the Ashdown House. I would like to thank MIT that provides awesome research environments like these.

Contents

Part I

Novel Frameworks for Auctions

Chapter 1

Knightian Analysis of the Vickrey Mechanism

This chapter is based on the result published in [44] as well as the online ArXiv: `http://arxiv.org/abs/1403.6413`.

We analyze the Vickrey mechanism for auctions of multiple identical goods when the players have both Knightian uncertainty over their own valuations and incomplete preferences. In this model, the Vickrey mechanism is no longer dominant-strategy, and we prove that all dominant-strategy mechanisms are inadequate. However, we also prove that, in undominated strategies, the social welfare produced by the Vickrey mechanism in the worst case is not only very good, but also essentially optimal.

1.1 Introduction

We prove that the classical Vickrey mechanism guarantees good social welfare even when the players have *extremely* limited knowledge about themselves.

Recall that the Vickrey mechanism efficiently allocates multiple identical goods by ensuring that it is a dominant strategy for each player i to report his true valuation, θ_i^*. In real life, however, a player i may be uncertain about θ_i^*, as it may depend on variables that are not directly observable by him. A simple way to capture a player i's uncertainty about his own valuation is the 'single-distribution' model, where i does not know θ_i^*, but only the true distribution from which θ_i^* has been drawn. We instead investigate a more general form of self uncertainty.

Knightian Valuation Uncertainty. In our model, the only information that a player i has about θ_i^* (and more generally about the true valuation profile, θ^*) consists of a *set of distributions*, from one of which θ_i^* has been drawn. We refer to this model as *Knightian valuation uncertainty* or *the Knightian valuation model*, as it is a special case of the uncertainty model envisaged by Frank H. Knight almost a century ago [91], and later formalized by Truman F. Bewley [30].

Knightian valuation uncertainty may arise from conflicting expert opinions. Consider a multi-unit auction of a novel good. Unable to evaluate his valuation, a player i hires multiple (properly incentivized) independent experts to figure it out, trusting that at least one of them will be right. If each of them reports a different distribution for θ_i^*, either because time was limited or because some of the experts made errors, then i is ultimately faced with a set of distributions, from one of which θ_i^* has been drawn.

Incomplete Preferences. One may of course assume that players with Knightian valuation uncertainty have complete preferences, and in particular maxmin preferences, as defined by Gilboa and Schmeidler [70]. Such preferences are certainly defendable, however, quoting Bewley [30], they "do not lead to the sorts of economic behavior which make Knightian behavior interesting."

In our paper, players have *incomplete preferences*. A player i, only knowing that his true valuation has been selected from one of multiple distributions, prefers an outcome ω to another outcome ω' if and only if his expected utility for ω is higher than or equal to his expected utility for ω' with respect to all such distributions (and strictly greater for at least some of them). As a consequence, some outcomes or some strategies may be incomparable to him.

Finally, we do not assume that a player with incomparable strategies chooses a 'reference strategy'. That is, we do not rely on the *inertia* assumption of Bewley [30]. However, we assume that the players are risk-neutral.

Findings. In the Knightian valuation model, the Vickrey mechanism is no longer dominant-strategy, but multi-unit dominant-strategy mechanisms still exist: for instance, the 'degenerate' mechanism, which assigns all copies the good to a random player. Our Theorem 1.6 shows that all dominant-strategy mechanisms, as well as all ex-post Nash mechanisms, whether deterministic or randomized, must essentially be degenerate. That is, we provide natural conditions under which the allocations of such mechanisms are unresponsive to each player's action and thus cannot be efficient. Importantly, Theorem 1.6 applies also to mechanisms that allow a player to report a set of valuation distributions rather than a single valuation.

Since dominant-strategy mechanisms cannot achieve even an approximately efficient outcome in our model, it is natural to ask what social-welfare performance can be guaranteed in undominated strategies. After all, one may be quite confident that a player will not choose a strategy outside his undominated set.

Our Theorem 1.8 characterizes the set of undominated strategies of a player with Knightian valuation uncertainty in the Vickrey mechanism. A simple corollary of this characterization, Corollary 1.10, guarantees that, in undominated strategies, the social-welfare performance of the Vickrey mechanism is good even in the worst case.

This guarantee, of course, does not exclude that a different mechanism may perform even better. However, our Theorem 1.14 shows that the worst-case per-

formance of the Vickrey mechanism is, *de facto*, asymptotically optimal among *all* undominated-strategy mechanisms, probabilistic or not, no matter what their strategy spaces may be. That is, as the number of players grows, no mechanism assigning finitely many pure strategies to each player can out-perform the Vickrey one in the worst case.

In Sum. Our theorems together show that, for risk-neutral players, the classical Vickrey mechanism is very robust to alternative specifications of preferences and information structures. Indeed, as most things classical, it outlives the confines in which it was conceived, and continues to be relevant in new and unforeseen settings. We believe that such robustness is an important property of a mechanism.

Related Work. Knightian uncertainty has received much attention in decision theory. Aumann [14]; Dubra, Maccheroni and Ok [55]; Ok [124]; and Nascimento [112] investigate decision with incomplete orders of preferences. Various criteria for selecting a single distribution out of a set of distributions have been studied by Danan [49]; Schmeidler [139]; Gilboa and Schmeidler [70]; and Maccheroni, Marinacci and Rustichini [102]. Bose, Ozdenoren and Pape [34] and Bodoh-Creed [33] use the model from Gilboa and Schmeidler [70] to study auctions. General equilibrium models with incompletely ordered preferences have been considered by Mas-Colell [105]; Gale and Mas-Colell [67]; Shafer and Sonnenschein [141]; and Fon and Otani [64]. Rigotti and Shannon [135] have characterized the set of equilibria in a financial market problem with incomplete preferences.

Mechanisms with Knightian uncertainty were first considered by Lopomo, Rigotti, and Shannon [99]. They do not focus on auctions, but on the rental extraction problem. (See Appendix 1.G for a technical comparison.)

Lopomo, Rigotti, and Shannon also studied variants of the notions they proposed in [99] for a principal-agent model with Knightian uncertainty [100].

Di Tillio, Kos and Messner [53] and Bose and Renou [35] have studied *ambiguous* mechanisms, assuming that the players have maxmin preferences [70]. Informally, ambiguous mechanisms do not map a profile of strategies to a single outcome, but to an outcome arbitrarily chosen from a set of outcomes. Thus, in a sense, they 'exogenously introduce Knightian uncertainty'.

Full implementation in (traditional) undominated strategies was proposed by Jackson [79, 80]. An example of such implementation in the exact-valuation model is given by the mechanism of Babaioff et al. [21] for efficiency in multi-good auctions where each player may be interested in different bundles of the goods, but has the same value for each such bundle.

1.2 Model

1.2.1 Notation for Multi-Unit Auctions

We study auctions of a homogenous good in which players have multi-unit demand. We denote by n the number of players; by m the number of copies of the good; by $[n]$ the set $\{1, 2, \ldots, n\}$; and by $[m]$ the set $\{1, 2, \ldots, m\}$. The set of all possible allocations is $\mathcal{A} \stackrel{\text{def}}{=} \left\{A \in \mathbb{Z}^n_{\geq 0} \,\middle|\, \sum_{i=0}^{n} A_i = m\right\}$. In an allocation $A \in \mathcal{A}$, A_0 is the number of unallocated copies and A_i the number of copies allocated to player i.

As in [15, 159], we assume non-increasing marginal valuations. For each player i, the set of possible valuations is $\Theta_i \stackrel{\text{def}}{=} \{\theta_i : [m] \to \mathbb{R}_{\geq 0} \mid \theta_i(1) \geq \cdots \geq \theta_i(m) \geq 0\}$, where for each valuation $\theta_i \in \Theta_i$ and each copy $j \in [m]$, $\theta_i(j)$ represents player i's marginal value for a j-th copy of the good. (We may also refer to such a θ_i as an m-dimensional vector, and to $\theta_i(j)$ as its j-th coordinate.) The set of all possible valuation profiles is $\Theta \stackrel{\text{def}}{=} \Theta_1 \times \cdots \times \Theta_n$. The profile of the players' true valuations is $\theta^* \stackrel{\text{def}}{=} (\theta^*_1, \ldots, \theta^*_n) \in \Theta$.

The set of possible outcomes is $\Omega \stackrel{\text{def}}{=} \mathcal{A} \times \mathbb{R}^n_{\geq 0}$. If $(A, P) \in \Omega$, we refer to P_i as the price charged to player i. The utility of a player i, with valuation θ_i, for an outcome $\omega = (A, P)$ is $U_i(\theta_i, \omega) \stackrel{\text{def}}{=} \sum_{j=1}^{A_i} \theta_i(j) - P_i$.

For every set X, we denote by $\Delta(X)$ the set of all countably additive probability measures on X. If $\omega \in \Delta(\Omega)$, then $U_i(\theta_i, \omega)$ is the expected utility of player i.

Relative to a valuation profile θ, the social welfare of an outcome $\omega = (A, P) \in \Omega$, or the social welfare of an allocation $A \in \mathcal{A}$, is $\mathrm{SW}(\theta, \omega) = \mathrm{SW}(\theta, A) \stackrel{\text{def}}{=} \sum_i \sum_{j=1}^{A_i} \theta_i(j)$. The maximum social welfare relative to θ is $\mathrm{MSW}(\theta) \stackrel{\text{def}}{=} \max_{A \in \mathcal{A}} \mathrm{SW}(\theta, A)$. The maximum social welfare is $\mathrm{MSW} \stackrel{\text{def}}{=} \mathrm{MSW}(\theta^*)$.

A mechanism M specifies, for each player i, a set of strategies S_i. We interchangeably refer to each member of S_i as a pure *strategy/action/report* of i, and, similarly, to a member of $\Delta(S_i)$ as a mixed strategy/action/report of i.[1] After each player i, simultaneously with his opponents, reports a strategy s_i in S_i, M maps the reported strategy profile s to an outcome $M(s) \in \Omega$. If M is probabilistic, then $M(s) \in \Delta(\Omega)$.[2]

When in a mechanism M the players jointly choose a profile of (possibly mixed) strategies $\sigma = (\sigma_1, \ldots, \sigma_n) \in \Delta(S_1) \times \cdots \times \Delta(S_n)$, we respectively denote by $M^{\mathsf{P}}_i(\sigma)$ and $M^{\mathsf{A}}_{i,j}(\sigma)$ the expected price of player i and the probability that player i wins j copies of the good.

[1] Often, in pre-Bayesian settings, the notion of a strategy and that of an action are distinct. Indeed, a strategy s_i of a player i maps the set of all possible types of i to the set of i's possible actions/reports. But since strategies are universally quantified in all relevant definitions of this paper, we need not separate (and for simplicity do not separate) the notions of strategies and actions.

[2] With our risk-neutral players, it would suffice to consider outcomes drawn from $\Delta(\mathcal{A}) \times \mathbb{R}^n_{\geq 0}$.

1.2.2 Knightian Valuation Uncertainty

In our model the players are risk-neutral and a player i's sole information about the entire true valuation profile $\theta^* = (\theta_1^*, \ldots, \theta_n^*)$ consists of a non-empty set of distributions, $\mathcal{K}_i \subset \Delta(\Theta_i)$, from one of which θ_i^* has been drawn. (The players' true valuations are uncorrelated.)

Because a risk-neutral player cares only about his expected utility, and because in an auction each Θ_i is convex, in our model a player i may 'collapse' each distribution $D_i \in \mathcal{K}_i$ to its expectation $\mathbb{E}_{\theta_i \sim D_i}[\theta_i] \in \Theta_i$. Accordingly, for auctions, our model can be equivalently restated in the following non-distributional language.

Definition 1.1 (Knightian valuation model)**.** *For each player i, i's sole information about θ^* is a non-empty set $K_i \subset \Theta_i$, the* candidate (valuation) set of i*, such that $\theta_i^* \in K_i$. We refer to an element of K_i as a* candidate valuation*. We denote by $\mathbb{K}_i$ the set of all possible candidate sets of i, and let $\mathbb{K} \stackrel{\text{def}}{=} \mathbb{K}_1 \times \cdots \times \mathbb{K}_n$.*

We stress that $\mathbb{K}_i$ can be an arbitrary subset of 2^{Θ_i} and that, in our model, i has no information about the true valuation θ_j^* or the candidate set K_j of an opponent j.

In this paper, we refer to a player or an auction as *Knightian* to emphasize that we are considering the player or the auction in the Knightian valuation model.

In this model, a mechanism's performance will of course depend on the inaccuracy of the players' candidate sets, which we measure as follows.

Definition 1.2. *For all players i, candidate set K_i, and copies $j \in [m]$, we let*

$$K_i(j) \stackrel{\text{def}}{=} \{\theta_i(j) \mid \theta_i \in K_i\}, \quad K_i^{\perp}(j) \stackrel{\text{def}}{=} \inf K_i(j), \quad \text{and} \quad K_i^{\top}(j) \stackrel{\text{def}}{=} \sup K_i(j).$$

A candidate set K_i is (at most) δ-approximate *if $K_i^{\top}(j) - K_i^{\perp}(j) \leq \delta$ for all $j \in [m]$. An auction is* (at most) δ-approximate *if, for each player i,*

$$\mathbb{K}_i \subset \mathbb{K}_i^{\delta} \stackrel{\text{def}}{=} \{K_i \in 2^{\Theta_i} \mid K_i \text{ is } \delta\text{-approximate}\}.$$

We set $\mathbb{K}^{\delta} \stackrel{\text{def}}{=} \mathbb{K}_1^{\delta} \times \cdots \mathbb{K}_n^{\delta}$.

Note that a candidate set K_i may not be convex. For instance, in a single-good auction, K_i may consist of the two valuations a and b, and thus not contain $\frac{a+b}{2}$. Let us stress that the possibility of 'holes' in K_i is the necessary sub-product of the fact that each K_i is derived from an underlying set of distributions, $\mathcal{K}_i$, which is allowed to be totally arbitrary.[3]

[3]Note that candidate sets may be very *expressive.* In a single-good setting, consider a player i who believes that his true valuation is either a or b, but more probably a than b. This belief corresponds to the set of distributions $\mathcal{K}_i' = \{D_p \mid p \in [0.5, 1]\}$ where each D_p is the distribution taking value a with probability p, and value b with probability $1 - p$. Then, if i collapses each distribution D_p to its expected value, he *de facto* ends with the following set of candidate valuations: $K_i' = \{pa + (1-p)b \mid p \in [0.5, 1]\} \subseteq \Theta_i$. (If, after translating the above belief to a new candidate set K_i', player i formed further beliefs about the probabilities of the valuations in K_i', then he could again translate these beliefs to a new candidate set K_i''. And so on.)

1.3 The First Theorem

In this section, we prove that, under natural conditions, all dominant-strategy (and ex-post Nash) mechanisms must yield inefficient allocations in the Knightian valuation model. We stress that this result holds when such mechanisms are allowed to elicit from each player not just a single valuation, but an arbitrary report: in particular, a set of valuations.

Since it is easy to see that the revelation principle continues to apply in our setting (see Appendix 1.A only for completeness sake), we state Theorem 1.6 in terms of Knightian dominant-strategy truthfulness mechanisms, formally defined below.

Recall that $\mathbb{K}_i$ is the set of all possible candidate sets of player i.

Definition 1.3. *A mechanism is* Knightian direct *if, for each player i, $S_i = \mathbb{K}_i$. Such a mechanism M is* Knightian dominant-strategy-truthful (Knightian DST) *if*

$$\forall K_i, K_i' \in \mathbb{K}_i \; \forall K_{-i} \in \mathbb{K}_{-i} \; \forall \theta_i \in K_i \qquad U_i\big(\theta_i, M(K_i, K_{-i})\big) \geq U_i\big(\theta_i, M(K_i', K_{-i})\big).$$

To state Theorem 1.6, we also define a simple relation between candidate sets.

Definition 1.4. *In an m-unit auction, two candidate sets K_i and K_i' in $\mathbb{K}_i$ are*

- adjacent, *if* $\operatorname{span}\big\{\big(\theta_i(1) - \theta_i'(1), \ldots, \theta_i(m) - \theta_i'(m)\big) \,\big|\, \theta_i, \theta_i' \in K_i \cap K_i'\big\} = \mathbb{R}^m$, *and*
- connected, *if there exist $K_i^{(1)}, \ldots, K_i^{(t)} \in \mathbb{K}_i$ such that $K_i = K_i^{(1)}$, $K_i' = K_i^{(t)}$, and $K_i^{(k)}$ is adjacent to $K_i^{(k+1)}$ for all $k \in \{1, \ldots, t-1\}$.*

Example 1.5. When $m = 1$, that is, in the case of single-good auctions, each candidate set is a subset of the non-negative reals, and thus two candidate sets K_i and K_i' in $\mathbb{K}_i$ are adjacent if and only if $|K_i \cap K_i'| \geq 2$. Indeed, taking two different reals $x, y \in K_i \cap K_i'$, the fact that $x - y \neq 0$ implies that the 1-dimensional vector $(x - y)$ spans the 1-dimensional space $\mathbb{R}$. Accordingly, if the intervals $[1, 3]$, $[2, 4]$, and $[3, 5]$ are possible candidate sets in $\mathbb{K}_i$, then $[1, 3]$ is adjacent to $[2, 4]$, $[2, 4]$ is adjacent to $[3, 5]$, and $[1, 3]$ is connected (but not adjacent) to $[3, 5]$.

Consider next an m-unit auction. Let K_i be the candidate set consisting of all the valuations $\theta_i \in \Theta_i$ such that $\theta_i(j) \in [1, 3]$ for all $j \in [m]$, and K_i' the candidate set consisting of all the valuations $\theta_i' \in \Theta_i$ such that $\theta_i'(j) \in [2, 4]$ for all $j \in [m]$. Then, K_i and K_i' are adjacent if they both belong to $\mathbb{K}_i$. This is so because the set of m-dimensional vectors $\big\{\big(\theta_i(1) - \theta_i'(1), \ldots, \theta_i(m) - \theta_i'(m)\big) \,\big|\, \theta_i, \theta_i' \in K_i \cap K_i'\big\}$ contains the m vectors $(1, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, $\cdots$, $(0, \ldots, 0, 1)$, which span $\mathbb{R}^m$.

When we say that the candidate set is K_i, we assume that all (partial) beliefs that player i may have about his own valuation θ_i^* have already been taken into account.

Theorem 1.6. *In an m-unit Knightian auction, for all $\delta > 0$, all $\mathbb{K} \subseteq \mathbb{K}^\delta$, all (possibly probabilistic) Knightian DST mechanisms M,*[4] *all $(K_1, \ldots, K_n) \in \mathbb{K}$, all players i, all $K_i' \in \mathbb{K}_i$ connected to K_i, and all copies $j \in [m]$,*

$$M_{i,j}^{\mathsf{A}}(K_i, K_{-i}) = M_{i,j}^{\mathsf{A}}(K_i', K_{-i}) \quad \textit{and} \quad M_i^{\mathsf{P}}(K_i, K_{-i}) = M_i^{\mathsf{P}}(K_i', K_{-i}) \ .$$

The proof of Theorem 1.6 can be found in Appendix 1.B.

Theorem 1.6 essentially states that the probability that a Knightian DST mechanism M assigns a given number of copies of the good to a given player i, and also the price player i pays, are independent of the candidate sets i reports, provided that they are connected and that the reports of i's opponents are fixed.

This independence from individual players' reports prevents a Knightian DST mechanism from guaranteeing high social welfare, when the players' possible candidate sets are sufficiently rich. For instance, consider a single-good auction in which $\delta = 2$, and each $\mathbb{K}_i$ includes the intervals $[0,2]$, $[1,3]$, $[2,4], \ldots, [B, B+2]$ for some large integer B. Then, no matter what the DST mechanism M might be, when the reported profile of candidate sets is $K = \big([0,2], [0,2], \ldots, [0,2]\big) \in \mathbb{K}$, one of the players, without loss of generality player 1, must receive the good with probability at most $1/n$: in symbols, $M_{1,1}^{\mathsf{A}}(K) \leq 1/n$. This implies that the probability that player 1 gets the good remains at most $1/n$ even when all his opponents report the interval $[0,2]$ and he reports $[B, B+2]$. This is so because the intervals $[0,2]$ and $[B, B+2]$ are connected and thus Theorem 1.6 implies that $M_{1,1}^{\mathsf{A}}\big([0,2], [0,2], \ldots, [0,2]\big) = M_{1,1}^{\mathsf{A}}\big([B, B+2], [0,2], \ldots, [0,2]\big)$. Accordingly, if $[B, B+2]$ were the true candidate set of player 1, and $[0,2]$ the true candidate set for everyone else, then the maximum social welfare would be at least B, while the expected social welfare delivered by M would be at most $B/n + 2$.[5]

1.4 The Second Theorem

Our second theorem proves a very attractive relationship between a player's candidate set and his undominated strategies in the Vickrey mechanism for multi-unit

[4] Note that Theorem 1.6 holds even if the mechanism M is allowed to know δ and $\mathbb{K}$ in advance.

[5] We note that such poor social-welfare performance indeed relies on the richness of the players' possible candidate sets. If the players' possible candidate sets were guaranteed to be *sufficiently separated*, then a properly designed dominant-strategy mechanism could always achieve the *maximum* social welfare. For instance, consider an n-player auction of a single good where

- the inaccuracy parameter $\delta = 1/3$,
- the set of possible candidate sets $\mathbb{K}_i = \{[kn + i, kn + i + \frac{1}{3}] \,|\, k \in \mathbb{Z}_+\}$ for each player i, and
- the mechanism M is such that (1) $S_i = \{kn + i \,|\, k \in \mathbb{Z}_+\}$ for each player i, and (2) for all $s \in S_1 \times \cdots \times S_n$, $M(s) = \mathsf{2P}(s)$, where $\mathsf{2P}$ is the second-price mechanism.

Then, it is clear that (a) for player i whose true candidate set is $[kn + i, kn + i + \frac{1}{3}]$, reporting $kn + i$ is a dominant strategy, and (b) when dominant strategies are played, M produces an outcome with maximum social welfare.

Knighitian auctions.

Recall that the Vickrey mechanism, denoted by $\mathsf{Vickrey}$, is a direct mechanism (i.e., satisfies $S_i = \Theta_i$) and maps a profile of valuations $\theta \in \Theta_1 \times \cdots \times \Theta_n$, to an outcome (A, P); where $A \in \arg\max_{A \in \mathcal{A}} \mathrm{SW}(\theta, A)$, $P_i = \mathrm{MSW}(\theta_{-i}) - \sum_{k \neq i} \sum_{j=1}^{A_k} \theta_k(j)$, and possible ties are broken lexicographically.[6]

For the Knightian valuation model, we define undominated strategies as follows.

Definition 1.7. *In a mechanism M, a pure strategy $s_i \in S_i$ of a player i is* (weakly) dominated *by another possibly mixed strategy $\sigma_i \in \Delta(S_i)$ of i with respect to his K_i, in symbols $s_i \prec_{(i,K_i)} \sigma_i$, if*

(1) $\forall \theta_i \in K_i \; \forall s_{-i} \in S_{-i} \quad U_i\big(\theta_i, M(\sigma_i, s_{-i})\big) \geq U_i\big(\theta_i, M(s_i, s_{-i})\big)$, *and*

(2) $\exists \theta_i \in K_i \; \exists s_{-i} \in S_{-i} \quad U_i\big(\theta_i, M(\sigma_i, s_{-i})\big) > U_i\big(\theta_i, M(s_i, s_{-i})\big)$.[7]

A strategy $s_i \in S_i$ is (weakly Knightian) undominated, *if there exists no $\sigma_i \in \Delta(S_i)$ such that $s_i \prec_{(i,K_i)} \sigma_i$. We denote the set of undominated strategies of player i by $\mathsf{UD}_i(K_i)$.*

If K is a product or a profile of candidate sets, that is, if $K = (K_1, \ldots, K_n)$ or $K = K_1 \times \cdots \times K_n$, then $\mathsf{UD}(K) \stackrel{\text{def}}{=} \mathsf{UD}_1(K_1) \times \cdots \times \mathsf{UD}_n(K_n)$.

Our notion of an undominated strategy intends to capture the 'weakest condition' for which s_i should be discarded in favor of σ_i, and is a natural extension of its classical counterpart.[8]

Note that Jackson's more involved definition of an undominated strategy is not necessary in our paper.[9]

Now let us formally state our second theorem.

---

[6] More precisely, on a reported valuation profile θ, the Vickrey mechanism sorts the values $\{\theta_i(j) \,|\, i \in [n], j \in [m]\}$ in a non-increasing order, and then chooses the m largest entries to assign the m copies of the good. Namely, if $\theta_i(1), \theta_i(2), \ldots, \theta_i(j)$ belong to the largest m entries, but not $\theta_i(j+1)$, then $\mathsf{Vickrey}$ assigns j copies of the good to player i. If ties occur in this ordering, that is, if $\theta_i(j) = \theta_{i'}(j')$, then $\theta_i(j)$ precedes $\theta_{i'}(j')$ if and only if either (1) $i < i'$ or (2) $i = i'$ and $j < j'$.

[7] This notion is thus different from *strong dominance*, where inequality (1) is always strict. For strong dominance in the the exact-valuation case, see, for instance, [66, 95].

[8] Of course, other extensions are also possible. To express condition (2) in Definition 1.7, we must quantify the true valuation $\theta_i \in K_i$ and the pure strategy subprofile of i's opponents $s_{-i} \in S_{-i}$. There are three alternatives to consider. Namely, (a) $\forall \theta_i \forall s_{-i}$, (b) $\exists \theta_i \forall s_{-i}$, and (c) $\forall \theta_i \exists s_{-i}$. Alternatives (a) and (b) do not yield the classical notion of (weak) dominance when K_i is a singleton. Alternative (c) fails to capture the 'weakest condition' for which s_i should be discarded in favor of σ_i. (Indeed, since σ_i is already no worse than s_i, for player i to discard strategy s_i in favor of σ_i, it should suffice for s_i to be strictly worse than σ_i for a *single* possible valuation $\theta_i \in K_i$.)

[9] To meaningfully deal with the possibility of having an infinite sequence of pure strategies one dominating another, Jackson put forward, in the exact-valuation case, a more involved notion of an undominated strategy [79]. However, this more involved notion is unnecessary, even in the Knightian setting, for the class of *bounded* mechanisms. This class includes the Vickrey and all finite mechanisms, and thus all mechanisms analyzed in this paper in undominated strategies.

Theorem 1.8. *In an m-unit Knightian auction with the Vickrey mechanism, for all players i and all candidate sets K_i, the set of undominated strategies $\mathsf{UD}_i(K_i)$ coincides with the set of all strategies $v_i \in \Theta_i$ satisfying the following condition*

$$\forall j \in [m] \quad v_i(j) \in \left[K_i^{\perp}(j), K_i^{\top}(j)\right] \ .$$

Theorem 1.8 is proved in Appendix 1.C.

Theorem 1.8 is obvious for $m = 1$, but less obvious when there are multiple copies of the good. In particular, a player i may consider 'under-reporting' his value for the j-th copy of the good, but 'over-reporting' his value for the k-th copy. For example, in a 3-unit auction, where K_i consists of all valuations $\theta_i \in \Theta_i$ such that

$$\theta_i(1) \in [100, 110], \quad \theta_i(2) \in [95, 105], \quad \text{and} \quad \theta_i(3) \in [90, 100] \ ,$$

by reporting the valuation $v_i = (113, 98, 80)$, i over-reports his value for the first copy but under-reports that for the third copy. Such a strategy v_i is, in general, *not dominated* by reporting the highest —respectively, the lowest— possible value for each copy of the good: that is, it is not dominated by reporting $(110, 105, 100)$ —respectively, $(100, 95, 90)$. However, one can still carefully construct a strategy v_i^* that dominates v_i, and therefore conclude that a rational player will not use any such strategy v_i. In our example, such a v_i^* could be $(110, 98, 80)$, $(113, 98, 90)$, or $(110, 98, 90)$. A general (but not the only) way to construct a v_i^* dominating v_i is to set $v_i^*(j) = v_i(j)$ for every copy j such that $v_i(j)$ belongs to $[K_i^{\perp}(j), K_i^{\top}(j)]$, set $v_i^*(j) = K_i^{\perp}(j)$ if $v_i < K_i^{\perp}(j)$, and set $v_i^*(j) = K_i^{\top}(j)$ if $v_i > K_i^{\top}(j)$. (We rely on the non-increasing marginal valuation assumption in order to show that the so-constructed v_i^* dominates v_i.)

The above construction of v_i^* is the key idea to show that every $v_i \in \mathsf{UD}_i(K_i)$ satisfies $v_i(j) \in \left[K_i^{\perp}(j), K_i^{\top}(j)\right]$ for every copy j. A similar idea is needed to show the other direction. The details can be found in Appendix 1.C.

Remark 1.9. For a Knightian player i, the set of undominated strategies $\mathsf{UD}_i(K_i)$ may strictly contain the candidate set K_i. For example, in a single-good second-price auction, if $K_i = \{4, 7, 21\}$, then not only reporting 4, 7, or 21 is an undominated strategy for player i, but so is reporting 9. As for another example, in a 2-unit Vickrey auction, if $K_i = \{(78, 60), (80, 50)\}$, then not only reporting $(78, 50)$ and $(80, 50)$, but also $(79, 51)$, $(79, 52)$, ..., $(79, 59)$, etc., are undominated strategies.

The multiplicity of undominated strategies in the two examples above emphasizes that our Knightian player i has incomplete preferences. Assume for a moment that he had complete preferences: for instance, maxmin preferences. Then, his only undominated strategy (and thus his only dominant strategy) would consist of reporting

4 in the former example, and $(78, 52)$ in the latter one.[10]

Theorem 1.8 has a simple corollary (proved for completeness in Appendix 1.D) about the social-welfare performance of the Vickrey mechanism.

Corollary 1.10. *In an m-unit Knightian auction, for all $\delta \geq 0$, all products K of δ-approximate candidate sets, all profiles $v \in \mathsf{UD}(K)$, and all $\theta \in K$*

$$\mathrm{SW}\big(\theta, \mathsf{Vickrey}(v)\big) \geq \mathrm{MSW}(\theta) - 2m\delta \ .$$

That is, the social welfare realized by the Vickrey mechanism is at most $2m\delta$ away for the maximum one, no matter which undominated strategies the players may choose. The following example shows that this performance guarantee of the Vickrey mechanism is actually tight in the worst case.

Example 1.11. Consider a two-player 10-approximate m-unit auction in which the candidate sets are

$$\begin{aligned} K_1 &= \big\{(90, 90, \ldots, 90), (100, 100, \ldots, 100)\big\} \\ K_2 &= \big\{(100, 100, \ldots, 100), (110, 110, \ldots, 110)\big\} \ . \end{aligned}$$

In this case, the Vickrey mechanism may miss the maximum social welfare by $2\delta m$ as follows. Player 1 is 'optimistic' and bids the valuation $v_1 = (100, \ldots, 100)$; player 2 is 'pessimistic' and bids $v_2 = (100, \ldots, 100)$; the Vickrey mechanism (with the lexicographic tie-breaking rule) allocates all copies of the good to player 1; the true valuation θ_1^* of player 1 is $(90, \ldots, 90)$; and the true valuation θ_2^* of player 2 is $(110, \ldots, 110)$.

Accordingly, the realized social welfare is $90m$, while the maximum one is $110m = 90m + 2m\delta$. □

The relevance of worst-case analyses can of course be debated, but if the worst-case performance is good, then the typical performance can only be better. In our setting, a social-welfare loss of $2m\delta$ is small whenever δ is small relative to $\mathrm{MSW}(\theta)/m$. For instance, this is the case of a 10-unit auction in which a player's valuation for each copy of the good is a million dollars plus or minus \$100. Indeed, in this case, Corollary 1 implies that the Vickrey mechanism guarantees that the realized social welfare will

[10] In the spirit of Gilboa and Schmeidler [70], given an outcome $\omega = (A, P)$, we can define the worst-case utility of a player i with candidate set K_i to be $\min_{\theta_i \in K_i} U_i(\theta_i, \omega) = \min_{\theta_i \in K_i} \sum_{j=1}^{A_i} \theta_i(j) - P_i$. In the above 2-unit auction example, this worst-case utility is $-P_i$, if $A_i = 0$; is $78 - P_i$, if $A_i = 1$; and is $130 - P_i$, if $A_i = 2$. Therefore, i's worst-case utility coincides with the utility of a (non-Knightian) player whose true valuation is precisely $(78, 52)$. Indeed, $\min_{\theta_i \in K_i} U_i(\theta_i, \omega) = U_i\big((78, 52), \omega\big)$ for *all* possible outcomes ω. Now, a player i with maxmin preferences compares every two outcomes ω and ω' using his worst-case utility function, $\min_{\theta_i \in K_i} U_i(\theta_i, \cdot)$. It is thus equivalent for such a player i to compare ω and ω' using the exact utility function $U_i\big((78, 52), \cdot\big)$. Accordingly, since the Vickrey mechanism is dominant-strategy in the classical setting, it is a dominant strategy for i to report $(78, 52)$ in the above 2-unit auction.

be at least ten millions minus $\$2,000$, no matter how the players may choose their undominated strategies. (A performance loss that is at most linear in δ should not be underestimated. After all, Theorem 1.6 shows that the social welfare performance of any DST mechanism can be terrible, no matter how small, but positive, δ may be.)

1.5 The Third Theorem

Our third theorem shows that the worst-case social welfare performance of the Vickrey mechanism is essentially optimal, in the Knightian setting, relative to all possible undominated-strategy mechanisms.

Note that, in principle, there may be an undominated-strategy mechanism M missing the maximum social welfare by at most δm.[11] Our upcoming Theorem 1.14, however, rules out the existence of such mechanisms, so long as they give each player a finite set of strategies.[12]

We stress that Theorem 1.14 applies not just to finite mechanisms eliciting a single valuation from each player, but to *all* finite mechanisms, including those allowing a player to report a set of valuations. Thus, in our Knightian valuation model, the social welfare optimality of the Vickrey mechanism (which allows a player to report only a single valuation) may be surprising.

We could simply state Theorem 1.14 by saying that, for every finite mechanism M, there exists a profile of δ-approximate candidate sets for which M misses the maximum social welfare by essentially $2m\delta$: more precisely, by $2m\delta(1 - 1/n) + \varepsilon$, where ε is an arbitrarily small positive constant. To be more informative, however, we wish to state Theorem 1.14 so as to highlight the candidate set profiles causing this maximal loss in social welfare.

Let V and W be two sets of real numbers (with diameter at most δ and with at least two elements in common), whose union has diameter at least $2\delta - \varepsilon/m$. For instance, $V = [x - \delta, x]$ and $W = [x - 2\delta + \varepsilon/m, x - \delta + \varepsilon/m]$. Then, the following definition expresses that for each player there are at least two candidate sets, one for which the value of each copy of the good is in V, and one for which that the value of each copy of the good is in W. More precisely, recalling that $\mathbb{K}_i$ is the set of all possible candidate sets of player i, that $\mathbb{K} = (\mathbb{K}_1, \ldots, \mathbb{K}_n)$, and that in a δ-approximate multi-unit Knightian auction $\mathbb{K} \subseteq \mathbb{K}^\delta$, we have the following

Definition 1.12. *In a δ-approximate multi-unit Knightian auction, $\mathbb{K}$ is* ε-basic *if there exist two subsets V and W of non-negative numbers such that*

[11] For instance, a mechanism M could achieve such performance by asking each player i to report a single valuation, and incentivizing him to report a valuation v_i which is the 'mid-point' of his candidate set K_i: i.e., $v_i(j) = \frac{1}{2}(K_i^\perp(j) + K_i^\top(j))$ for all $j \in [m]$.

[12] This finiteness restriction, although crucial for our proof, is quite mild in practice (and is indeed natural when mechanisms are implemented via computers). The Vickrey mechanism itself becomes finite if it explicitly asks each player to report, for each copy of the good, an integral number of cents between 0 and 10^{100}.

(a) $\max V - \min V \leq \delta$ *and* $\max W - \min W \leq \delta$,
(b) $|V \cap W| > 1$ *and* $\max V - \min W \geq 2\delta - \varepsilon/m$, *and*
(c) for every player i, $\mathbb{K}_i$ *contains the following two candidate sets*

$$\widetilde{K}_i \stackrel{\text{def}}{=} \{\theta_i \in \Theta_i \mid \forall j, \theta_i(j) \in V\} \quad \textit{and} \quad \widetilde{K}'_i \stackrel{\text{def}}{=} \{\theta_i \in \Theta_i \mid \forall j, \theta_i(j) \in W\} \ .$$

Example 1.13. Consider a 3-unit 10-approximate Knightian auction, in which for each player i, $\mathbb{K}_i$ includes the following two candidate sets:

$$[88, 98] \times [88, 98] \times [88, 98] \quad \text{and}$$
$$[80, 90] \times [80, 90] \times [80, 90] \ .$$

Then, $\mathbb{K}$ is 6-basic (corresponding to $V = [88, 98]$ and $W = [80, 90]$).

There is no magic about the choice of the numbers 80 and 88 in the above example. The main point is that the intersection of the two intervals $[80, 90]$ and $[88, 98]$ coincides with the interval $[88, 90]$, whose length is $\varepsilon/m = 6/3 = 2$.

Theorem 1.14. *In a multi-unit Knightian auction, for all* $\delta > 0$, *all* $\varepsilon > 0$, *all* ε*-basic* $\mathbb{K} \subseteq \mathbb{K}^\delta$, *all (possibly probabilistic) finite mechanisms* M, *there exist products* $K \in \mathbb{K}$, *valuation profiles* $\theta \in K$, *and undominated strategy profiles* $s \in \mathsf{UD}(K)$, *such that*

$$\mathbb{E}\big[\mathrm{SW}(\theta, M(s))\big] \leq \mathrm{MSW}(\theta) - 2\delta m(1 - 1/n) + \varepsilon \ .$$

Above, the expectation is over the possible random choices of the mechanism M.

The proof of Theorem 1.14 can be found in Appendix 1.E. Although mechanism finiteness is a natural restriction in practice, we wish to remark that Theorem 1.14 continues to hold under alternative but more complex assumptions.[13]

Appendix

1.A Knightian Revelation Principle

Let us explicitly show that a version of the revelation principle [69, 50, 110] holds also in our Knightian setting. Recall that $\mathbb{K}_i$ is the set of all possible candidate sets for player i.

[13] As it will become clear from our proof, Theorem 1.14 also holds for all bounded mechanisms such that, for all players i, the strategy set S_i is a compact Hausdorff space and, for all copies j, the families of allocation functions $\big\{M^{\mathsf{A}}_{i,j}(s_i, \cdot)\big\}_{s_i \in S_i}$ and price functions $\big\{M^{\mathsf{P}}_i(s_i, \cdot)\big\}_{s_i \in S_i}$ are equicontinuous. However, the Vickrey mechanism can be trivially modified to be finite, but not trivially made equicontinuous.

Definition 1.15. *Let M be a mechanism in which S_i is the set of actions of player i. Then, the profile of functions $\big(s_i\colon \mathbb{K}_i \to \Delta(S_i)\big)_i$ is an* ex-post (possibly mixed) Nash equilibrium *of M if for all $K \in \mathbb{K}$, all players i, all $a_i \in S_i$, and all $\theta_i \in K_i$,*

$$U_i(\theta_i, M(s_i(K_i), s_{-i}(K_{-i}))) \geq U_i(\theta_i, M(a_i, s_{-i}(K_{-i}))) \ .$$

Lemma 1.16 (**Revelation Principle**)**.** *Let M be a mechanism that has an ex-post Nash equilibrium s. Then, there exists a Knightian DST mechanism M' such that*

$$\forall K \in \mathbb{K} \quad M'(K_1, \ldots, K_n) = M(s_1(K_1), \ldots, s_n(K_1)) \ .$$

Proof. Let M' be the Knightian direct mechanism so defined:

$$\forall K \in \mathbb{K} \quad M'(K_1, \ldots, K_n) \stackrel{\text{def}}{=} M(s_1(K_1), \ldots, s_n(K_1)) \ .$$

(The above equality is between distributions if M is probabilistic.)

All that is left to prove is that the mechanism M' is dominant-strategy truthful. To this end, let K_i be the true candidate set of player i. Then, for all $K_i' \in \mathbb{K}_i$, all $\theta_i \in K_i$, and all $K_{-i} \in \mathbb{K}_{-i}$,

$$\begin{aligned} U\big(\theta_i, M'(K_i, K_{-i})\big) &= U\big(\theta_i, M(s_i(K_i), s_{-i}(K_{-i}))\big) \\ \geq U\big(\theta_i, M(s_i(K_i'), s_{-i}(K_{-i}))\big) &= U\big(\theta_i, M'(K_i', K_{-i})\big) \ , \end{aligned}$$

where the inequality follows from Definition 1.15 of the ex-post Nash equilibrium by setting $a_i = s_i(K_i')$. This completes the proof. □

Because every dominant-strategy mechanism must have an ex-post Nash equilibrium (consisting of each player choosing his dominant strategy), the above theorem holds also when M is a dominant-strategy mechanism.

1.B Proof of Theorem 1.6

We start by proving, as a separate claim, that Theorem 1.6 holds in the case of *adjacent* (instead of *connected*) candidate sets. Namely,

Claim 1.17. *For every player i, every two adjacent candidate sets $K_i, K_i' \in \mathbb{K}_i$ of i, and every subprofile K_{-i} of candidate sets for i's opponents,*

$$M_{i,j}^{\mathsf{A}}(K_i, K_{-i}) = M_{i,j}^{\mathsf{A}}(K_i', K_{-i}) \quad \textit{and} \quad M_i^{\mathsf{P}}(K_i, K_{-i}) = M_i^{\mathsf{P}}(K_i', K_{-i}) \ .$$

Proof. Because the true candidate set of player i may coincide with K_i, and because, when this is the case, reporting K_i should dominate reporting K_i', we have that, for all $\theta_i'' \in K_i$, the following inequality holds:

$$\sum_{j=1}^{m} M_{i,j}^{\mathsf{A}}(K_i, K_{-i}) \cdot \sum_{\ell=1}^{j} \theta_i''(\ell) - M_i^{\mathsf{P}}(K_i, K_{-i}) \geq \sum_{j=1}^{m} M_{i,j}^{\mathsf{A}}(K_i', K_{-i}) \cdot \sum_{\ell=1}^{j} \theta_i''(\ell) - M_i^{\mathsf{P}}(K_i', K_{-i}). \tag{1.1}$$

Similarly, because the true candidate set of player i may coincide with K_i', and because, when this is the case, reporting K_i' should dominate reporting K_i, we also

have that, for all $\theta_i'' \in K_i'$, the following inequality holds:

$$\sum_{j=1}^{m} M_{i,j}^{\mathsf{A}}(K_i', K_{-i}) \cdot \sum_{\ell=1}^{j} \theta_i''(\ell) - M_i^{\mathsf{P}}(K_i', K_{-i}) \geq \sum_{j=1}^{m} M_{i,j}^{\mathsf{A}}(K_i, K_{-i}) \cdot \sum_{\ell=1}^{j} \theta_i''(\ell) - M_i^{\mathsf{P}}(K_i, K_{-i}). \tag{1.2}$$

Next, for every pair of valuations $\theta_i, \theta_i' \in K_i \cap K_i'$, we choose $\theta_i'' = \theta_i$ in inequality (1.1) and $\theta_i'' = \theta_i'$ in inequality (1.2). Summing up the resulting inequalities, the M_i^{P} price terms cancel out, yielding the following inequality:

$$\sum_{j=1}^{m} M_{i,j}^{\mathsf{A}}(K_i, K_{-i}) \cdot \sum_{\ell=1}^{j} \big(\theta_i(\ell) - \theta_i'(\ell)\big) \geq \sum_{j=1}^{m} M_{i,j}^{\mathsf{A}}(K_i', K_{-i}) \cdot \sum_{\ell=1}^{j} \big(\theta_i(\ell) - \theta_i'(\ell)\big) \ . \tag{1.3}$$

Similarly, setting $\theta_i'' = \theta_i'$ in (1.1) and $\theta_i'' = \theta_i$ in (1.2), and summing up the resulting inequalities, we deduce that:

$$\sum_{j=1}^{m} M_{i,j}^{\mathsf{A}}(K_i, K_{-i}) \cdot \sum_{\ell=1}^{j} \big(\theta_i(\ell) - \theta_i'(\ell)\big) \leq \sum_{j=1}^{m} M_{i,j}^{\mathsf{A}}(K_i', K_{-i}) \cdot \sum_{\ell=1}^{j} \big(\theta_i(\ell) - \theta_i'(\ell)\big) \ . \tag{1.4}$$

(1.3) and (1.4) together imply, after some rearrangement of the terms, that

$$\sum_{j=1}^{m} \Big(M_{i,j}^{\mathsf{A}}(K_i, K_{-i}) - M_{i,j}^{\mathsf{A}}(K_i', K_{-i})\Big) \cdot \Big(\sum_{\ell=1}^{j} \theta_i(\ell) - \theta_i'(\ell)\Big) = 0 \ . \tag{1.5}$$

Now, let us denote by a the m-dimensional vector such that $a_j \stackrel{\text{def}}{=} \sum_{k=j}^{m} M_{i,k}^{\mathsf{A}}(K_i, K_{-i}) - M_{i,k}^{\mathsf{A}}(K_i', K_{-i})$ for every $j \in [m]$. Then, (1.5) can be re-written as saying that the following inner product between two vectors is zero:

$$\big(a_1, \ldots, a_m\big) \cdot \big(\theta_i(1) - \theta_i'(1), \ldots, \theta_i(m) - \theta_i'(m)\big) = 0 \ . \tag{1.6}$$

Finally, using our assumption that the vectors $\big(\theta_i(1) - \theta_i'(1), \ldots, \theta_i(m) - \theta_i'(m)\big)$ span the entire $\mathbb{R}^m$, we easily conclude that $(a_1, \ldots, a_m) = (0, \ldots, 0)$, which in turn implies the desired equality of the allocation probabilities: $M_{i,j}^{\mathsf{A}}(K_i, K_{-i}) = M_{i,j}^{\mathsf{A}}(K_i', K_{-i})$. Plugging this equality into (1.1) and (1.2) immediately yields the desired equality of the prices: $M_i^{\mathsf{P}}(K_i, K_{-i}) = M_i^{\mathsf{P}}(K_i', K_{-i})$. □

It is now straightforward to see that Theorem 1.6 follows by the definition of connected candidate sets, and the repeated applications of the above claim. Namely, recall that $K_i, K_i' \in \mathbb{K}_i$ are connected if there exist $K_i^{(1)}, \ldots, K_i^{(t)} \in \mathbb{K}_i$ such that $K_i = K_i^{(1)}$, $K_i' = K_i^{(t)}$, and $K_i^{(k)}$ is adjacent to $K_i^{(k+1)}$ for all $k \in \{1, \ldots, t-1\}$. Therefore, we conclude that, for all $j \in [m]$,

$$M_{i,j}^{\mathsf{A}}(K_i^{(1)}, K_{-i}) = M_{i,j}^{\mathsf{A}}(K_i^{(2)}, K_{-i}) = \cdots = M_{i,j}^{\mathsf{A}}(K_i^{(t)}, K_{-i}) \ ,$$

and similarly that

$$M_i^{\mathsf{P}}(K_i^{(1)}, K_{-i}) = M_i^{\mathsf{P}}(K_i^{(2)}, K_{-i}) = \cdots = M_i^{\mathsf{P}}(K_i^{(t)}, K_{-i}) \ . \qquad \blacksquare$$

1.C Proof of Theorem 1.8

Recall that the Vickrey mechanism is direct, that is, $S_i = \Theta_i$ for all players i. Recall also that multi-unit auctions have non-increasing marginal valuations, that is, $\theta_i(1) \geq \theta_i(2) \geq \cdots \geq \theta_i(m)$ for each $\theta_i \in \Theta_i$. Therefore, $K_i^{\perp}(1), \ldots, K_i^{\perp}(m)$ and $K_i^{\top}(1), \ldots, K_i^{\top}(m)$ are non-decreasing sequences. That is, $K_i^{\top}, K_i^{\perp} \in \Theta_i$. Accordingly, both $K_i^{\top}$ and $K_i^{\perp}$ are valid reports for player i in the Vickrey mechanism.

We start by proving, by contradiction, that

$$v_i \in \mathsf{UD}_i(K_i) \implies v_i(j) \geq K_i^{\perp}(j) \text{ for all } j \in [m]. \tag{1.7}$$

Assume that implication (1.7) is false; let $j^* \in [m]$ be the first coordinate j such that $v_i(j) < K_i^{\perp}(j)$; and define the function $v_i^* : [m] \to \mathbb{R}_{\geq 0}$ as follows:

$$v_i^*(j) = \begin{cases} v_i(j), & \text{if } j \neq j^*; \\ K_i^{\perp}(j), & \text{if } j = j^*. \end{cases}$$

Since v_i and $K_i^{\perp}$ are monotonically non-increasing, so is v_i^*. Indeed,

- if $j^* > 1$, then $v_i^*(j^* - 1) = v_i(j^* - 1) \geq K_i^{\perp}(j^* - 1) \geq K_i^{\perp}(j^*) = v_i^*(j^*)$
- if $j^* < m$, then $v_i^*(j^*) = K_i^{\perp}(j^*) > v_i(j^*) \geq v_i(j^* + 1) = v_i^*(j^* + 1)$.

Thus also v_i^* is a valid valuation in Θ_i. We now reach a contradiction by showing that v_i^* weakly dominates v_i, that is,

$$\forall\, \theta_i \in K_i \;\forall\, v_{-i} \qquad U_i\big(\theta_i, \mathsf{Vickrey}(v_i^*, v_{-i})\big) \geq U_i\big(\theta_i, \mathsf{Vickrey}(v_i, v_{-i})\big) \ , \tag{1.8}$$

$$\exists\, \theta_i' \in K_i \;\exists\, v_{-i}' \qquad U_i\big(\theta_i', \mathsf{Vickrey}(v_i^*, v_{-i}')\big) > U_i\big(\theta_i', \mathsf{Vickrey}(v_i, v_{-i}')\big) \ . \tag{1.9}$$

To show (1.8), choose arbitrarily $v_{-i} \in \Theta_{-i}$, and consider the following two cases:

(1) *In* $\mathsf{Vickrey}(v_i^*, v_{-i})$ *and* $\mathsf{Vickrey}(v_i, v_{-i})$*, i receives the same number of copies.*

In this case, inequality (1.8) holds because its two sides are equal for all θ_i.

(2) *In* $\mathsf{Vickrey}(v_i^*, v_{-i})$ *and* $\mathsf{Vickrey}(v_i, v_{-i})$*, i receives different numbers of copies.*

In this case, one can carefully verify that player i wins j^* copies of the good in $\mathsf{Vickrey}(v_i^*, v_{-i})$ and only $j^* - 1$ copies in $\mathsf{Vickrey}(v_i, v_{-i})$.[14] Thus, (1.8) holds because of the following two reasons:

- *i's price for his extra j^*-th copy of the good is $\leq K_i^{\perp}(j^*)$.*

[14] Recall that, when each player reports a valuation v_i, the Vickrey mechanism orders the nm values $\{v_i(j) \,|\, i \in [n], j \in [m]\}$ (breaking ties lexicographically), and allocates the m copies of the good by looking at the first m values in this order. Since the only difference between v_i^* and v_i is that $v_i^*(j^*) > v_i(j^*)$, the ordering of the reported nm values is minimally affected. That is, if player i receives different numbers of copies in outcome (v_i, v_{-i}) and outcome (v_i^*, v_{-i}), then it must be that $v_i(j^*)$ is outside the largest m numbers under (v_i, v_{-i}), but $v_i^*(j^*)$ is within the largest m numbers under (v_i^*, v_{-i}). This implies that i wins $j^* - 1$ copies in $\mathsf{Vickrey}(v_i, v_{-i})$ but j^* copies in $\mathsf{Vickrey}(v_i^*, v_{-i})$.

Indeed, Vickrey guarantees that i pays for his j^*-th copy at most the value he reports for it. That is, for his j^*-th copy, i pays at most $v_i^*(j^*)$, which in turn is equal to $K_i^{\perp}(j^*)$.

- *i's value for this j^*-th copy is $\geq K_i^{\perp}(j^*)$.*
  Indeed, for any candidate valuation θ_i in K_i, $\theta_i(j^*) \geq K_i^{\perp}(j^*)$.

Therefore, inequality (1.8) holds. Let us now show that also inequality (1.9) holds. To do so, we need to construct a 'witness' candidate valuation $\theta_i' \in K_i$ and a 'witness' strategy sub-profile v'_{-i}. In fact, we construct some v'_{-i} so that (1.9) holds for all θ_i'. Let v'_{-i} be the strategy subprofile in which, for every player $k \neq i$,

$$\forall j \in [m] \qquad v_k'(j) \stackrel{\text{def}}{=} x \stackrel{\text{def}}{=} \frac{v_i(j^*) + K_i^{\perp}(j^*)}{2} < K_i^{\perp}(j^*) \ .$$

Then, player i wins exactly j^* copies in $\mathsf{Vickrey}(v_i^*, v'_{-i})$ and pays x for each one of them; and wins exactly $j^* - 1$ copies in $\mathsf{Vickrey}(v_i, v'_{-i})$ and pays x for each one of them. Indeed, there are exactly $j^* - 1$ numbers greater than x in v_i, exactly j^* in v_i^*, and x is the reported value of every other player, in v'_{-i}, for every single copy of the good. As a result, i's utility in $\mathsf{Vickrey}(v_i^*, v'_{-i})$ is strictly greater than that in $\mathsf{Vickrey}(v_i, v'_{-i})$. This is so because in the outcome $\mathsf{Vickrey}(v_i^*, v'_{-i})$, i pays an extra price x for his j^*-th copy, while being guaranteed that his true valuation for the j^*-th copy, $\theta_i(j^*)$, is strictly larger than x, because $x < K_i^{\perp}(j^*) \leq \theta_i(j^*)$. Therefore, we conclude that (1.9) holds for all candidate $\theta_i' \in K_i$ and the above defined v'_{-i}.

Since both (1.8) and (1.9) hold, valuation v_i^* (weakly) dominates v_i, contradicting the hypothesis that $v_i \in \mathsf{UD}_i(K_i)$. This contradiction proves (1.7).

An absolutely symmetrical argument shows that[15]

$$v_i \in \mathsf{UD}_i(K_i) \implies v_i(j) \leq K_i^{\top}(j) \text{ for all } j \in [m]. \tag{1.10}$$

Together, statements (1.7) and (1.10) imply that all undominated strategies $v_i \in \mathsf{UD}_i(K_i)$ satisfy $v_i(j) \in [K_i^{\perp}(j), K_i^{\top}(j)]$ for each copy j.

Let us now prove the other direction: namely, that every strategy v_i satisfying $v_i(j) \in [K_i^{\perp}(j), K_i^{\top}(j)]$ for each copy j is undominated for player i.

We proceed by contradiction. Suppose that there exists a valuation v_i^* that weakly dominates v_i. We are going to derive a contradiction by showing that

$$\exists\, \theta_i' \in K_i \ \exists\, v'_{-i} \qquad U_i\big(\theta_i, \mathsf{Vickrey}(v_i^*, v'_{-i})\big) < U_i\big(\theta_i, \mathsf{Vickrey}(v_i, v'_{-i})\big) \ . \tag{1.11}$$

Since, by the definition of weak dominance, v_i^* must be different from v_i, there are two (not mutually exclusive) cases to consider:

(a) $v_i^*(j) > v_i(j)$ for some $j \in [m]$, and

(b) $v_i^*(j) < v_i(j)$ for some $j \in [m]$.

In case (a), let

[15] In this symmetrical case, one needs to define $j^* \in [m]$ to be the *last* coordinate such that $v_i(j^*) > K_i^{\top}(j)$.

- j^* be the first coordinate $j \in [m]$ such that $v_i^*(j) > v_i(j)$,
- y be a real number such that $v_i^*(j^*) > y > v_i(j^*)$, and
- ε be a real number in the open interval $\big(0, y - v_i(j^*)\big)$.

To show (1.11), we let θ_i' be an arbitrary valuation in K_i satisfying $\theta_i'(j^*) \leq v_i(j^*) + \varepsilon$. (Such a valuation always exists since $K_i^\perp(j^*) = \inf\{\theta_i(j^*) \,|\, \theta_i \in K_i\} \leq v_i(j^*)$.) Next, we construct the required strategy sub-profile v'_{-i} as follows: for each player $k \neq i$ and each copy j, $v_k'(j) \stackrel{\text{def}}{=} y$. Let us now compare player i's utilities in the outcomes $\mathsf{Vickrey}(v_i^*, v'_{-i})$ and $\mathsf{Vickrey}(v_i, v'_{-i})$.

In $\mathsf{Vickrey}(v_i^*, v'_{-i})$, i wins at least j^* copies, because $v_i^*(1) \geq \cdots \geq v_i^*(j^*) > y$; moreover, he pays y for each such copy, because y is the value that every other player reports, in v'_{-i}, for every single copy of the good. By contrast, in $\mathsf{Vickrey}(v_i, v'_{-i})$, i wins exactly $j^* - 1$ copies, because $v_i(1) \geq \cdots \geq v_i(j^* - 1) \geq v_i^*(j^* - 1) > y$ and $v_i(j^*) < y$; moreover, he again pays y for each of them. Thus, to prove that

$$U_i\big(\theta_i, \mathsf{Vickrey}(v_i^*, v'_{-i})\big) < U_i\big(\theta_i, \mathsf{Vickrey}(v_i, v'_{-i})\big) \ ,$$

it suffices to point out that, for each copy $j \geq j^*$ that i wins in $\mathsf{Vickrey}(v_i^*, v'_{-i})$, i's true value is $\theta_i'(j) \leq \theta_i'(j^*) \leq v_i(j^*) + \varepsilon < y$. This ends the proof of (1.11) in case (a).

In case (b), we instead let j^* be the last coordinate $j \in [m]$ such that $v_i^*(j) < v_i(j)$. An absolutely symmetrical argument shows that (1.11) also holds for this case.

In sum, Theorem 1.8 holds. ■

1.D Proof of Corollary 1.10

Let $v \in \mathsf{UD}(K)$ be any profile of undominated strategies, and $A = (A_0, A_1, \ldots, A_n)$ represent the allocation in the outcome $\mathsf{Vickrey}(v)$, where each player i receives A_i copies of the goods, and A_0 is the number of unallocated copies. For any $\theta \in K$, let $B = (B_0, B_1, \ldots, B_n)$ represent the allocation that maximizes social welfare under θ, i.e., $B = \arg\max_{B \in \mathcal{A}} \big\{ \sum_{i=1}^n \sum_{\ell=1}^{B_i} \theta_i(\ell) \big\}$. Then,

$$\mathrm{SW}(\theta, \mathsf{Vickrey}(v)) = \sum_{i=1}^n \sum_{\ell=1}^{A_i} \theta_i(\ell) \stackrel{(1)}{\geq} \sum_{i=1}^n \sum_{\ell=1}^{A_i} \big(K_i^\top(\ell) - \delta\big) \stackrel{(2)}{\geq} \sum_{i=1}^n \sum_{\ell=1}^{A_i} \big(v_i(\ell) - \delta\big)$$

$$\stackrel{(3)}{\geq} \Big(\sum_{i=1}^n \sum_{\ell=1}^{A_i} v_i(\ell)\Big) - m\delta \stackrel{(4)}{\geq} \Big(\sum_{i=1}^n \sum_{\ell=1}^{B_i} v_i(\ell)\Big) - m\delta \stackrel{(5)}{\geq} \Big(\sum_{i=1}^n \sum_{\ell=1}^{B_i} K_i^\perp(\ell)\Big) - m\delta$$

$$\stackrel{(6)}{\geq} \Big(\sum_{i=1}^n \sum_{\ell=1}^{B_i} (\theta_i(\ell) - \delta)\Big) - m\delta \stackrel{(7)}{\geq} \mathrm{MSW}(\theta) - 2m\delta.$$

Above

- Inequality (1) holds because $\theta \in K$, and thus $\theta_i(\ell) \geq K_i^\perp(\ell) \geq K_i^\top(\ell) - \delta$;
- Inequality (2) holds by Theorem 1.8;
- Inequality (3) holds because we have only m copies of the good: $\sum_{i=1}^n A_i \leq m$;
- Inequality (4) holds because the Vickrey mechanism maximizes social welfare

with respect to v, and thus, relative to v, $(A_0, \ldots, A_n)$ is no worse than any other allocation, and in particular no worse than $(B_0, \ldots, B_n)$;

- Inequality (5) holds again by Theorem 1.8;
- Inequality (6) holds because $\theta \in K$, and thus $\theta_i(\ell) \leq K_i^{\top}(\ell) \leq K_i^{\perp}(\ell) + \delta$; and
- Inequality (7) holds because of the definition of $\text{MSW}(\theta)$ and the fact that $\sum_{i=1}^{n} B_i \leq m$. ■

1.E Proof of Theorem 1.14

1.E.1 A Structural Lemma

The following lemma applies to *all* finite mechanisms, including those that allow players to report sets of valuations, or anything else. (Indeed, the revelation principle no longer holds for mechanisms that are not dominant-strategy or ex-post Nash. Thus, we must be able to deal with general mechanisms with arbitrary strategy spaces.)

Lemma 1.18. *Let M be a finite mechanism and i a player, let $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_m)$ be two valuations in Θ_i such that $x_j > y_j$ for all copies $j \in [m]$, and let K_i and $\widetilde{K}_i$ be two candidate sets for i such that,*

$$\forall\, t \in \{0, 1, \ldots, m\} \quad (x_1, \ldots, x_t, y_{t+1}, \ldots, y_m) \in K_i \cap \widetilde{K}_i.^{16} \tag{1.12}$$

Then, for every $\varepsilon > 0$, there are mixed strategies $\sigma_i \in \Delta(\mathsf{UD}_i(K_i))$ and $\widetilde{\sigma}_i \in \Delta(\mathsf{UD}_i(\widetilde{K}_i))$ such that, for all $s_{-i} \in S_{-i}$ and all $j \in [m]$,

$$\left| M_{i,j}^{\mathsf{A}}(\sigma_i, s_{-i}) - M_{i,j}^{\mathsf{A}}(\widetilde{\sigma}_i, s_{-i}) \right| < \varepsilon \ .^{17}$$

Proof. First of all, it is simple to see (but anyway proved in Appendix 1.F) that for every finite mechanism, the set of undominated strategies of a Knightian player is always non-empty. Therefore, the sets $\mathsf{UD}_i(K_i)$ and $\mathsf{UD}_i(\widetilde{K}_i)$ are both non-empty. If there exists a common (pure) strategy $s_i \in \mathsf{UD}_i(K_i) \cap \mathsf{UD}_i(\widetilde{K}_i)$, then setting $\sigma_i = \widetilde{\sigma}_i = s_i$ proves Lemma 1.18. Therefore, let us assume in the rest of the proof that $\mathsf{UD}_i(K_i)$ and $\mathsf{UD}_i(\widetilde{K}_i)$ are *totally disjoint*.

Let s_i be a pure strategy in $\mathsf{UD}_i(K_i)$. Then, $\mathsf{UD}_i(K_i) \cap \mathsf{UD}_i(\widetilde{K}_i) = \varnothing$ implies that $s_i \notin \mathsf{UD}_i(\widetilde{K}_i)$. By definition, $s_i \notin \mathsf{UD}_i(\widetilde{K}_i)$ implies the existence of a (possibly mixed) strategy $\widetilde{\sigma}_i \in \Delta(\mathsf{UD}_i(\widetilde{K}_i))$ that (weakly) dominates s_i for player i with respect to candidate set $\widetilde{K}_i$. In symbols, as per Definition 1.7, $\widetilde{\sigma}_i \succ_{(i,\widetilde{K}_i)} s_i$.

[16] Recall that all valuations in Θ_i are non-increasing. Our chosen vectors $(x_1, \ldots, x_t, y_{t+1}, \ldots, y_m)$ are indeed non-increasing, because we have $x_j > y_j$ and both x and y are non-increasing.

[17] In fact, Lemma 1.18 can be strengthened to ensure that the prices are close too: namely, $\left| M_i^{\mathsf{P}}(\sigma_i, s_{-i}) - M_i^{\mathsf{P}}(\widetilde{\sigma}_i, s_{-i}) \right| < \varepsilon$. However, this strengthened version of Lemma 1.18 is not needed in order to prove Theorem 1.14.

Next, we argue that

$$\exists\, \tau_i \in \Delta(\mathsf{UD}_i(K_i)) \text{ such that } \tau_i \succ_{(i,K_i)} \widetilde{\sigma}_i \ .^{18} \tag{1.13}$$

Let us write the possibly mixed strategy $\widetilde{\sigma}_i$ as a sum of pure ones, $\widetilde{\sigma}_i = \sum_{t\in X} \alpha^{(t)} \widetilde{s}_i^{(t)}$. Here, X is a finite index set, each $\widetilde{s}_i^{(t)}$ is a pure strategy from $\mathsf{UD}_i(\widetilde{K}_i)$, each $\alpha^{(t)} > 0$, and $\sum_{t\in X} \alpha^{(t)} = 1$. Invoking again the disjointedness of $\mathsf{UD}_i(K_i)$ and $\mathsf{UD}_i(\widetilde{K}_i)$, we deduce that $\widetilde{s}_i^{(t)} \notin \mathsf{UD}_i(K_i)$ for each $t \in X$. This implies the existence of a strategy $\tau_i^{(t)} \in \Delta(\mathsf{UD}_i(K_i))$ such that $\tau_i^{(t)} \succ_{(i,K_i)} \widetilde{s}_i^{(t)}$. Thus, by defining $\tau_i \stackrel{\text{def}}{=} \sum_{t\in X} \alpha^{(t)} \tau_i^{(t)}$, we have that τ_i dominates $\widetilde{\sigma}_i$. Thus, (1.13) holds.

Similarly, we could argue that there exists some $\widetilde{\tau}_i \in \Delta(\mathsf{UD}_i(\widetilde{K}_i))$ such that $\widetilde{\tau}_i \succ_{(i,\tilde{K}_i)} \tau_i$. Continuing in this fashion, going back and forth between $\Delta(\mathsf{UD}_i(K_i))$ and $\Delta(\mathsf{UD}_i(\widetilde{K}_i))$, we obtain an infinite chain of (possibly repeating) strategies,

$$\sigma_i^{(1)} \prec_{(i,\tilde{K}_i)} \widetilde{\sigma}_i^{(1)} \prec_{(i,K_i)} \sigma_i^{(2)} \prec_{(i,\tilde{K}_i)} \widetilde{\sigma}_i^{(2)} \prec_{(i,K_i)} \cdots$$

This (weak) dominance chain implies the following utility inequalities: for all $s_{-i} \in S_{-i}$ and all $k \in \mathbb{N}$:

$$\begin{array}{lll} \forall \widetilde{\theta}_i \in \widetilde{K}_i & U_i\big(\widetilde{\theta}_i, M(\sigma_i^{(k)}, s_{-i})\big) & \leq \ U_i\big(\widetilde{\theta}_i, M(\widetilde{\sigma}_i^{(k)}, s_{-i})\big) \\ \forall \theta_i \in K_i & U_i\big(\theta_i, M(\widetilde{\sigma}_i^{(k)}, s_{-i})\big) & \leq \ U_i\big(\theta_i, M(\sigma_i^{(k+1)}, s_{-i})\big) \end{array} \tag{1.14}$$

Next, for every $t \in \{0, 1, \ldots, m\}$, we define

$$z_t \stackrel{\text{def}}{=} (z_{t,1}, z_{t,2}, \ldots, z_{t,m}) \stackrel{\text{def}}{=} (x_1, x_2, \ldots, x_t, y_{t+1}, \ldots, y_m) \in K_i \cap \widetilde{K}_i \ .$$

Choosing $\theta_i = \widetilde{\theta}_i = z_t$ in (1.14), we obtain that for all $s_{-i} \in S_{-i}$ and all $k \in \mathbb{N}$,

$$U_i\big(\widetilde{\theta}_i, M(\sigma_i^{(k)}, s_{-i})\big) \leq U_i\big(\widetilde{\theta}_i, M(\widetilde{\sigma}_i^{(k)}, s_{-i})\big) = U_i\big(\theta_i, M(\widetilde{\sigma}_i^{(k)}, s_{-i})\big) \leq U_i\big(\theta_i, M(\sigma_i^{(k+1)}, s_{-i})\big) \ .$$

Putting together the above inequalities for $k = 1, 2, \ldots$, we get the following infinite and non-decreasing sequence of real numbers (for each $s_{-i} \in S_{-i}$):

$$U_i\big(z_t, M(\sigma_i^{(1)}, s_{-i})\big) \leq U_i\big(z_t, M(\widetilde{\sigma}_i^{(1)}, s_{-i})\big) \leq U_i\big(z_t, M(\sigma_i^{(2)}, s_{-i})\big) \leq \cdots$$

This sequence is upperbounded by $x_1 + \cdots + x_m$. (Indeed, $z_{t,l} \leq x_l$ for each l. So, the i's valuation is at most $x_1 + \cdots + x_m$, while i's price is non-negative.) Thus, because of the Bolzano-Weierstrass theorem (i.e., because any non-decreasing and upper bounded sequence of real numbers must converge), for every $s_{-i} \in S_{-i}$ and $t \in \{0, 1, \ldots, m\}$, letting $D \stackrel{\text{def}}{=} \min_{l\in[m]}\{x_l - y_l\}$, there must exist some $H_\varepsilon^{(s_{-i},t)} \in \mathbb{N}$ such that, for all $k > H_\varepsilon^{(s_{-i},t)}$:

$$\begin{aligned} &\Big| \Big(\textstyle\sum_{j\in[m]} M_{i,j}^{\mathsf{A}}(\sigma_i^{(k)}, s_{-i}) \big(\sum_{l=1}^{j} z_{t,l} \big) - M_i^{\mathsf{P}}(\sigma_i^{(k)}, s_{-i}) \Big) \\ &- \Big(\textstyle\sum_{j\in[m]} M_{i,j}^{\mathsf{A}}(\widetilde{\sigma}_i^{(k)}, s_{-i}) \big(\sum_{l=1}^{j} z_{t,l} \big) - M_i^{\mathsf{P}}(\widetilde{\sigma}_i^{(k)}, s_{-i}) \Big) \Big| \end{aligned}$$

[18] Note that, while we have only defined what it means for a *pure* strategy to be dominated by a possibly mixed one, the definition trivially extends to the case of dominated strategies that are *mixed*, as is the case in "$\tau_i \succ_{(i,K_i)} \widetilde{\sigma}_i$" in (1.13).

$$= \left|U_i\big(z_t, M(\sigma_i^{(k)}, s_{-i})\big) - U_i\big(z_t, M(\widetilde{\sigma}_i^{(k)}, s_{-i})\big)\right| \leq \frac{\varepsilon D}{4} \ . \qquad (1.15)$$

At this point, we invoke the finiteness of the mechanism in order to define the following maximum value:

$$H_\varepsilon \stackrel{\text{def}}{=} \max\left\{H_\varepsilon^{(s_{-i},t)} : s_{-i} \in S_{-i}, t \in \{0,1,\ldots,m\}\right\} \in \mathbb{N} \ .$$

As a result, (1.15) holds for every $k > H_\varepsilon$, $s_{-i} \in S_{-i}$, and $t \in \{0,1,\ldots,m\}$. We now claim that, by picking an arbitrary $k > H_\varepsilon$, the strategies $\sigma_i^{(k)}$ and $\widetilde{\sigma}_i^{(k)}$ must be the two 'sufficiently close' strategies we are looking for.

To prove this, consider an arbitrary strategy subprofile $s_{-i} \in S_{-i}$ and an integer $t \in [m]$, and apply (1.15) twice, once for t and once for $t-1$. Combining the resulting two inequalities and applying the triangle inequality, we have:[19]

$$\begin{aligned}
\frac{\varepsilon D}{2} \geq \Bigg| \quad & \Big(\textstyle\sum_{j\in[m]} M_{i,j}^{\mathsf{A}}(\sigma_i^{(k)}, s_{-i})\big(\sum_{l=1}^{j} z_{t,l}\big) - M_i^{\mathsf{P}}(\sigma_i^{(k)}, s_{-i})\Big) \\
& - \Big(\textstyle\sum_{j\in[m]} M_{i,j}^{\mathsf{A}}(\widetilde{\sigma}_i^{(k)}, s_{-i})\big(\sum_{l=1}^{j} z_{t,l}\big) - M_i^{\mathsf{P}}(\widetilde{\sigma}_i^{(k)}, s_{-i})\Big) \\
& - \Big(\textstyle\sum_{j\in[m]} M_{i,j}^{\mathsf{A}}(\sigma_i^{(k)}, s_{-i})\big(\sum_{l=1}^{j} z_{t-1,l}\big) - M_i^{\mathsf{P}}(\sigma_i^{(k)}, s_{-i})\Big) \\
& + \Big(\textstyle\sum_{j\in[m]} M_{i,j}^{\mathsf{A}}(\widetilde{\sigma}_i^{(k)}, s_{-i})\big(\sum_{l=1}^{j} z_{t-1,l}\big) - M_i^{\mathsf{P}}(\widetilde{\sigma}_i^{(k)}, s_{-i})\Big) \quad \Bigg| \\
= \Bigg| & \Big(\textstyle\sum_{j=t}^{m} M_{i,j}^{\mathsf{A}}(\sigma_i^{(k)}, s_{-i})\big(x_t - y_t\big)\Big) - \Big(\textstyle\sum_{j=t}^{m} M_{i,j}^{\mathsf{A}}(\widetilde{\sigma}_i^{(k)}, s_{-i})\big(x_t - y_t\big)\Big)\Bigg| \\
= (x_t - y_t)\Bigg| & \Big(\textstyle\sum_{j=t}^{m} M_{i,j}^{\mathsf{A}}(\sigma_i^{(k)}, s_{-i})\Big) - \Big(\textstyle\sum_{j=t}^{m} M_{i,j}^{\mathsf{A}}(\widetilde{\sigma}_i^{(k)}, s_{-i})\Big)\Bigg| \ ,
\end{aligned}$$

which further implies, using $x_t - y_t \geq D > 0$, that

$$\left|\Big(\textstyle\sum_{j=t}^{m} M_{i,j}^{\mathsf{A}}(\sigma_i^{(k)}, s_{-i})\Big) - \Big(\textstyle\sum_{j=t}^{m} M_{i,j}^{\mathsf{A}}(\widetilde{\sigma}_i^{(k)}, s_{-i})\Big)\right| \leq \frac{\varepsilon}{2} \ . \qquad (1.16)$$

Let us now use (1.16) to argue that the following set of inequalities hold:

$$\forall t \in [m] \quad \left|M_{i,t}^{\mathsf{A}}(\sigma_i^{(k)}, s_{-i}) - M_{i,t}^{\mathsf{A}}(\widetilde{\sigma}_i^{(k)}, s_{-i})\right| \leq \varepsilon \ . \qquad (1.17)$$

Indeed, for $t = m$, (1.17) can be derived by plugging $t = m$ into (1.16). Else, for each $t \in \{1,2,\ldots,m-1\}$, we apply (1.16) twice, once for t and once for $t+1$, and again combine the resulting inequalities with the triangle inequality to deduce (1.17).

This completes the proof of Lemma 1.18. □

1.E.2 Deducing Theorem 1.14 from Lemma 1.18

Because $\mathbb{K}$ is ε-basic, let V and W be the corresponding subsets of reals from Definition 1.12. Denote by $a, b \in V \cap W$ any two disjoint reals in $V \cap W$ such that $a > b$. For each player i, consider the following two δ-approximate candidate

[19] That is, $|a-b| \leq \varepsilon$ and $|c-d| \leq \varepsilon$ imply $|(a-b)-(c-d)| \leq 2\varepsilon$.

sets

$$\widetilde{K}_i \stackrel{\text{def}}{=} \{\theta_i \in \Theta_i \mid \forall j, \theta_i(j) \in V\} \quad \text{and} \quad \widetilde{K}'_i \stackrel{\text{def}}{=} \{\theta_i \in \Theta_i \mid \forall j, \theta_i(j) \in W\} \ ,$$

and according to the ε-basic assumption on $\mathbb{K}_i$, we have $\widetilde{K}_i, \widetilde{K}'_i \in \mathbb{K}_i$. Next, consider the following two valuations that belong to Θ_i for every i:

$$x = (a, a, \ldots, a) \quad \text{and} \quad y = (b, b, \ldots, b) \ .$$

It is simple to verify that x, y, $\widetilde{K}_i$ and $\widetilde{K}'_i$ satisfy the hypothesis of Lemma 1.18 (or more precisely, (1.12)). Thus, for any $\varepsilon' > 0$, the following holds:

$$\begin{gathered}\text{for all } i \in [n] \text{ there exist } \sigma_i \in \Delta(\mathsf{UD}_i(\widetilde{K}_i)) \text{ and } \sigma'_i \in \Delta(\mathsf{UD}_i(\widetilde{K}'_i)) \text{ such that} \\ \forall s_{-i} \in S_{-i} \ \ \forall j \in [m] \quad \left|M^{\mathsf{A}}_{i,j}(\sigma_i, s_{-i}) - M^{\mathsf{A}}_{i,j}(\sigma'_i, s_{-i})\right| < \varepsilon' \ .\end{gathered} \tag{1.18}$$

Consider the allocation of M under the strategy profile $\sigma' = (\sigma'_1, \sigma'_2, \ldots, \sigma'_n)$. Because there are m copies of the good, there ought to be one player who, in expectation, receives no more than $\frac{m}{n}$ copies. Without loss of generality, let him be player 1: that is, $\sum_{j=1}^{m} j \cdot M^{\mathsf{A}}_{1,j}(\sigma'_1, \ldots, \sigma'_n) \leq \frac{m}{n}$. Thus, by (1.18) and multiple applications of the triangle inequality, we have

$$\sum_{j=1}^{m} j \cdot M^{\mathsf{A}}_{1,j}(\sigma_1, \sigma'_{-1}) \leq \frac{m}{n} + \varepsilon' m^2 \ .$$

By averaging, there exists a *pure* strategy profile $s = (s_1, s_{-1})$ in the support of (σ_1, σ'_{-1}) satisfying

$$\sum_{j=1}^{m} j \cdot M^{\mathsf{A}}_{1,j}(s_1, s_{-1}) \leq \frac{m}{n} + \varepsilon' m^2 \ . \tag{1.19}$$

Now let

$$K \stackrel{\text{def}}{=} (K_1, \ldots, K_n), \text{ where } K_i \stackrel{\text{def}}{=} \begin{cases} \widetilde{K}_1 & \text{if } i = 1 \\ \widetilde{K}'_i & \text{if } i = 2, \ldots, n \end{cases}$$

$$\theta \stackrel{\text{def}}{=} (\theta_1, \ldots, \theta_n), \text{ where } \theta_i \stackrel{\text{def}}{=} \begin{cases} \big(\max V, \ldots, \max V\big) & \text{if } i = 1 \\ \big(\min W, \ldots, \min W\big) & \text{if } i = 2, \ldots, n. \end{cases}$$

Because we know that $(\sigma_1, \sigma'_{-1}) \in \Delta(\mathsf{UD}_1(K_1)) \times \cdots \times \Delta(\mathsf{UD}_n(K_n))$ from (1.18), we deduce that $s \in \mathsf{UD}(K)$. It is also obvious that $\theta \in K$ and $\mathrm{MSW}(\theta) = m \cdot \max V$.

Next, we show that s, K, and θ satisfy the desired inequality of Theorem 1.14. Indeed,

$$\begin{aligned}\mathbb{E}\Big[\mathrm{SW}\big(\theta, M(s)\big)\Big] &\stackrel{(*)}{\leq} \Big(\frac{m}{n} + \varepsilon' m^2\Big) \cdot \max V + \Big(m - \frac{m}{n} - \varepsilon' m^2\Big) \cdot \min W \\ &= m \max V - \Big(m - \frac{m}{n} - \varepsilon' m^2\Big)(\max V - \min W) \\ &\leq m \max V - \Big(m - \frac{m}{n} - \varepsilon' m^2\Big)(2\delta - \frac{\varepsilon}{m}) \\ &= \mathrm{MSW}(\theta) - 2\delta m\big(1 - 1/n\big) + 2\delta\varepsilon' m^2 + \frac{\varepsilon}{m}\Big(m - \frac{m}{n} - \varepsilon' m^2\Big)\end{aligned}$$

$$\leq \mathrm{MSW}(\theta) - 2\delta m\big(1 - 1/n\big) + \varepsilon + 2\delta\varepsilon' m^2 - \frac{\varepsilon}{n} \ .$$

Above, inequality $(*)$ holds because, when θ is the true-valuation profile, the value for each copy of the good is $\max V$ for player 1, and is $\min W$ for every player other than player 1. However, in the outcome $M(s)$, owing to (1.19), in expectation player 1 can receive at most $\frac{m}{n} + \varepsilon' m^2$ copies of the good.

Finally, noticing that $\varepsilon' > 0$ can be arbitrarily small, we can choose ε' to satisfy $2\delta\varepsilon' m^2 - \frac{\varepsilon}{n} \leq 0$. This implies that $\mathbb{E}\big[\mathrm{SW}\big(\theta, M(s)\big)\big] \leq \mathrm{MSW}(\theta) - 2\delta m\big(1 - 1/n\big) + \varepsilon$. Therefore, Theorem 1.14 holds. ■

1.F The Set of Undominated Strategies is Non-Empty

It is trivial to see that, no matter what candidate set K_i a player i may have, $\mathsf{UD}_i(K_i)$ is non-empty in the Vickrey mechanism. In fact, Theorem 1.8 implies that $\mathsf{UD}_i(K_i)$ includes at least all the valuations in K_i.

Below, we argue that $\mathsf{UD}_i(K_i)$ is also always non-empty for all finite mechanisms.

Fact 1.19. *Let M be a finite mechanism, i a player, and K_i a candidate set of i. Then, $\mathsf{UD}_i(K_i) \neq \varnothing$.*

Proof. Let $S_i = \{s_1, \ldots, s_t\}$ be the finite pure-strategy set of player i. We proceed by contradiction. Suppose that every strategy in S_i is (weakly) dominated, with respect to K_i, by some strategy in $\Delta(S_i)$. Then, in particular, s_1 is dominated. Thus, there exists a mixed strategy $\sum_{k=1}^{t} \alpha_k s_k \in \Delta(S_i)$ such that

$$s_1 \prec_{(i,K_i)} \sum_{k=1}^{t} \alpha_k s_k \ , \tag{1.20}$$

where $\alpha \in \Delta \stackrel{\text{def}}{=} \{x \in [0,1]^t \mid \sum_{k=1}^{t} x_k = 1\}$. Notice that, by condition (2) in Definition 1.7, we cannot have $s_1 \prec_{(i,K_i)} s_1$. Therefore, we must have $\alpha_1 < 1$. Now, we simplify (1.20) by subtracting $\alpha_1 s_1$ on both sides and rescaling:

$$s_1 \prec_{(i,K_i)} \sum_{k=2}^{t} \frac{\alpha_k}{1 - \alpha_1} s_k \ . \tag{1.21}$$

Next, since s_2 is dominated, let it be dominated by $\sum_{k=1}^{t} \beta_k s_k$. In symbols,

$$s_2 \prec_{(i,K_i)} \sum_{k=1}^{t} \beta_k s_k \tag{1.22}$$

for some $\beta \in \Delta$. By substituting (1.21) into (1.22), we can rewrite (1.22) as

$$s_2 \prec_{(i,K_i)} \sum_{k=2}^{t} \beta'_k s_k \tag{1.23}$$

for some $\beta' \in \Delta$ such that $\beta'_1 = 0$. Again, by subtracting $\beta'_2 s_2$ on both sides and rescaling, we obtain

$$s_2 \prec_{(i,K_i)} \sum_{k=3}^{t} \beta''_k s_k \ , \tag{1.24}$$

for some $\beta'' \in \Delta$ such that $\beta''_1 = \beta''_2 = 0$. We substitute (1.24) into (1.21), and obtain

$$s_1 \prec_{(i,K_i)} \sum_{k=3}^{t} \alpha'_k s_k \ ,$$

for some $\alpha' \in \Delta$ such that $\alpha'_1 = \alpha'_2 = 0$.

This process, similar to Gaussian elimination in linear systems, can be continued until we obtain $s_k \prec_{(i,K_i)} s_t$ for every $k = 1, \ldots, t-1$. Thus, s_t must be an undominated strategy for player i, contradicting the hypothesis that $\mathsf{UD}_i(K_i) = \varnothing$. □

1.G The Work of Lopomo, Rigotti, and Shannon

Their Model. In order to "strip away issues pertaining to higher order beliefs and strategic uncertainty", Lopomo, Rigotti, and Shannon [99] focus on single-player mechanisms. Thus, so do we when recalling their work.

In their model, *true state of the world* comprises all the information the player is uncertain about, and the player's utility function, U, maps $O \times T \times S$ to $\mathbb{R}$, where

(a) O is the set of all possible outcomes,

(b) $T \stackrel{\text{def}}{=} [0,1]$ is the set of all possible player types, and

(c) S is the set of all possible true states of the world.

When the player's type is $t \in T$, the only information the player has about the true state of the world $s \in S$ is that s is drawn from a distribution $\Pi(t)$ in $\Delta(S)$.

In their model, the player knows his own type $t \in T$, and a mechanism knows the true state of the world $s \in S$. The player is allowed to report just his own type, and then a mechanism chooses an outcome based not only on this report, but also on the true state: that is, each mechanism ϕ is a function $\phi\colon T \times S \to O$.

By contrast, in our auction setting, a mechanism chooses an outcome solely based on the players' reports. Indeed, since each player is uncertain about his own valuation, the true state of the world should include the true valuation profile θ^*, and if a mechanism knew θ^*, then it would be trivial to choose an outcome of maximum social welfare.

In their Knightian setting, they provide a general notion of a dominant-strategy mechanism, *optimal incentive compatibility (optimal IC)*, and a very restrictive notion of a dominant-strategy mechanism, *ex-post incentive compatibility (ex-post IC)*. Formally, a mechanism ϕ is

- optimal IC if, $\forall t \in T$, $\forall \sigma \in \Delta(T)$, and $\forall \pi \in \Pi(t)$: $\mathbb{E}_{s\sim\pi}\big[U(\phi(t,s),t,s)\big] \geq \mathbb{E}_{s\sim\pi}\big[\mathbb{E}_{\theta\sim\sigma}\big[U(\phi(\theta,s),t,s)\big]\big]$, and

- ex-post IC if, $\forall t, \theta \in T$ and $\forall s \in S$: $U(\phi(t,s),t,s) \geq U(\phi(\theta,s),t,s)$.

Their First Theorem. They assume that, for every type $t \in T$, there exists a neighborhood $N(t) \subset T$ such that, for all continuous functions $g\colon S \to \mathbb{R}$,

$$\text{if } \int_S g(s)d\pi = 0 \text{ for every } \pi \in \bigcap_{t' \in N(t)} \Pi(t'), \text{ then } g = 0.$$

Under this assumption, their first theorem shows that every optimal IC mechanism satisfying an additional technical condition (i.e., ex-post cyclical monotonicity) must be ex-post IC.

Therefore, their first theorem has the same spirit of our Theorem 1.6. In both theorems, some form of overlapping of a player's possible 'belief/knowledge sets' implies that every dominant-strategy mechanism must be of a very restrictive form. However, due to the differences in models and assumptions, it is unclear whether our already simple proof of Theorem 1.6 can be more simply derived from theirs. Even ignoring all other differences, there cannot be any subjective map from their type space to ours. In their case, a player's type space (i.e., $T = [0,1]$) has the cardinality of the continuum. In our case, the type space of a given player i (i.e., $\mathbb{K}_i$) may have the cardinality of the power set of the continuum.

Chapter 2

Knightian Self Uncertainty in the VCG Mechanism for Unrestricted Combinatorial Auctions

This chapter is based on the result published in [43].

We study the social welfare performance of the VCG mechanism in the well-known and challenging model of self uncertainty initially put forward by Frank H. Knight and later formalized by Truman F. Bewley. Namely, the only information that each player i has about his own true valuation consists of a set of distributions, from one of which i's valuation has been drawn.

We assume that each player knows his true valuation up to an additive inaccuracy δ, and study the social welfare performance of the VCG mechanism relative to $\delta > 0$. In this paper, we focus on the social welfare performance of the VCG mechanism in *unrestricted combinatorial auctions*,[1] both in undominated strategies and regret-minimizing strategies. Denote by MSW the maximum social welfare.

Our first theorem proves that, in an n-player m-good combinatorial auction, the VCG mechanism may produce outcomes whose social welfare is $\leq \text{MSW} - \Omega(2^m\delta)$, even when $n = 2$ and each player chooses an *undominated strategy*. We also geometrically characterize the set of undominated strategies in this setting.

Our second theorem shows that the VCG mechanism performs well in *regret-minimizing strategies*: the guaranteed social welfare is $\geq \text{MSW} - 2\min\{m, n\}\delta$ if

[1]We acknowledge that the VCG mechanism admits computational-complexity issues [37, 57]; in this paper we choose to focus on how the Knightian players rationally behave in VCG ignoring such complexity issues. It turns out this is already a very non-trivial question to tackle, not to say that in practice it is also interesting to study the VCG mechanism on selling 10 goods to 10 players, which is computationally tractable on a modern PC.

each player chooses a pure regret-minimizing strategy, and $\geq \text{MSW} - O(n^2\delta)$ if mixed strategies are allowed.

2.1 Introduction

2.1.1 Theorem 2.1: VCG Auction in Undominated Strategies

In an *(unrestricted) combinatorial auction* of n players and m goods, the set of possible allocations $\mathcal{A}$ consists of all possible partitions of $[m]$ (the set of m goods) into $1+n$ subsets $(A_0, A_1, \ldots, A_n)$, where A_0 is the (possibly empty) set of unassigned goods and A_i is the (possibly empty) set of goods assigned to player i. Given an allocation $A = (A_0, A_1, \ldots, A_n)$, player i has valuation $\theta_i^*(A_i) \in \mathbb{R}_{\geq 0}$ if $A_i \neq \varnothing$ and 0 if $A_i = \varnothing$.[2]

In a *Knightian* (unrestricted) combinatorial auction, the only information i has about the true valuation profile θ_i^* lies in K_i. Letting $K_i(S) := \{\theta_i(S)\}_{\theta_i \in K_i}$, we say that K_i is *δ-approximate* if $\sup K_i(S) - \inf K_i(S) \leq \delta$ for all non-empty $S \subseteq [m]$. We prove that,

Theorem 2.1 (Informal)**.** *In a δ-approximate combinatorial Knightian auction with $n \geq 2$ players and m goods, the VCG cannot, in undominated strategies, guarantee social welfare greater than* $\text{MSW} - (2^{m+1} - 5)\delta$.

(The formal statement and proof of Theorem 2.1 can be found in Section 2.4.)

In fact, in this case we have been able to characterize UD_i, the set undominated strategies of a player i. This time, UD_i is *much larger* than K_i. Player i may choose an (almost arbitrary) constant fraction of the coordinates $\mathcal{S} \subseteq 2^{[m]}$, and deviate from $K_i(S)$ by an additive factor as large as $\Theta(2^m\delta)$ for all $S \in \mathcal{S}$. This strategy remains undominated for player i!

Perhaps more surprisingly, characterizing the undominated strategies of the VCG in unrestricted combinatorial auctions is much harder. Indeed, even *describing* the resulting set UD_i is challenging. (Indeed, we resort to geometry in order to describe it in a succinct way.)

Theorem 2.1 is somewhat disconcerting, if we feel that the VCG should always be the mechanism of choice for getting good social welfare, even when the players are Knightian, and even when the players are belief-free. But there are other solution concepts to consider.

[2] All of our results for combinatorial auctions actually also hold even under a mild restriction on the players' valuation, namely, when they are *set-monotone* (or with *free disposal*): that is, $\theta_i(S) \leq \theta_i(T)$ whenever $S \subseteq T$.

2.1.2 Theorem 2.2: VCG Auctions in Regret-Minimizing Strategies

So far we have analyzed the VCG under all solution concepts traditionally used in private-value and belief-free auctions of incomplete information, assuming that the players are *utility maximizers*. We now analyze the VCG's performance in Knightian auctions in regret-minimizing strategies. The notion of a regret-minimizing strategy naturally extends to the Knightian setting. Informally, the regret of a strategy s_i of a player i is the maximum difference, taken over all possible strategy choices of i's opponents and all possible choices of θ_i in K_i, between the utility i gets by playing s_i and the utility he gets by best responding to those choices. A regret-minimizing player i chooses strategies that minimize his regret.

With respect to *pure* regret-minimizing strategies, we prove the following

Theorem 2.2 (Informal)**.** *In a δ-approximate combinatorial Knightian auction with n players and m goods, the VCG guarantees social welfare $\geq$ MSW $- 2\min\{n, m\}\delta$ in pure regret-minimizing strategies.*

(We prove Theorem 2.2 in Section 2.5.)

That is, in combinatorial Knightian auctions, the performance of the VCG in (pure) regret minimizing strategies is absolutely stellar. Theorem 2.2 is less intuitive than it seems, because in a combinatorial, Knightian, VCG auction it is not obvious which strategies are regret-minimizing. Consider a player i who (1) happens to know that his true valuation for some subset of the good S lies in some interval $[x_S, x_S + \delta]$, and (2) chooses to play a pure, regret-minimizing strategy v_i. At first glance, it would appear that $v_i(S)$ should coincide with the center of the interval, that is, $v_i(S) = x_S + \delta/2$. In reality, however, $v_i(S)$ need not even belong to the interval $[x_S, x_S + \delta]$. Nevertheless, we prove that it cannot lie too far from the interval.

Mixed Strategies. For simplicity, Theorem 2.2 has been stated for pure strategies. Indeed, as shown in Appendix 2.D.1, significant difficulties arise when dealing with mixed strategies. For instance, we must deal with the fact that a regret-minimizing mixed strategy can, in expectation and for *each* subset S, be *arbitrarily* far away from $K(S)$! However, Theorem 2.2 essentially continues to hold when allowing *mixed* strategies, but with a worse bound. Roughly, $\min\{n, m\}$ is replaced by n^2 (or even $n \log n$ if the valuations are set-monotone).[3]

[3]That is, $v_i(S) \leq v_i(T)$ for all $S \subseteq T \subseteq [m]$, all i, and all $v_i \in \Theta_i$. The interested reader can consult Appendix 2.D for the mixed-strategy version of Theorem 2.2.

2.1.3 The Meaningfulness of Theorem 2.2 and a Rationality Bridge Lemma

In principle, Theorem 2.2 or any other implementation in regret-minimizing strategies would be irrelevant, in the exact-valuation or in the Knightian setting, if at least one player is not a regret minimizer but a utility maximizer. However, we show that a separate lemma relating these two basic models of rationality *in all games* (with or without Knightian players), indicates that Theorem 2.2 may retain some meaningfulness. Let us explain.

- A utility-maximizing player $\mathcal{U}$ eliminates all his dominated strategies to compute his set of undominated ones, UD. Notice that $\mathcal{U}$ cannot further refine UD based on utility maximization alone. If UD consists of a single strategy s (necessarily a dominant one), then $\mathcal{U}$ of course chooses s. But:
  if UD *contains multiple strategies, which ones might* $\mathcal{U}$ *prefer?*
- A regret-minimizing player $\mathcal{R}$ eliminates all his non regret-minimizing strategies so as to compute his set of regret-minimizing strategies, RM. He might even continue this process k times, until he is satisfied or no further elimination is possible. Let us denote the final set of strategies he obtains this way by RM^k. If RM^k consists of a single strategy s, he of course chooses s. But:
  if RM^k *contains multiple strategies, which ones might* $\mathcal{R}$ *prefer?*

A possible answer is that, when he is no longer able to apply his 'favorite way of reasoning', even a die-hard utility maximizer $\mathcal{U}$ will resort to regret minimization to refine UD, and even a die-hard regret minimizer $\mathcal{R}$ will resort to utility maximization to refine RM^k. In principle, the two final sets of strategies obtained by such different refinement procedures could be vastly different. Our mentioned lemma, however, guarantees that they coincide.

Abusing notation a bit, consider UD and RM also to be 'operators' acting on sets of strategies. In this case $\mathsf{UD}(\mathsf{UD}) = \mathsf{UD}$, while $\mathsf{RM}^2 \stackrel{\text{def}}{=} \mathsf{RM}(\mathsf{RM})$ may be a strict subset of RM. Then, our structural lemma can be expressed as follows.

Lemma 2.3 (Rationality Bridge Lemma, proved in Chapter 3)**.**

The set of strategies obtained after applying, in arbitrary order, k *times the operator* RM *and at least once the operator* UD *coincides with* $\mathsf{RM}^k \cap \mathsf{UD}$*.*

For instance, $\mathsf{RM}(\mathsf{RM}(\mathsf{UD}(\mathsf{RM}(\mathsf{RM}(\mathsf{UD}))))) = \mathsf{RM}^4(\mathsf{UD}) = \mathsf{RM}^4 \cap \mathsf{UD}$.

A formal statement and proof of the above lemma can be found in Appendix 3.2. Here we wish just to mention the following implication for mechanism design:

For all mechanisms M *and social choice correspondences* f*,*
if M *implements* f *in* RM *strategies or in* UD *strategies,*

then M is automatically guaranteed to implement f also in $\mathsf{RM}(\mathsf{UD})$ *strategies.*[4]

Relative to the VCG, this guarantee implies that Theorem 2.2 continues to hold in $\mathsf{RM}(\mathsf{UD})$ strategies. That is, assuming that the players consider solely pure strategies,

Corollary 2.4. *In a δ-approximate combinatorial Knightian auction with n players and m goods, the VCG guarantees social welfare $\geq \mathrm{MSW} - 2\min\{n, m\}\delta$ (not only when the players are regret minimizers, but also) when the players are utility maximizers who use regret only to break ties.*

(A similar corollary holds for the mentioned mixed-strategy version of Theorem 2.2.)

2.1.4 In Sum

The fact that the VCG is no longer dominant-strategy in Knightian auctions is 'no big loss'. Indeed, no dominant strategy mechanism can do better than assigning the goods at random, even in single-good auctions.

The fact that the VCG has excellent, and indeed essentially optimal, social-welfare performance in undominated strategies in multi-unit (and thus also in single-good) Knightian auctions demonstrates the wide relevance of the VCG.

The fact that the social-welfare performance of the VCG in combinatorial Knightian auctions is extremely poor in undominated strategies is just another hard fact of life. However, per the Rationality Bridging Lemma, once we assume that even die-hard utility maximizers resort to regret minimization when they are forced to break ties, then the VCG continues to be *the* mechanism of choice for good social welfare, even in the Knightian setting and in unrestricted combinatorial auctions.

In sum, as most things classical, the VCG outlives the confines in which it was conceived, and continues to be relevant in new and unforeseen settings.

2.1.5 Roadmap

We discuss the related work in Section 2.2, and provide basic definitions in Section 2.3.

The proof of Theorem 2.1 is very technically involved, so we divide it into four sections. In Section 2.4 we sketch a two-paged proof of a weaker form of Theorem 2.1 to gain intuition. In Appendix 2.A, we state the stronger version of Theorem 2.1 that also includes the geometric characterization of the player's undominated strategies. The full proof is contained in Appendix 2.B and 2.C.

We provide the full proof of the pure strategy version of Theorem 2.2 in Section 2.5, and in Appendix 2.D, we state and prove the mixed-strategy version of Theorem 2.2.

The proof of our structural lemma can be found in Appendix 3.2.

[4]Indeed, for $i = 1$ the bridging lemma implies that $\mathsf{RM}(\mathsf{UD}) = \mathsf{RM} \cap \mathsf{UD} \subseteq \mathsf{RM}$. Of course, to enforce the same guarantee one could just demand that M implements f in $\mathsf{RM} \cup \mathsf{UD}$ strategies, but this is a very strong demand. Indeed $\mathsf{RM} \cup \mathsf{UD}$ could be a much larger set than $\mathsf{RM} \cap \mathsf{UD}$.

2.2 Related Work

Models of Type Uncertainty. The Knightian model was originally proposed by Knight [91] and formalized by Bewley [30].

Knightian players have received much attention in *decision theory*. In particular, Aumann [14], Dubra, Maccheroni and Ok [55], Ok [124], and Nascimento [112] investigate decision with incomplete orders of preferences. Various criteria for selecting a single distribution out of a set of distributions have been studied by Danan [49], Schmeidler [139], Gilboa and Schmeidler [70]. (In fact, Bose, Ozdenoren and Pape [34] and Bodoh-Creed [33] use the model from [70] to study auctions.)

General equilibrium models with incompletely ordered preferences have been considered by Mas-Colell [105], Gale and Mas-Colell [67], Shafer and Sonnenschein [141], and Fon and Otani [64]. More recently, Rigotti and Shannon [135] characterize the set of equilibria in a financial market problem.[5]

Single-player mechanisms, in the Knightian model, for the rent-extraction problem have been studied by Lopomo, Rigotti, and Shannon [99], under two notions of implementation. Namely, (1) when reporting the truth is at least as good as any other strategy, and (2) when reporting the truth is not strictly eliminated in favor of another strategy.[6]

Although they are quite different from the Knightian model, a few other models of player uncertainty should be mentioned. For instance, Milgrom [108], in single-good auctions, studies the revenue difference between second-price and English auctions, when the players do not exactly know their own valuations, but only that they are drawn from a *common* distribution. Sandholm [137] presents an example of an auction (with a non quasi-linear utility function) where a player's valuation is drawn from the uniform distribution over $[0, 1]$, and argues that reporting the expected valuation (i.e., 0.5) is no longer dominant-strategy. Mechanisms for scheduling, when each player knows a single distribution where his type is drawn, have been studied by Porter, Ronen, Shoham and Tennenholtz [132], and by Feige and Tennenholtz [60]. Thompson and Leyton-Brown [157] provide an extensive summary of works on Bayesian self-uncertainties.

Undominated Strategies. Implementations in undominated strategies trace back to Jackson [79, 80]. Although being a well-known solution concept, very few positive results on mechanism design have been achieved so far. Beyond the positive example in [79], Babaioff et al. [21] provide an efficient mechanism for single-value multi-minded auctions, and Abreu and Matsushima [3] achieve perfect revenue in the

[5] A strategy profile is an equilibrium if no player can deviate and strictly benefit no matter which distribution is picked from his set. Notice that such an equilibrium is not a notion of dominance.

[6] Notice that, not envisaging other players, these are not notions of dominance in the Knightian setting. Indeed, even in the exact-valuation setting, the notion of dominance should take into account all possible choices of strategies of the other players.

complete information setting. Our prior work on the Knightian mechanism design is another example [42].

Regret-Minimizing Strategies. Regret-minimizing strategies are also known as regret-minimax strategies. The suggestion of adopting regret-minimizing (a.k.a. regret-minimax) strategies traces back to Savage's reading [138] of the work of Wald [161], and has been axiomatized by Milnor [109]. The notion of regret has been treated differently in different settings. A unified axiomatic characterization of minimax regret has been recently given by Stoye [155].

Mechanisms have also been studied under minimax regret. Linhart and Radner [98] study minimax-regret strategies in a sealed-bid mechanism for bilateral bargaining under complete information. Engelbrecht-Wiggans [58] and Selten [140] analyze first- and second-price sealed-bid auctions by incorporating regret for the bidders. In more general settings, minimax-regret strategies are mostly studied when a player has (Bayesian or set-theoretic) beliefs about his opponents. In particular, Hyafil and Boutilier [76] and Renou and Schlag [134] study two different notions of minimax-regret equilibrium, both coinciding with ours when players do not form beliefs about their opponents. Halpern and Pass [71] propose the solution concept of iterated regret minimization using beliefs.

Regret Minimizers vs. Utility Maximizers. Many empirical studies compare utility maximizers and regret minimizers, see for instance Chorus, Arentze and Timmermans [45], and Hensher, Greene and Chorus [75]. Recently, Engelbrecht-Wiggans and Katok [59] and Filiz and Ozbay [62] provide experimental evidence for regret in first- and second-price auctions. To the best of our knowledge, we are the first to study players who use regret for refining their sets of undominated strategies.

2.3 Classical and Knightian Basic Notions

Recall that, in an auction, the set of possible outcomes is $\Omega \stackrel{\text{def}}{=} \mathcal{A} \times \mathbb{R}^n_{\geq 0}$, where $\mathcal{A}$ denotes the set of all possible allocations of the good(s). If $(A, P) \in \Omega$, we refer to A, $A = (A_0, A_1, \ldots, A_n)$, as the realized allocation, to each P_i as the price charged to player i, to each A_i as the allocation of player i, and to A_0 as the unallocated good(s). A valuation θ_i of a player i is a function, from i's possible allocations to non-negative reals, mapping the empty allocation to 0. The set of all possible valuations for a player i is denoted by Θ_i, and i's true valuation by θ_i^*. We assume quasi-linear utility functions. That is, the utility function U_i of a player i maps a valuation θ_i and an outcome $\omega = (A, P)$ to $U_i(\theta_i, \omega) \stackrel{\text{def}}{=} \theta_i(A_i) - P_i$.

As already said, in a Knightian auction the only information that a player i has about θ_i^* —and the entire profile θ^*— consists of a subset $K_i \subset \Theta_i$, the candidate (valuation) set, guaranteed to contain θ_i^*. A player i has no information or belief about θ_{-i}^* or K_{-i} of his opponents. The true valuations of the players are uncorrelated.

By saying that K is a profile —respectively, a product— of candidate sets, we mean that $K = (K_1, \ldots, K_n)$ —respectively, that $K = K_1 \times \cdots \times K_n$.

Let us now clarify the specific auctions we consider.

δ-approximate Knightian Auctions. Recall that, in an (unrestricted) combinatorial auction, there are n players and m distinct goods. The set of possible allocations $\mathcal{A}$ consists of all possible partitions A of $[m]$ into $1 + n$ subsets, $A = (A_0, A_1, \ldots, A_n)$, where A_0 is the (possibly empty) set of unassigned goods and A_i is the (possibly empty) set of goods assigned to player i. For each player i, $\Theta_i = \{\theta_i : 2^{[m]} \to \mathbb{R}_{\geq 0} \mid \theta_i(\varnothing) = 0\}$.

In an (unrestricted) combinatorial *Knightian* auction, a player i's candidate set K_i is a subset of the above Θ_i. If $S \subset [m]$, then we let $K_i(S) \stackrel{\text{def}}{=} \{\theta_i(S) \mid \theta_i \in K_i\}$. We say that K_i is δ-*approximate* if $\sup K_i(S) - \inf K_i(S) \leq \delta$ for all $S \subseteq [m]$.

A Knightian auction is δ-*approximate* if each candidate set K_i is δ-approximate.

(Possibly Incomplete) Preferences. In a Knightian auction, a utility-maximizing player i with candidate set K_i strictly prefers an outcome ω to an outcome ω' if and only if the following two conditions hold:

(1) $U_i(\theta_i, \omega) \geq U_i(\theta_i, \omega')$ for all $\theta_i \in K_i$ and

(2) $U_i(\theta_i', \omega) > U_i(\theta_i', \omega')$ for some $\theta_i' \in K_i$.

Social welfare. The social welfare of an allocation A, $\mathrm{SW}(A)$, is defined to be $\sum_i \theta_i^*(A_i)$; and the maximum social welfare, MSW, is defined to be $\max_{A \in \mathcal{A}} \mathrm{SW}(A)$. (That is, social welfare and maximum social welfare continue to be defined relative to the players' true valuations θ_i^*, whether or not the players know them exactly.)

More generally, the social welfare of an allocation A relative to a valuation profile θ, $\mathrm{SW}(\theta, A)$, is $\sum_i \theta_i(A_i)$; and the maximum social welfare relative to θ, $\mathrm{MSW}(\theta)$, is $\max_{A \in \mathcal{A}} \mathrm{SW}(\theta, A)$. Thus, $\mathrm{SW}(A) = \mathrm{SW}(\theta^*, A)$ and $\mathrm{MSW} = \mathrm{MSW}(\theta^*)$.

The VCG mechanism. In our auctions, the VCG mechanism (with any tie-breaking rule) maps a profile of valuations $\theta \in \Theta_1 \times \cdots \times \Theta_n$, to an outcome (A, P), where

$A \in \arg\max_{A \in \mathcal{A}} \mathrm{SW}(\theta, A)$ and, for each player i, $P_i = \mathrm{MSW}(\theta_{-i}) - \sum_{j \neq i} \theta(A_i)$.

General mechanisms and strategies. Every auction mechanism M considered in this paper specifies, for each player i, a set S_i. We interchangeably refer to each member of S_i as a pure *strategy/action/report* of i, and similarly, a member of $\Delta(S_i)$ a mixed strategy/action/report of i.[7] After each player i, simultaneously with his opponents, reports a strategy s_i in S_i, M maps the reported strategy profile s to an

[7]Often, in pre-Bayesian settings, the notion of a strategy and that of an action are distinct. Indeed, a strategy s_i of a player i maps the set of all possible types of i to the set of i' possible actions/reports. But since strategies are universally quantified in all relevant definitions of this paper, we have no need to separate (and for simplicity refrain from separating) the notions of strategies and actions.

outcome $M(s) \in \Omega$. If M is probabilistic, then $M(s) \in \Delta(\Omega)$, and, for each player i, $U_i(\theta_i, M(s)) \stackrel{\text{def}}{=} \mathbb{E}_{\omega \sim M(s)}[U_i(\theta_i, \omega)]$.

Note that $S_i = \Theta_i$ in the VCG case, but in general the set S_i is arbitrary.

Knightian undominated strategies. Given a mechanism M, a pure strategy s_i of a player i with a candidate set K_i is *(weakly) undominated*,[8] in symbols $s_i \in \mathsf{UD}_i(K_i)$, if i does not have another (possibly mixed) strategy σ_i such that

(1) $\forall s_{-i}\, \forall \theta_i \in K_i \quad U_i\big(\theta_i, M(\sigma_i, s_{-i})\big) \geq U_i\big(\theta_i, M(s_i, s_{-i})\big)$, and

(2) $\exists s_{-i}\, \exists \theta_i \in K_i \quad U_i\big(\theta_i, M(\sigma_i, s_{-i})\big) > U_i\big(\theta_i, M(s_i, s_{-i})\big)$.

If K is a product/profile of candidate sets, then $\mathsf{UD}(K) \stackrel{\text{def}}{=} \mathsf{UD}_1(K_1) \times \cdots \times \mathsf{UD}_n(K_n)$.[9]

Knightian regret-minimizing strategies. Given a mechanism M, the (maximum) regret of a pure strategy s_i of a player i with candidate set K_i is

$$R_i(K_i, s_i) \stackrel{\text{def}}{=} \max_{\theta_i \in K_i} \max_{s_{-i}} \Big(\max_{s'_i} U_i\big(\theta_i, M(s'_i, s_{-i})\big) - U_i\big(\theta_i, M(s_i, s_{-i})\big)\Big) \ .$$

A pure strategy s_i is *regret-minimizing* among all pure strategies of a player i with a candidate set K_i, in symbols $s_i \in \mathsf{RM}_i^{\mathsf{pure}}(K_i)$, if $R_i(K_i, s_i) \geq R_i(K_i, s'_i)$ for all other pure strategies s'_i of i. We let $\mathsf{RM}^{\mathsf{pure}}(K) \stackrel{\text{def}}{=} \mathsf{RM}_1^{\mathsf{pure}}(K_1) \times \cdots \times \mathsf{RM}_n^{\mathsf{pure}}(K_n)$.

When allowing mixed strategies, the (expected) regret of a (possibly mixed) strategy σ_i of a player i with candidate set K_i is

$$R_i(K_i, \sigma_i) \stackrel{\text{def}}{=} \max_{\theta_i \in K_i} \max_{s_{-i}} \Big(\max_{s'_i} U_i\big(\theta_i, M(s'_i, s_{-i})\big) - \mathbb{E}_{s_i \sim \sigma_i} U_i\big(\theta_i, M(s_i, s_{-i})\big)\Big) \ .$$

We similarly define $\mathsf{RM}_i^{\mathsf{mix}}(K_i)$ as the set of strategies of a player i that minimize regret among all mixed strategies, and let $\mathsf{RM}^{\mathsf{mix}}(K) \stackrel{\text{def}}{=} \mathsf{RM}_1^{\mathsf{mix}}(K_1) \times \cdots \times \mathsf{RM}_n^{\mathsf{mix}}(K_n)$.

2.4 A Weaker Version of Theorem 2.1

It suffices to consider the case where there are $n = 2$ players, because all players other than players 1 and 2 can be made to report 0 on every subset of the goods, and thus not affect the choice of outcome. We now sketch the proof for the following slightly weaker version of Theorem 2.1. (We shall discuss in Appendix 2.A the stronger statement of our theorem as well as a characterization of a player's undominated strategies.)

[8] This is not to be confused with the *strong dominance* that requires the inequality to be strict for all pairs (s_{-i}, θ_i). For this notion in the exact-valuation case, see for instance [66, 95].

[9] As pointed out by Jackson [79] in the exact-valuation case, the general notion of an undominated strategy is more complex. However, for *bounded* mechanisms, the simpler notion above coincides with the general notion, even in the Knightian setting. Since this class of mechanisms includes the VCG and all finite mechanisms, we adopt this simpler notion for this paper.

Theorem 2.1'. *In a combinatorial Knightian auction with 2 players and m goods, consider the VCG with any tie-breaking rule, then there exist products of δ-approximate candidate sets $K = K_1 \times K_2$ and profiles $(v_1, v_2) \in \mathsf{UD}(K)$, such that*

$$\textit{(best-case } \theta\textit{)} \quad \forall \theta \in K_1 \times K_2 \quad \mathrm{SW}\big(\theta, \mathsf{VCG}(v_1, v_2)\big) \leq \mathrm{MSW}(\theta) - (2^m - 3)\delta \quad (2.1)$$

$$\textit{(worst-case } \theta\textit{)} \quad \exists \theta \in K_1 \times K_2 \quad \mathrm{SW}\big(\theta, \mathsf{VCG}(v_1, v_2)\big) \leq \mathrm{MSW}(\theta) - (2^m - 1)\delta. \quad (2.2)$$

Proof Sketch. Let $\pi_1, \ldots, \pi_{2^m-1}$ be any permutation of all non-empty subsets of $[m]$ such that, whenever $j < k$, $\pi_j \not\supseteq \pi_k$.[10] We set $\pi_{2^m} \stackrel{\text{def}}{=} \pi_1$, and denote by $\overline{S}$ the complement of a subset S: that is, $\overline{S} \stackrel{\text{def}}{=} [m] \setminus S$.

We begin by choosing a highly-deviating strategy for player 1, and argue that it is undominated. Specifically, choose arbitrarily a real number x larger than δ, and then choose a candidate set K_1 and a strategy (i.e., a valuation) v_1 as follows:

$$K_1 \stackrel{\text{def}}{=} \left\{ \theta_1 \in \Theta_1 \;\middle|\; \forall \text{ non-empty } S \subseteq [m],\ \theta_1(S) \in [x - \delta/2, x + \delta/2] \right\} \text{ and}$$

$$v_1(\pi_i) \stackrel{\text{def}}{=} x + (i-1)\delta \quad \forall i \in \{1, \ldots, 2^m - 1\} \ .$$

Note that $v_1 \notin K_1$. (Indeed, $v_1(\pi_i) \in K_1(\pi_1)$ only for $i = 1$.)

We now prove that the strategy v_1 is undominated. More precisely,

Claim 2.5. $v_1 \in \mathsf{UD}_1(K_1)$.

Proof. We proceed by contradiction. Assume towards contradiction that v_1 is weakly dominated by a strategy $v_1' \neq v_1$. (There are two cases to consider: v_1' is pure and v_i' is mixed. For simplicity we analyze only the first one.) Assume that v_1' is pure.

(There are two cases to consider: either v_i' is a constant shift of v_i or it is not. For brevity, we analyze only the second, harder, case.) Assume that v_i' is not a constant shift of v_i. Then

$$\exists j \in \{1, \ldots, 2^m - 1\} \quad \exists \Delta > 0 \quad v_1(\pi_{j+1}) - v_1(\pi_j) > \Delta > \max_{T \subseteq \pi_{j+1}} v_1'(T) - \max_{T \subseteq \pi_j} v_1'(T) \ . \quad (2.3)$$

Else, that is, if for all $i \in \{1, \ldots, 2^m - 1\}$

$$v_1(\pi_{i+1}) - v_1(\pi_i) \leq \max_{T \subseteq \pi_{i+1}} v_1'(T) - \max_{T \subseteq \pi_i} v_1'(T),$$

then summing up all these $2^m - 1$ inequalities we get $0 \leq 0$; hence, all the inequalities are in fact tight. So there must exist some constant c such that $v_1(\pi_i) = v_1'(\pi_i) + c$ for $i \in \{1, \ldots, 2^m - 1\}$, which we have assumed not to be the case.

(There are now two more cases to consider: $j \notin \{2^m - 2, 2^m - 1\}$ and $j \in \{2^m - 2, 2^m - 1\}$. For brevity we analyze only the first, hard, one.) Assume that $j \notin \{2^m - 2, 2^m - 1\}$. In this case neither $\overline{\pi_j}$ nor $\overline{\pi_{j+1}}$ is empty.

[10] In particular, we can order the subsets of $[m]$ by increasing cardinality, and lexicographically within a given cardinality: that is, when $m = 3$, $\{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$.

We contradict the assumption that v_1' weakly dominates v_1 by exhibiting a valuation $\theta_1 \in K_1$ and a "witness" strategy v_2 for player 2 such that

$$U_1(\theta_1, \mathsf{VCG}(v_1, v_2)) > U_1(\theta_1, \mathsf{VCG}(v_1', v_2)) \ .$$

We define v_2 as follows. Let H be a huge number (e.g., much higher than $v_1(\pi)$ and $v_1'(\pi)$ for any subset π of the goods) and let $v_2(\overline{\pi_{j+1}}) = H - \Delta$, $v_2(\overline{\pi_j}) = H$, and $v_2(T) = 0$ for all other subsets T. (Here we rely on the combinatorial nature of the auction: we have complete freedom on how to choose the valuation v_2.)

We now argue that the allocation in the outcome $\mathsf{VCG}(v_1, v_2)$ is $(\pi_{j+1}, \overline{\pi_{j+1}})$ and player 1's price is Δ. Indeed, because H was chosen to be sufficiently large, the only outcomes we should consider are $(T, \overline{\pi_{j+1}})$ and $(T', \overline{\pi_j})$ where $T \subseteq \pi_{j+1}$ and $T' \subseteq \pi_j$. By construction π_{j+1} maximizes $v_1(T)$ among all $T \subseteq \pi_{j+1}$, and π_j maximizes $v_1(T)$ among all $T \subseteq \pi_j$; in particular, the only two possible allocations are $(\pi_j, \overline{\pi_j})$ and $(\pi_{j+1}, \overline{\pi_{j+1}})$. Because $v_1(\pi_{j+1}) - v_1(\pi_j) > \Delta = v_2(\overline{\pi_j}) - v_2(\overline{\pi_{j+1}})$, the outcome that is chosen is $(\pi_{j+1}, \overline{\pi_{j+1}})$. As for the price: player 2 is allocated $\overline{\pi_{j+1}}$ but, if player 1 did not exist, player 2 would be allocated $\overline{\pi_j}$, and gain Δ in utility; thus player 1's price is indeed Δ.

Next, we argue that the allocation in the outcome $\mathsf{VCG}(v_1', v_2)$ is $(T^*, \overline{\pi_j})$, where T^* maximizes $v_1'(T)$ among all $T \subseteq \pi_j$, and player 1's price is 0. As before, because H was chosen to be sufficiently large, the only outcomes we should consider are $(T, \overline{\pi_{j+1}})$ and $(T', \overline{\pi_j})$ where $T \subseteq \pi_{j+1}$ and $T' \subseteq \pi_j$. This time by relying on the fact that

$$v_2(\overline{\pi_j}) - v_2(\overline{\pi_{j+1}}) = \Delta > \max_{T \subseteq \pi_{j+1}} v_1'(T) - \max_{T \subseteq \pi_j} v_1'(T)$$

we deduce that the outcome is in fact $(T^*, \overline{\pi_j})$. As for the price: player 2 is allocated $\overline{\pi_j}$ and, if player 1 did not exist, player 2 would still be allocated $\overline{\pi_j}$; thus player 1's price is indeed 0.

We now define $\theta_1 \in K_1$ as follows: $\theta_1(\pi_{j+1}) = x + \delta/2$, $\theta_1(\pi_j) = x - \delta/2$, and $\theta_1(\pi)$ is arbitrarily chosen for all other subsets π. For our choices of θ_1, v_1, v_1' and v_2 we have:

$$\begin{aligned} U_1(\theta_1, \mathsf{VCG}(v_1, v_2)) &= (x + \delta/2) - \Delta \\ U_1(\theta_1, \mathsf{VCG}(v_1', v_2)) &= (x - \delta/2) - 0 \ . \end{aligned}$$

By (2.3) and the construction of v_1, it is immediately seen that $\delta = v_1(\pi_{j+1}) - v_1(\pi_j) > \Delta$. Thus the first utility is greater than the second one, contradicting the fact that v_1' weakly dominates v_1. □

Having constructed $v_1 \in \mathsf{UD}_1(K_1)$, we continue the proof of Theorem 2.1' by letting:

$$v_2(S) \stackrel{\text{def}}{=} \begin{cases} (2^m - i - 1.5)\delta & \text{if } S = \overline{\pi_i} \text{ for some } i \in \{1, \ldots, 2^m - 2\} \\ x + (2^m - 2.5)\delta & \text{if } S = [m] \end{cases} ,$$

$$K_2 \stackrel{\text{def}}{=} \left\{ \theta_2 \in \Theta_2 \;\middle|\; \forall i \in \{1, \ldots, 2^m - 1\},\ \theta_2(\pi_i) \in \left[v_2(\pi_i), v_2(\pi_i) + \delta \right] \right\} \ .$$

Note that, by construction, $v_2 \in K_2$, which easily implies the following

Claim 2.6. $v_2 \in \mathsf{UD}_2(K_2)$. (For brevity we do not prove this implication.)

Having specified K_1, v_1, K_2, and v_2, all we have left is analyzing the social welfare performance.

Let us first compute the allocation of the outcome $\mathsf{VCG}(v_1, v_2)$. The only allocations to consider are $(\pi_{2^m-1}, \varnothing)$, $(\varnothing, \pi_{2^m-1})$, and $(\pi_i, \overline{\pi_i})$, for some index $i \in \{1, \ldots, 2^m - 2\}$. (In principle, one may also consider allocations where some goods remain unallocated. However, since v_1 and v_2 are strictly monotone —that is, $v_j(S) < v_j(T)$ for all $S \subsetneq T$ and all $j \in \{1, 2\}$— all goods must be allocated in the outcome of $\mathsf{VCG}(v_1, v_2)$.)

Now we compare the social welfare relative to (v_1, v_2) for such allocations:

$$\begin{aligned} v_1(\pi_{2^m-1}) + v_2(\varnothing) &= (x + (2^m - 2)\delta) + 0 = x + (2^m - 2)\delta \ , \\ v_1(\varnothing) + v_2(\pi_{2^m-1}) &= 0 + (x + (2^m - 2.5)\delta) = x + (2^m - 2.5)\delta \ , \text{ and} \\ v_1(\pi_i) + v_2(\overline{\pi_i}) &= (x + (i-1)\delta) + (2^m - i - 1.5)\delta = x + (2^m - 2.5)\delta \ . \end{aligned}$$

Thus, in the outcome $\mathsf{VCG}(v_1, v_2)$ the allocation is $(\pi_{2^m-1}, \varnothing)$. Hence, the social welfare is

$$\mathrm{SW}\big((\theta_1, \theta_2), \mathsf{VCG}(v_1, v_2)\big) = \theta_1(\pi_{2^m-1}) \ .$$

On the other hand, the maximum social welfare is

$$\mathrm{MSW}(\theta_1, \theta_2) \geq \theta_2(\pi_{2^m-1}) \ .$$

Now notice that for all $\theta \in K$, we have

$$\begin{aligned} \mathrm{MSW}(\theta) - \mathrm{SW}\big(\theta, \mathsf{VCG}(v_1, v_2)\big) &\geq \theta_2(\pi_{2^m-1}) - \theta_1(\pi_{2^m-1}) \\ &\geq (x + (2^m - 2.5)\delta) - (x + \delta/2) = (2^m - 3)\delta \ . \end{aligned}$$

That is, (2.1) holds. To prove (2.2), we choose θ as follows:

$$\begin{aligned} \theta_1(\pi_i) &\stackrel{\text{def}}{=} x - \delta/2 \quad \forall\, i \in \{1, \ldots, 2^m - 1\} \ , \\ \theta_2(\pi_i) &\stackrel{\text{def}}{=} v_2(\pi_i) + \delta \quad \forall\, i \in \{1, \ldots, 2^m - 1\} \ . \end{aligned}$$

Now notice that

$$\begin{aligned} \mathrm{MSW}(\theta) - \mathrm{SW}\big(\theta, \mathsf{VCG}(v_1, v_2)\big) &\geq \theta_2(\pi_{2^m-1}) - \theta_1(\pi_{2^m-1}) \\ &= (x + (2^m - 1.5)\delta) - (x - \delta/2) = (2^m - 1)\delta \ . \end{aligned}$$

That is, (2.2) also holds. This concludes our proof sketch of the weaker version of Theorem 2.1. ■

2.5 Proof of Theorem 2.2

Theorem 2.2. *In a combinatorial Knightian auction with n players and m goods, let the VCG mechanism break ties by preferring subsets with smaller cardinalities.*[11] *Then, for all δ, all products K of δ-approximate candidate sets, all profiles $\theta \in K$, and all profiles of strategies $v \in \mathsf{RM}^{\mathsf{pure}}(K)$,*

$$\mathrm{SW}(\theta, \mathsf{VCG}(v)) \geq \mathrm{MSW}(\theta) - 2\min\{m,n\}\delta \ .$$

Proof. We begin by noting that, because the VCG is dominant-strategy-truthful in the exact-valuation model, the (maximum) regret of a pure strategy v_i of a player i with candidate set K_i in the VCG mechanism becomes

$$\begin{aligned} R_i(K_i, v_i) &\stackrel{\text{def}}{=} \max_{\theta_i \in K_i} \max_{v_{-i}} \Big(\max_{v_i'} U_i\big(\theta_i, \mathsf{VCG}(v_i', v_{-i})\big) - U_i\big(\theta_i, \mathsf{VCG}(v_i, v_{-i})\big)\Big) \\ &= \max_{\theta_i \in K_i} \max_{v_{-i}} \Big(U_i\big(\theta_i, \mathsf{VCG}(\theta_i, v_{-i})\big) - U_i\big(\theta_i, \mathsf{VCG}(v_i, v_{-i})\big)\Big) \ , \end{aligned}$$

Moreover, by the very definition of the VCG, we have

$$U_i\big(\theta_i, \mathsf{VCG}(v_i, v_{-i})\big) = \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) - \mathrm{MSW}(v_{-i}) \ .^{12}$$

Therefore in the VCG case, we can further simplify the definition of regret as follows:

$$\begin{aligned} R_i(K_i, v_i) &= \max_{\theta_i \in K_i} \max_{v_{-i}} \Big(\mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(\theta_i, v_{-i})\big) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big)\Big) \\ &= \max_{\theta_i \in K_i} \max_{v_{-i}} \Big(\mathrm{MSW}\big(\theta_i, v_{-i}\big) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big)\Big) \ . \end{aligned} \tag{2.4}$$

Let us adopt a notation analogous to that of the proof in Chapter 1. Namely, for each player i, each candidate set $K_i \subset \Theta_i$, and each subset $T \subseteq [m]$, we let

$$\begin{gathered} K_i(T) \stackrel{\text{def}}{=} \{\theta_i(T)\}_{\theta_i \in K_i}, \quad K_i^{\perp}(T) \stackrel{\text{def}}{=} \inf K_i(T), \\ K_i^{\top}(T) \stackrel{\text{def}}{=} \sup K_i(T), \quad K_i^{\mathrm{mid}}(T) \stackrel{\text{def}}{=} (K_i^{\perp}(T) + K_i^{\top}(T))/2 \ . \end{gathered}$$

To prove Theorem 2.2, we rely on two intermediate claims. The first one identifies, for every player i, a strategy v_i with regret no larger than δ.

Claim 2.7. *For every player i, let $v_i^*(T) \stackrel{\text{def}}{=} K_i^{\mathrm{mid}}(T)$ for each $T \subseteq [M]$. Then $R_i(K_i, v_i^*) \leq \delta$.*

Proof of Claim 2.7. According to the first equality of (2.4), it suffices to show that

$$\forall \theta_i \in K_i \ \forall v_{-i}, \quad \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(\theta_i, v_{-i})\big) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i^*, v_{-i})\big) \leq \delta \ .$$

Let $\omega_1 = \mathsf{VCG}(\theta_i, v_{-i})$ and $\omega_2 = \mathsf{VCG}(v_i^*, v_{-i})$.

Recall that, in a combinatorial auction, a valuation $\theta_i \in \Theta_i$ of player i maps subsets of $[m]$ to $\mathbb{R}_{\geq 0}$. For convenience, we extend θ_i to map an outcome $\omega = (A, P)$ to $\mathbb{R}_{\geq 0}$ as follows: $\theta_i(\omega) \stackrel{\text{def}}{=} \theta_i(A_i)$.

[11] If giving subsets A or $B \subsetneq A$ to player i provides the same social welfare, then the VCG will give B to player i.

[12] This is because, suppose that the VCG mechanism picks an outcome $\omega = \mathsf{VCG}(v_i, v_{-i})$, allocating player i subset A_i and others A_{-i}. Then, i's price is $\mathrm{MSW}(v_{-i}) - v_{-i}(A_{-i})$ in ω. This induces a total utility of $\theta_i(A_i) + v_{-i}(A_{-i}) - \mathrm{MSW}(v_{-i}) = \mathrm{SW}((\theta_i, v_{-i}), \omega) - \mathrm{MSW}(v_{-i})$.

Under this notation, we have $v_i^*(\omega_2) + v_{-i}(\omega_2) \geq v_i^*(\omega_1) + v_{-i}(\omega_1)$, because the VCG maximizes social welfare relative to the strategy profile (v_i^*, v_{-i}). Using this inequality, we deduce that

$$\begin{aligned} & \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(\theta_i, v_{-i})\big) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i^*, v_{-i})\big) \\ = \ & \big(\theta_i(\omega_1) + v_{-i}(\omega_1)\big) - \big(\theta_i(\omega_2) + v_{-i}(\omega_2)\big) \\ = \ & \big(\theta_i(\omega_1) - \theta_i(\omega_2)\big) + \big(v_{-i}(\omega_1) - v_{-i}(\omega_2)\big) \\ \leq \ & \big(\theta_i(\omega_1) - \theta_i(\omega_2)\big) + \big(v_i^*(\omega_2) - v_i^*(\omega_1)\big) \ . \end{aligned}$$

Suppose player i gets subset $T_1 \subseteq [M]$ in outcome ω_1, and subset $T_2 \subseteq [M]$ in outcome ω_2. Then

$$\begin{aligned} \big(\theta_i(\omega_1) - \theta_i(\omega_2)\big) + \big(v_i^*(\omega_2) - v_i^*(\omega_1)\big) = \ & \big(\theta_i(T_1) - v_i^*(T_1)\big) + \big(v_i^*(T_2) - \theta_i(T_2)\big) \\ \leq \ & K_i^\top(T_1) - K_i^{\mathrm{mid}}(T_1) + K_i^{\mathrm{mid}}(T_2) - K_i^\perp(T_2) \\ \leq \ & \frac{\delta}{2} + \frac{\delta}{2} = \delta \ . \qquad \square \end{aligned}$$

Let us now prove another claim.

Claim 2.8. *Let v_i be any strategy of player i such that $R_i(K_i, v_i) \leq \delta$. Then:*

(a) for every $T \subseteq [M]$:

$$K_i^{\mathrm{mid}}(T) - \max_{T' \subseteq T} v_i(T') \leq \delta - \frac{K_i^\top(T) - K_i^\perp(T)}{2} \ , \ \textit{and}$$

(b) for every $T \subseteq [M]$ such that $v_i(T) > v_i(T')$ for all $T' \subsetneq T$:

$$|v_i(T) - K_i^{\mathrm{mid}}(T)| \leq \delta - \frac{K_i^\top(T) - K_i^\perp(T)}{2} \ .$$

Proof. Since the case of $T = \varnothing$ is trivial, we assume below that $T \neq \varnothing$. We first prove part (a).

Suppose that (a) is not true. Then, there exists T such that

$$K_i^{\mathrm{mid}}(T) - \max_{T' \subseteq T} v_i(T') > \delta - \frac{K_i^\top(T) - K_i^\perp(T)}{2} \ . \tag{2.5}$$

We contradict our assumption on v_i by showing that $R_i(K_i, v_i) > \delta$.

To show $R_i(K_i, v_i) > \delta$, as per (2.4), we must find some v_{-i} and some θ_i so that

$$\mathrm{MSW}\big(\theta_i, v_{-i}\big) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) > \delta \ . \tag{2.6}$$

Let j be an arbitrary player other than i. We choose $\theta_i \in K_i$ such that $\theta_i(T) =$

$K_i^\top(T)$,[13] and v_{-i} as follows: for every $S \subseteq [m]$

$$v_j(S) \stackrel{\text{def}}{=} \begin{cases} H & \text{if } S = \overline{T} \\ H + \varepsilon + \max_{T' \subseteq T} v_i(T') & \text{if } S = [M] \\ 0 & \text{otherwise} \end{cases}$$

and

$$v_k(S) \stackrel{\text{def}}{=} 0 \text{ for every } k \notin \{i, j\}.$$

Above, $\varepsilon > 0$ is some sufficiently small real number, and H is some huge real number (that is, H is much bigger than $v_i(S)$ for any subset S).[14] It then is easy to verify that the outcome $\mathsf{VCG}(v_i, v_{-i})$ allocates $\varnothing$ to player i, and $[M]$ to player j. Therefore,

$$\mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) = \theta_i(\varnothing) + v_j([m]) = H + \varepsilon + \max_{T' \subseteq T} v_i(T') \ .$$

On the other hand, $\mathrm{MSW}(\theta_i, v_{-i}) \geq \theta_i(T) + v_j(\overline{T}) = K_i^\top(T) + H$, and therefore

$$\begin{aligned} \mathrm{MSW}\big(\theta_i, v_{-i}\big) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) &\geq \Big(K_i^\top(T) + H\Big) - \Big(H + \varepsilon + \max_{T' \subseteq T} v_i(T')\Big) \\ = K_i^\top(T) - \varepsilon - \max_{T' \subseteq T} v_i(T') &= \frac{K_i^\top(T) - K_i^\perp(T)}{2} + K_i^{\mathrm{mid}}(T) - \varepsilon - \max_{T' \subseteq T} v_i(T') \ . \end{aligned}$$

Finally, since $K_i^{\mathrm{mid}}(T) - \max_{T' \subseteq T} v_i(T')$ is strictly greater than $\delta - \frac{K_i^\top(T) - K_i^\perp(T)}{2}$, according to (2.5), there exists some sufficiently small $\varepsilon > 0$ to make $\frac{K_i^\top(T) - K_i^\perp(T)}{2} + K_i^{\mathrm{mid}}(T) - \varepsilon - \max_{T' \subseteq T} v_i(T') > \delta$. This proves (2.6) and concludes the proof of Claim 2.8a.

We now prove part Claim 2.8b.

One side of Claim 2.8b is easy: that is, $v_i(T) - K_i^{\mathrm{mid}}(T) \geq -(\delta - \frac{K_i^\top(T) - K_i^\perp(T)}{2})$. Indeed, this inequality follows from $\max_{T' \subseteq T} v_i(T') = v_i(T)$ and Claim 2.8a.

To show the other side, that is, $v_i(T) - K_i^{\mathrm{mid}}(T) \leq \delta - \frac{K_i^\top(T) - K_i^\perp(T)}{2}$, we again proceed by contradiction. Suppose there is some T such that

$$v_i(T) - K_i^{\mathrm{mid}}(T) > \delta - \frac{K_i^\top(T) - K_i^\perp(T)}{2} \ . \tag{2.7}$$

We contradict our assumption on v_i by showing that $R_i(K_i, v_i) > \delta$. Similarly to case (a), we need to find some v_{-i} and some θ_i so that inequality (2.6) holds.

Let j be an arbitrary player other than i. This time, we choose $\theta_i \in K_i$ such that $\theta_i(T) = K_i^\perp(T)$,[13] and choose v_{-i} as follows: for every $S \subseteq [m]$

$$v_j(S) = \begin{cases} H & \text{if } S = \overline{T} \\ H - \varepsilon + v_i(T) & \text{if } S = [M] \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad v_k(S) \stackrel{\text{def}}{=} 0 \text{ for every } k \notin \{i, j\}.$$

[13]Here we have implicitly assumed that $K_i^\top(T) = \sup K_i(T) = \max K_i(T)$, and thus we can pick $\theta_i \in K_i$ so that $\theta_i(T) = K_i^\top(T)$. If this is not the case, one can construct an infinite sequence $\theta_i^{(1)}, \theta_i^{(2)}, \cdots$ so that $\theta_i(T)$ approaches to $K_i^\top(T)$, and the rest of the proof remains unchanged.

[14]Notice that when $T = [M]$ we have $\overline{T} = \varnothing$ and one cannot assign $v_j(\varnothing)$ to be a nonzero number. In that case we can choose $H = 0$, and the rest of the proof still goes through.

Again, $\varepsilon > 0$ is sufficiently small, and H is huge.[14] It then is easy to verify that the outcome $\mathsf{VCG}(v_i, v_{-i})$ allocates T to player i and $\overline{T}$ to player j. Therefore,

$$\mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) = \theta_i(T) + v_j(\overline{T}) = K_i^{\perp}(T) + H \ .$$

On the other hand, $\mathrm{MSW}(\theta_i, v_{-i}) \geq \theta_i(\varnothing) + v_j([M]) = H - \varepsilon + v_i(T)$. Therefore,

$$\begin{aligned}\mathrm{MSW}\big(\theta_i, v_{-i}\big) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) &\geq (H - \varepsilon + v_i(T)) - (K_i^{\perp}(T) + H) \\ &= v_i(T) - K_i^{\mathrm{mid}}(T) + \frac{K_i^{\top}(T) - K_i^{\perp}(T)}{2} - \varepsilon \ .\end{aligned}$$

Finally, since $v_i(T) - K_i^{\mathrm{mid}}(T)$ is strictly greater than $\delta - \frac{K_i^{\top}(T) - K_i^{\perp}(T)}{2}$ according to (2.7), there exists some sufficiently small $\varepsilon > 0$ to make $v_i(T) - K_i^{\mathrm{mid}}(T) + \frac{K_i^{\top}(T) - K_i^{\perp}(T)}{2} - \varepsilon > \delta$. This proves (2.6) and concludes the proof of Claim 2.8b.

In sum, Claim 2.8 holds. □

Now we return to the proof of Theorem 2.2. Let $v = (v_1, \ldots, v_n) \in \mathsf{RM}^{\mathsf{pure}}(K)$ be a regret-minimizing pure strategy profile, and let $\theta \in K$ be a valuation profile.

For every player i, the strategy v_i^* (i.e., the one reporting the 'middle points') has a regret at most δ, owing to Claim 2.7. Since v_i minimizes regret among all his strategies, we immediately have $R_i(K_i, v_i) \leq R_i(v_i^*, K_i) \leq \delta$. This shows that v_i satisfies the initial hypothesis of Claim 2.8.

Now, letting $(A_0, A_1, \ldots, A_n)$ be the allocation in the outcome $\mathsf{VCG}(v_1, \ldots, v_n)$, we immediately have $v_i(A_i) \geq v_i(T')$ for any $T' \subsetneq A_i$ by the definition of the VCG. Furthermore, by our choice of the tie-breaking rule, this inequality must be strict: that is, $v_i(A_i) > v_i(T')$ for any $T' \subsetneq A_i$. Therefore, letting $T = A_i$, T satisfies the hypothesis in Claim 2.8b. Thus, we conclude that

$$\begin{aligned}\forall i \in [n], \quad |v_i(A_i) - K_i^{\mathrm{mid}}(A_i)| \leq \delta - \frac{K_i^{\top}(A_i) - K_i^{\perp}(A_i)}{2} &\leq \delta - |\theta_i(A_i) - K_i^{\mathrm{mid}}(A_i)| \\ \implies |v_i(A_i) - \theta_i(A_i)| &\leq \delta \ . \quad (2.8)\end{aligned}$$

Notice that, if $A_i = \varnothing$, then $v_i(\varnothing) = \theta_i(\varnothing) = 0$.

Next, letting $(B_0, B_1, \ldots, B_n)$ be the allocation that maximizes the social welfare under θ, we have

$$\sum_{i=1}^{n} v_i(A_i) \geq \sum_{i=1}^{n} \max_{T' \subseteq B_i} v_i(T') \tag{2.9}$$

because the VCG maximizes social welfare relative to $v = (v_1, \ldots, v_n)$. Moreover, according to Claim 2.8a we have

$$\begin{aligned}\forall i \in [n], \quad K_i^{\mathrm{mid}}(B_i) - \max_{T' \subseteq B_i} v_i(T') \leq \delta - \frac{K_i^{\top}(B_i) - K_i^{\perp}(B_i)}{2} &\leq \delta - |\theta_i(B_i) - K_i^{\mathrm{mid}}(B_i)| \\ \implies \theta_i(B_i) - \max_{T' \subseteq B_i} v_i(T') &\leq \delta \ . \quad (2.10)\end{aligned}$$

Also notice that, if $B_i = \varnothing$, then $\theta_i(B_i) = \max_{T' \subseteq B_i} v_i(T') = 0$.

We are now ready to compute the social welfare guarantee.

$$\begin{aligned}
&\mathrm{SW}(\theta, \mathsf{VCG}(v)) = \textstyle\sum_{i=1}^n \theta_i(A_i) \geq \ \sum_{i=1}^n v_i(A_i) - \sum_{i\in[n], A_i\neq\varnothing} \delta && \text{(using (2.8))}\\
&\geq \ \sum_{i=1}^n \max_{T'\subseteq B_i} v_i(T') - \sum_{i\in[n], A_i\neq\varnothing} \delta && \text{(using (2.9))}\\
&\geq \ \textstyle\sum_{i=1}^n \theta_i(B_i) - \sum_{i\in[n], A_i\neq\varnothing} \delta - \sum_{i\in[n], B_i\neq\varnothing} \delta && \text{(using (2.10))}\\
&\geq \ \mathrm{MSW}(\theta) - 2\min\{n, m\}\delta \ .
\end{aligned}$$

This concludes the proof of Theorem 2.2. ■

Appendix

2.A Theorem 2.1: How to Obtain a Stronger Result and a Characterization

Payoff equivalence. Two strategies s_i and s_i' are *payoff-equivalent* for player i if for any strategy sub-profile s_{-i} of i's opponents and any $\theta_i \in K_i$, player i's utilities are the same when reporting s_i or s_i'. That is, there is no difference for i to report s_i or s_i'. Given a set of strategies S_i for player i, we denote by $\widehat{S}_i$ the set that also includes every strategy of i that is payoff-equivalent to some strategy in S_i. We will use this notation to simplify our statements of the results.

Remark. Two payoff-equivalent strategies of a player i may ultimately yield different outcomes, but they are effectively the same from i's point of view. Thus a solution concept cannot be meaningful unless, when it includes a strategy profile s, it also includes all strategy profiles s' such that s_i and s_i' are payoff equivalent for a player i.

We formally state Theorem 2.1 as follows.

Theorem 2.1. *In any unrestricted combinatorial auction with n (δ-approximate Knightian) players and m goods:*

(a) For any player i with candidate set K_i, $\mathsf{UD}_i(K_i) = \widehat{\mathbf{V}(K_i)}$.
(The set of strategies $\mathbf{V}(K_i)$ is formally defined in Definition 2.10, and geometrically described in Appendix 2.A.1 below.)

(b) Even if there are only two players, there exist products of δ-approximate candidate sets $K = K_1 \times K_2$ and profiles $(v_1, v_2) \in \mathsf{UD}(K)$, such that

$$\begin{aligned}
&\textit{(best-case } \theta\textit{)} \quad \forall\theta \in K_1\times K_2 \quad \mathrm{SW}\big(\theta, \mathsf{VCG}(v_1, v_2)\big) \leq \mathrm{MSW}(\theta) - (2^{m+1} - 5)\delta\\
&\textit{(worst-case } \theta\textit{)} \quad \exists\theta \in K_1\times K_2 \quad \mathrm{SW}\big(\theta, \mathsf{VCG}(v_1, v_2)\big) \leq \mathrm{MSW}(\theta) - (2^{m+1} - 3)\delta.
\end{aligned}$$

(In Appendix 2.B, we prove one direction of Theorem 2.1a: namely, $\mathsf{UD}_i(K_i) \supseteq \widehat{\mathbf{V}(K_i)}$. We shall prove $\mathsf{UD}_i(K_i) \subseteq \widehat{\mathbf{V}(K_i)}$ in the full version of the paper. In Appendix 2.C we show how to derive Theorem 2.1b from Theorem 2.1a.)

From sketch to proof. Let us say a few words about how the sketched proof in Section 2.4 can be extended to a full and slightly stronger proof. The first simplification we have made is to suppose that v_1' is a pure strategy. If instead v_1' is a mixed strategy, say it equals $\sum_j p^{(j)} v_1^{(j)}$ for $\sum_j p^{(j)} = 1$ where $v_1^{(j)}$ each is a pure strategy, then the first step is to distinguish between the following three cases (at least one of them always holds):

(a) $\exists j \in \{1, \ldots, 2^m - 1\},\ v_1(S_{j+1}) - v_1(S_j) > \min_j \left\{ \max_{T \subseteq S_{j+1}} v_1^{(j)}(T) - \max_{T \subseteq S_j} v_1^{(j)}(T) \right\}$

(b) $v_1(S_1) > \min_j \left\{ \max_{T \subseteq S_1} v_1^{(j)}(T) \right\}$

(c) $v_1(S_1) < \max_j \left\{ \max_{T \subseteq S_1} v_1^{(j)}(T) \right\}$

In the proof sketch above, we analyzed case (a) when v_1' happens to be a pure strategy. However, in a full proof, one has to analyze all three cases, without assuming that v_1' is pure. The analysis of each of these cases, is significantly more involved in this more general setting.

Furthermore, when analyzing case (a), we distinguished between the case $j \notin \{2^m - 2, 2^m - 1\}$ or $j \in \{2^m - 2, 2^m - 1\}$ and only analyzed the former. In the latter, the choices of "witnesses" $\theta_1 \in K_1$ and v_2 in order to create the contradiction $U_1(\theta_1, \mathsf{VCG}(v_1, v_2)) > U_1(\theta_1, \mathsf{VCG}(v_1', v_2))$ are different. Similarly, both (b) and (c) each have a witness specially crafted for it.

Only when all of (a), (b), and (c) are fully analyzed, we can really conclude that $v_1 \in \mathsf{UD}_1(K_1)$.

Finally, even if we expect $v_2 \in \mathsf{UD}_2(K_2)$ to be true, because $v_2 \in K_2$ (and thus v_2 is not a deviating strategy), actually proving that this is the case essentially amounts to an analysis that is not much more simple than the one required to show that the highly-deviating strategy v_1 is in $\mathsf{UD}_1(K_1)$. In our full proof in Appendix 2.C, we actually pick v_2 and K_2 more carefully (to be also highly-deviating), and doing so induces a slightly stronger result with the following social welfare upper bound:

$$\mathrm{SW}\big((\theta_1, \theta_2), \mathsf{VCG}(v_1, v_2)\big) \leq \mathrm{MSW}(\theta_1, \theta_2) - 2(2^m - 2)\delta \ .$$

2.A.1 Geometric Description of $\mathbf{V}(K_i)$

In this section, we just wish to provide an intuitive *description* of the set $\mathbf{V}(K_i)$, which will be formally defined in Definition 2.10.

The case of two goods. We first describe $\mathbf{V}(K_i)$ in the simpler case where there are only two goods on sale (i.e., $m = 2$). In this case, the non-empty subsets of the goods are $\{1\}, \{2\}, \{1, 2\}$; in particular, a valuation is a point (x, y, z) in three dimensions, and we can draw it. For the purpose of drawing, we fix the choice $K_i(\{1\}) = [6, 9]$, $K_i(\{2\}) = [8, 11]$ and $K_i(\{1, 2\}) = [10, 13]$.

We begin with two simple observations:

(a) any strategy that "bids below $\min K_i(S)$ at *every* coordinate $S \subseteq [m]$" is dominated; and

(b) any strategy that "bids above $\max K_i(S)$ at *every* coordinate $S \subseteq [m]$" is dominated.

Property (a) means that a strategy v_i such that, for *every* S, $v_i(S)$ is less than $\min K_i(S)$ cannot be in $\mathbf{V}(K_i)$. That is, $\mathbf{V}(K_i)$ does not share any strategies with the following cuboid (see Figure 2-1(a)):

$$\textsc{Cuboid}_1 \stackrel{\text{def}}{=} \left\{ (x,y,z) \;\middle|\; \begin{array}{l} x < \min K_i(\{1\}) \\ y < \min K_i(\{2\}) \\ z < \min K_i(\{1,2\}) \end{array} \right\} .$$

Similarly, property (b) means that a strategy v_i such that, for *every* S, $v_i(S)$ is greater than $\max K_i(S)$ cannot be in $\mathbf{V}(K_i)$. That is, $\mathbf{V}(K_i)$ does not share any strategies with the following cuboid (see Figure 2-1(b)):

$$\textsc{Cuboid}_2 \stackrel{\text{def}}{=} \left\{ (x,y,z) \;\middle|\; \begin{array}{l} x > \max K_i(\{1\}) \\ y > \max K_i(\{2\}) \\ z > \max K_i(\{1,2\}) \end{array} \right\} .$$

Provided that a strategy v_i is neither in $\textsc{Cuboid}_1$ nor $\textsc{Cuboid}_2$ (i.e., there are S' and S'' for which $v_i(S') > \min K_i(S')$ and $v_i(S'') < \max K_i(S'')$), there can be "many ways" in which v_i could be in $\mathbf{V}(K_i)$. To express this, we need an additional definition. For valuation sets (S_1, S_2, S_3), define

$$\textsc{Cyl}(S_1,S_2,S_3) \stackrel{\text{def}}{=} \left\{ (x,y,z) \;\middle|\; \begin{array}{l} x - y \geq \min S_1 - \max S_2 \\ y - z \geq \min S_2 - \max S_3 \\ z - x \geq \min S_3 - \max S_1 \end{array} \right\} .$$

Note that $\textsc{Cyl}(S_1, S_2, S_3)$ is a triangular cylinder defined by three halfspaces and its axis lies on the $x = y = z$ line. For a candidate set K_i, define (see Figure 2-1(c) and 2-1(d))

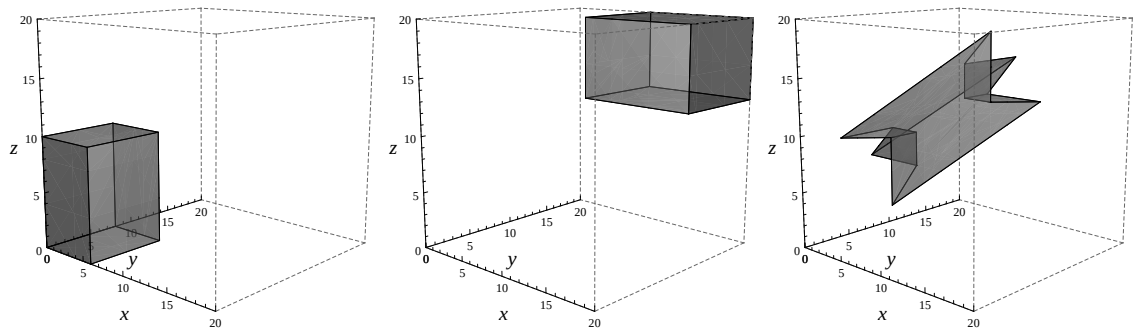
$$\begin{aligned} \textsc{Cyl}_1 &\stackrel{\text{def}}{=} \textsc{Cyl}(K_i(\{1\}), K_i(\{2\}), K_i(\{1,2\})) \\ \textsc{Cyl}_2 &\stackrel{\text{def}}{=} \text{“ } \textsc{Cyl}(K_i(\{2\}), K_i(\{1\}), K_i(\{1,2\})) \\ &\qquad \text{after the transformation } (x,y,z) \mapsto (y,x,z)\text{”}. \end{aligned}$$

Then, disregarding set boundaries, our definition of $\mathbf{V}(K_i)$ for $m = 2$ is as follows (see Figure 2-1(e)):

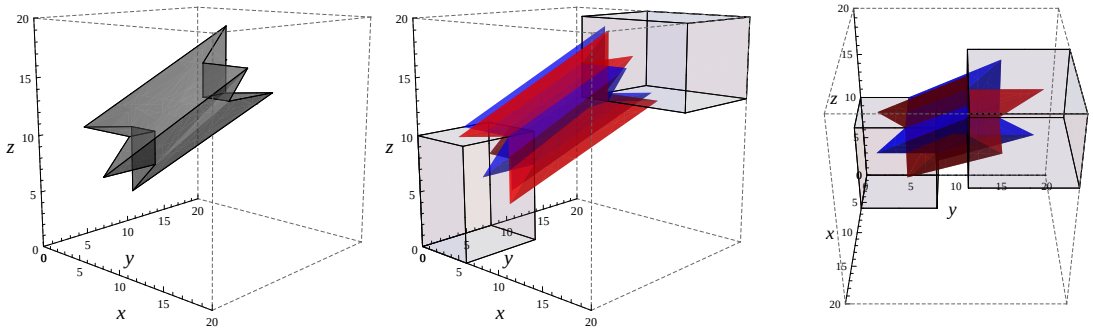
$$\mathbf{V}(K_i) = \textsc{Cyl}_1 \cup \textsc{Cyl}_2 - \textsc{Cuboid}_1 - \textsc{Cuboid}_2 \ .$$

The general case. In the general case (when m need not equal 2), we can analogously define $\textsc{Cuboid}_1$ and $\textsc{Cuboid}_2$. What becomes more complicated is the "cylinder structure" of $\mathbf{V}(K_i)$. Let us explain.

When $m = 2$, there are *two* cylinders in the definition of $\mathbf{V}(K_i)$ because there are two "proper" ways of ordering all non-empty subsets of the two goods: that is



(a) CUBOID$_1$ (b) CUBOID$_2$ (c) CYL$_1$



(d) CYL$_2$ (e) $\mathbf{V}(K_i)$ (f)

Figure 2-1: Here (f) is a PDF animated rotation (if viewed under Acrobat Reader), and can also be found at `http://people.csail.mit.edu/zeyuan/knightian/vcg.gif`.

$(\{1\},\{2\},\{1,2\})$ and $(\{2\},\{1\},\{1,2\})$. Thus, when $m=2$, $\mathbf{V}(K_i)$ is the union of the two cylinders respectively obtained by indexing the three sets $K_i(\{1\})$, $K_i(\{2\})$, and $K_i(\{1,2\})$ using the two proper orderings (and minus the two cuboids).

In the general case, there are more such "proper" orderings. Concretely, we say that a relabeling π of all the non-empty subsets of $[M]$ is *proper* if $j<k$ implies that $\pi(S_j) \not\supseteq \pi(S_k)$. (Note that $\pi(S_{2^m-1})=[m]$ is always the set of all goods.)

Analogously to the $m=2$ case, for each vector of sets $S=(S_1,\ldots,S_{2^m-1})$, we define the corresponding *fundamental cylinder* $\textsc{Cyl}(S)$. Then we consider the union of all fundamental cylinders corresponding to all vectors of sets obtained by properly relabeling $K_i=(K_i(S_1),\ldots,K_i(S_{2^m-1}))$. In sum, the description of $\mathbf{V}(K_i)$ in the general case is:

$$\mathbf{V}(K_i) = \bigcup_{\substack{\text{proper}\\ \pi}} \textsc{Cyl}(\pi(K_i)) - \textsc{Cuboid}_1 - \textsc{Cuboid}_2 \ .$$

For more details see Appendix 2.B.

2.B Proof of One Side of Theorem 2.1a

We introduce some notions before we proceed with the formal statement of the theorem. A *labeling* of all non-empty subsets of $[M]$ is a vector $\pi=(\pi_1,\ldots,\pi_{2^m-1})$, where the π_i's are the 2^m-1 distinct non-empty subsets of $[M]$.

Definition 2.9. *A labeling π of all non-empty subsets of $[M]$ is* **proper** *if $j<k \Rightarrow \pi_j \not\supseteq \pi_k$.*[15]

To make the result of our characterization clean, we assume that the candidate set K_i for the considered player i, is a cartesian product of intervals. That is, $K_i(T)=\{\theta_i(T)\}_{\theta_i\in K_i}=[a_T,b_T]$ for some $0\le a_T\le b_T$. We denote by $K_i^{\perp}(T)=a_T$ the minimum point in this interval and $K_i^{\top}(T)=b_T$ the maximum point in this set.

Definition 2.10. *For any player i with candidate set K_i, the set $\mathbf{V}(K_i)$ is the set of all strategies v_i satisfying the following two conditions:*

1. *at least one coordinate of v_i is below (resp., above) the corresponding upper (resp., lower) bound of K_i:*

$$\exists\, S'\subseteq[M],\quad v_i(S')\le K_i^{\top}(S')\ , \tag{2.11}$$

$$\exists\, S''\subseteq[M],\quad v_i(S'')\ge K_i^{\perp}(S'')\ ; \tag{2.12}$$

2. *there exists a proper labeling π of all non-empty subsets of $[M]$ such that, letting $\pi_{2^m}\overset{\text{def}}{=}\pi_1$,*

$$\forall\, j\in\{1,\ldots,2^m-1\},\quad v_i(\pi_j)-v_i(\pi_{j+1})\ge K_i^{\perp}(\pi_j)-K_i^{\top}(\pi_{j+1})\ . \tag{2.13}$$

[15] For instance, when m is equal to 3 such a permutation could be $(\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\})$, or $(\{1\},\{2\},\{3\},\{1,2\},\{2,3\},\{1,3\},\{1,2,3\})$, and there are plenty more such permutations.

In this section we prove the harder case of Theorem 2.1a: its "if" side. For this side, it suffices to show that if a strategy v_i is in $\mathbf{V}(K_i)$ then it is $\mathsf{UD}_i(K_i)$. In fact, our proof assumes for simplicity that both v_i and K_i satisfy some weak monotonicity conditions. We now proceed to formally state what we are going to prove, in Lemma 2.11 below.

Lemma 2.11 (one side of Theorem 2.1a)**.** *In the VCG mechanism for combinatorial auctions, no matter how ties are broken, for each player i having a weakly-monotone candidate set K_i the following holds.*[16] *If v_i is a strictly monotone strategy of i in $\mathbf{V}(K_i)$, then $v_i \in \mathsf{UD}_i(K_i)$.*

We fix a player i throughout, so we drop the subscript i everywhere. In fact, we can assume without loss of generality that $i = 1$, and that there is only another player, player 2, because all the other players can be chosen to report 0 and will thus not affect the analysis.

Assume by contradiction that a strategy v satisfying the hypothesis of the lemma is weakly dominated by some possibly mixed strategy $\{p_j, v^{(j)}\}_j$, where the probabilities p_j sum up to 1 and $v^{(j)} \neq v$ for all j. Our goal is to construct a "witness bid" $w\colon (2^{[M]} - \varnothing) \to \mathbb{R}_{\geq 0}$ for the second player and a "witness true valuation" $\theta \in K$ for the first player such that, if U is the utility function for the first player, then

$$U\big(\theta, \mathsf{VCG}(v, w)\big) > \sum_j p_j U\big(\theta, \mathsf{VCG}(v^{(j)}, w)\big) \ . \tag{2.14}$$

This will contradict the fact that the mixed strategy $\{p_j, v^{(j)}\}_j$ weakly dominates v. The construction of w and θ will be through a case analysis.

Notation.

- We call the player reporting v the first player, and the player reporting w the second player.
- We say that the allocation of $\mathsf{VCG}(v, w)$ is (S, T) if the first player receives $S \subseteq [M]$ and the second player receives $T \subseteq [M]$.
- We use $\mathrm{SW}[(S, T)] \stackrel{\text{def}}{=} v(S) + w(T)$ to denote the "apparent social welfare" of the allocation (S, T) (i.e., the social welfare when assuming that both players have the reported strategies (v, w) as their true valuations).
- Since the VCG mechanism maximizes social welfare relative to the reported strategies, we have that $\mathrm{SW}[\mathsf{VCG}(v, w)] = \max_{(S,T)}\{v(S) + w(T)\}$ where the maximization is over all $S, T \subseteq [M]$ with $S \cap T = \varnothing$.
- For notational simplicity, given a strategy v, we define its *monotonizer* $\widetilde{v}$ by $\widetilde{v}(S) \stackrel{\text{def}}{=} \max_{T \subseteq S} v(T)$.

[16] A candidate set K_i is weakly monotone if $K_i^{\perp}$ and $K_i^{\top}$ are weakly monotone: for all $S, T \subseteq [M]$ with $\varnothing \subsetneq S \subseteq T$, $K_i^{\perp}(S) \leq K_i^{\perp}(T)$ and $K_i^{\top}(S) \leq K_i^{\top}(T)$. A strategy is strictly monotone if for all $S, T \subseteq [M]$ with $\varnothing \subsetneq S \subseteq T$ it holds that $v_i(S) < v_i(T)$).

Next, among the following inequalities, at least one cannot hold:

$$\begin{cases} v(\pi_{i+1}) - v(\pi_i) \le \min_j \left\{ \widetilde{v^{(j)}}(\pi_{i+1}) - \widetilde{v^{(j)}}(\pi_i) \right\}, & \forall\, i \in \{1, \dots, 2^m - 1\} \\ v(S') \le \min_j \left\{ \widetilde{v^{(j)}}(S') \right\} \\ v(S'') \ge \max_j \left\{ \widetilde{v^{(j)}}(S'') \right\} \end{cases} \tag{2.15}$$

where π is any proper labeling guaranteed by the hypothesis of the lemma. Indeed, we now show that if all inequalities above hold, there must be a contradiction.

From the first inequality we deduce that, for each i and j, $v(\pi_{i+1}) - v(\pi_i) \le \widetilde{v^{(j)}}(\pi_{i+1}) - \widetilde{v^{(j)}}(\pi_i)$; for $i \in \{1, \dots, 2^m - 1\}$, all these sum up to $0 \le 0$. In particular, all such inequalities must be tight, so for each j, $v^{(j)}$ must be the same as v, up to a constant shift. In other words,

$$\forall\, S \subseteq [M] \text{ with } S \neq \varnothing\,, \quad v^{(j)}(S) = v(S) + c^{(j)} \text{ for some constant } c^{(j)}\ .$$

Substituting the above into the second and third inequality in (2.15), we deduce that $0 \le \min_j c^{(j)}$ and $0 \ge \max_j c^{(j)}$, and therefore the $c^{(j)}$ must all be 0, contradicting the fact that $v^{(j)} \neq v$.

Therefore, one of the three kinds of inequalities in (2.15) cannot hold; we thus have three cases, depending on which kind of inequality does not hold. We now show that, for each possible case (respectively discussed in Appendix 2.B.1, Appendix 2.B.2, and Appendix 2.B.3), (2.14) holds, and therefore the strategy v cannot be weakly dominated.

2.B.1 Case 1

Suppose that the first inequality of (2.15) does not hold for some i. For notational simplicity, assume that it does not hold for $i = 1$, i.e.,

$$v(\pi_2) - v(\pi_1) > \min_j \left\{ \widetilde{v^{(j)}}(\pi_2) - \widetilde{v^{(j)}}(\pi_1) \right\}\ .$$

We let $J = \arg\min_j \left\{ \widetilde{v^{(j)}}(\pi_2) - \widetilde{v^{(j)}}(\pi_1) \right\}$ be the set of minimizers, and let $j^* \in J$ be one of them. We can always choose some Δ such that

$$v(\pi_2) - v(\pi_1) > \Delta > \widetilde{v^{(j^*)}}(\pi_2) - \widetilde{v^{(j^*)}}(\pi_1)\ , \tag{2.16}$$

and for every $j \notin J$:

$$\widetilde{v^{(j)}}(\pi_2) - \widetilde{v^{(j)}}(\pi_1) > \Delta\ . \tag{2.17}$$

Now, we set the witness strategy of the other player to be $w(\overline{\pi_1}) = H + \Delta$, $w(\overline{\pi_2}) = H$ and $w(S) = 0$ anywhere else. Here H is some very large value. We will deal with the case when $\overline{\pi_1} = \varnothing$ or $\overline{\pi_2} = \varnothing$ later, since we cannot set the second player to have non-zero valuation on an empty set. We claim that:

Claim 2.12. *If $\overline{\pi_1} \neq \varnothing$ and $\overline{\pi_2} \neq \varnothing$:*

a. The allocation of $\mathsf{VCG}(v, w)$ is $\omega = (\pi_2, \overline{\pi_2})$.

b. *For all $j^* \in J$, the allocation of $\mathsf{VCG}(v^{(j^*)}, w)$ is $\omega = (T, \overline{\pi_1})$ for some $T \in \arg\max_{T \subseteq \pi_1} v^{(j^*)}(T)$ (or a probabilistic distribution over them in case of ties).*

c. *For all $j \notin J$, the allocation of $\mathsf{VCG}(v^{(j)}, w)$ is $\omega = (T, \overline{\pi_2})$ for some $T \in \arg\max_{T \subseteq \pi_2} v^{(j)}(T)$ (or a probabilistic distribution over them in case of ties).*

Proof. For any candidate allocation (S, T) of the VCG mechanism when the second player reports w, if $T \notin \{\overline{\pi_1}, \overline{\pi_2}\}$, then $\mathrm{SW}[(S,T)]$ does not contain the big term H and is thus smaller than any $\mathrm{SW}[\omega]$ in all three cases. Therefore, we only need to consider outcomes of the form $(S, \overline{\pi_1})$ and $(S, \overline{\pi_2})$.

a. In this case, $\mathrm{SW}[\omega] = v(\pi_2) + H$. If the allocation is of the form $(S, \overline{\pi_2})$, by the strict monotonicity of v, $(\pi_2, \overline{\pi_2}) = \omega$ must be the allocation with the best social welfare. If the allocation is of the form $(S, \overline{\pi_1})$, similarly, $(\pi_1, \overline{\pi_1})$ must be the allocation with the best social welfare, however, in this case $v(\pi_1) + w(\overline{\pi_1}) = v(\pi_1) + H + \Delta < v(\pi_2) + H = \mathrm{SW}[\omega]$, using (2.16). In sum, $\omega = (\pi_2, \overline{\pi_2})$ must be the allocation of the VCG mechanism.

b. In this case, $\mathrm{SW}[\omega] = \widetilde{v^{(j^*)}}(\pi_1) + H + \Delta$. For the allocation of $(S, \overline{\pi_1})$, S must be a subset of π_1 and therefore $S \in \arg\max_{T \subseteq \pi_1} v^{(j^*)}(T)$ as desired, since the VCG mechanism is outputting an allocation with the maximum reported social welfare. For the allocation of $(S, \overline{\pi_2})$, $\mathrm{SW}[(S, \overline{\pi_2})] \leq \widetilde{v^{(j^*)}}(\pi_2) + H < \widetilde{v^{(j^*)}}(\pi_1) + H + \Delta = \mathrm{SW}[\omega]$ (using (2.16)) is worse than the choice of ω. So the allocation must be of the desired form.

c. In this case, $\mathrm{SW}[\omega] = \widetilde{v^{(j)}}(\pi_2) + H$. For the allocation of $(S, \overline{\pi_1})$, we have that $\mathrm{SW}[(S, \overline{\pi_1})] \leq \widetilde{v^{(j)}}(\pi_1) + H + \Delta < \widetilde{v^{(j)}}(\pi_2) + H = \mathrm{SW}[\omega]$ (using (2.17)) is worse than the choice of ω. For the allocation of $(S, \overline{\pi_2})$, S must be a subset of π_2 and therefore $S \in \arg\max_{T \subseteq \pi_2} v^{(j)}(T)$ as desired, since the VCG mechanism is outputting an allocation with the maximum reported social welfare. In sum, the allocation must be of the desired form.

□

Claim 2.13. *When $\overline{\pi_1} = \varnothing$ or $\overline{\pi_2} = \varnothing$, Claim 2.12 only requires the following small changes:*

a. *When $\overline{\pi_1} = \varnothing$ (i.e., $\pi_1 = [M]$), at any time $(T, \overline{\pi_1})$ is a possible allocation declared in Claim 2.12, (T, R) for $R \subseteq \overline{T}$ is now also possible.*[17]

b. *When $\overline{\pi_2} = \varnothing$ (i.e., $\pi_2 = [M]$), at any time $(T, \overline{\pi_2})$ is a possible allocation declared in Claim 2.12, (T, R) for $R \subseteq \overline{T}$ is now also possible.*[18]

[17] As a consequence, Claim 2.12(a) and Claim 2.12(c) still hold, but Claim 2.12(b) will be changed to include the possible outcomes of $\omega = (T, R)$ where T is still in $\arg\max_{T \subseteq \pi_2} v^{(j)}(T)$ but $w \subseteq \overline{T}$.

[18] As a consequence, Claim 2.12(b) still holds, but Claim 2.12(a) and Claim 2.12(c) need small changes.

Proof.

a. This is because, due to the (strict) monotonicity of v we have $v(\pi_1) > v(\pi_2)$ and thus (2.16) tells us that $\Delta < 0$. Instead of choosing some sufficiently large H, we can choose $H = -\Delta$. It will make sure that $w(\varnothing) = w(\overline{\pi_1}) = 0$ while $w(\overline{\pi_2}) = -\Delta > 0$. The only place that we used H being sufficiently large, is where we declare that the only possible candidate allocation for $\mathsf{VCG}(\cdot, w)$ is of the form $(S, \overline{\pi_1})$ or $(S, \overline{\pi_2})$. This is no longer true as we have to also consider (S, R) for $R \neq \overline{\pi_1}$ or $\overline{\pi_2}$. However, since $w(R) = 0$, $\mathrm{SW}[(S, R)] = \mathrm{SW}[(S, \varnothing)] = \mathrm{SW}[(S, \overline{\pi_1})]$. This means, allocation (S, R) will be possible *only if* $(S, \overline{\pi_1})$ is possible.

b. This is because, due to the weak monotonicity of $\widetilde{v^{(j^*)}}$ we have $\widetilde{v^{(j^*)}}(\pi_2) \geq \widetilde{v^{(j^*)}}(\pi_1)$ and thus (2.16) tells us that $\Delta > 0$. Instead of choosing some sufficiently large H, we can choose $H = 0$. It will make sure that $w(\varnothing) = w(\overline{\pi_2}) = 0$ while $w(\overline{\pi_1}) = \Delta > 0$. The only place that we used H being sufficiently large, is where we declare that the only possible candidate allocation for $\mathsf{VCG}(\cdot, w)$ is of the form $(S, \overline{\pi_1})$ or $(S, \overline{\pi_2})$. This is no longer true as we have to also consider (S, R) for $R \neq \overline{\pi_1}$ or $\overline{\pi_2}$. However, since $w(R) = 0$, $\mathrm{SW}[(S, R)] = \mathrm{SW}[(S, \varnothing)] = \mathrm{SW}[(S, \overline{\pi_2})]$. This means, allocation (S, R) will be possible *only if* $(S, \overline{\pi_2})$ is possible.

□

Now, we have some knowledge about what outcomes could be outputted by the VCG mechanism, on input (v, w), and on $(v^{(j)}, w)$. We now come to the final part that is to show that (2.14) holds. We first compute the utilities in all three cases:

Claim 2.14. *If we choose $\theta(\pi_2) = K^{\top}(\pi_2)$ and $\theta(S) = K^{\perp}(S)$ for everything else (i.e., $S \neq \varnothing$ and $S \neq \pi_2$).*

a. $U(\theta, \mathsf{VCG}(v, w)) = K^{\top}(\pi_2) + H - \max_S w(S)$,

b. $U(\theta, \mathsf{VCG}(v^{(j^*)}, w)) \leq K^{\perp}(\pi_1) + H + \Delta - \max_S w(S)$ *for every* $j^* \in J$, *and*

c. $U(\theta, \mathsf{VCG}(v^{(j)}, w)) \leq K^{\top}(\pi_2) + H - \max_S w(S)$ *for every* $j \notin J$.

Proof.

a. We have proved in Claim 2.12(a) that $(\pi_2, \overline{\pi_2})$ is the only possible allocation in this case, and therefore $U(\theta, \mathsf{VCG}(v, w)) = U(\theta, (\pi_2, \overline{\pi_2})) = K^{\top}(\pi_2) + w(\overline{\pi_2}) - \max_S w(S) = K^{\top}(\pi_2) + H - \max_S w(S)$.

b. We have proved in Claim 2.12(b) that $(T, \overline{\pi_1})$ is the only possible allocation in this case, and therefore if $T \neq \pi_2$, we have $U(\theta, \mathsf{VCG}(v^{(j^*)}, w)) = K^{\perp}(T) +$

$w(\overline{\pi_1}) - \max_S w(S) \le K^\perp(\pi_1) + H + \Delta - \max_S w(S)$. (Here we used the weak monotonicity of $K^\perp$, i.e., $K^\perp(T) \le K^\perp(\pi_1)$.)

Otherwise, if $T = \pi_2$ (i.e., the allocation is $(\pi_2, \overline{\pi_1})$, we must have that $\pi_2 \subsetneq \pi_1$. By the (strict) monotonicity of v and (2.16), we have that $\Delta < V(\pi_2) - V(\pi_1) < 0$. In this case, since $w(\overline{\pi_1}) = H + \Delta = w(\overline{\pi_2}) + \Delta$, we know that $\mathrm{SW}[(\pi_2, \overline{\pi_2})] = \mathrm{SW}[(\pi_2, \overline{\pi_1})] - \Delta > \mathrm{SW}[(\pi_2, \overline{\pi_1})]$. This indicates that $(\pi_2, \overline{\pi_1})$ will never be a possible outcome , giving a contradiction.

c. We have proved in Claim 2.12(c) that $(T, \overline{\pi_2})$ is the only possible allocation in this case, and therefore $U(\theta, \mathsf{VCG}(v^{(j^*)}, w)) \le K^\top(T) + w(\overline{\pi_2}) - \max_S w(S) \le K^\top(\pi_2) + w(\overline{\pi_2}) - \max_S w(S) = K^\top(\pi_2) + H - \max_S w(S)$. (Here we used the weak monotonicity of $K^\top$, i.e., $K^\top(T) \le K^\top(\pi_2)$.)

We remark here that, in the case when $\overline{\pi_1} = \varnothing$ or $\overline{\pi_2} = \varnothing$, the allocation might also be (S, R) for some $w(R) = 0$, but one can check that the same conclusions still hold, by our choice of H.) □

Corollary 2.15. *(2.14) is satisfied.*

Proof. We recall that (2.13) tells us that $v(\pi_2) - v(\pi_1) \le K^\top(\pi_2) - K^\perp(\pi_1)$, but we have $v(\pi_2) - v(\pi_1) > \Delta$ in (2.16). This tells us that $K^\top(\pi_2) > K^\perp(\pi_1) + \Delta$.

Now, for every $j^* \in J$,

$$U(\theta, \mathsf{VCG}(v, w)) = K^\top(\pi_2) + H - \max_S w(S) > K^\perp(\pi_1) + H + \Delta - \max_S w(S) \ge U(\theta, \mathsf{VCG}(v^{(j^*)}, w))$$

while for every $j \notin J$,

$$U(\theta, \mathsf{VCG}(v, w)) = K^\top(\pi_2) + H - \max_S w(S) \ge U(\theta, \mathsf{VCG}(v^{(j)}, w))$$

The combination of them immediately implies (2.14) □

We recall that (2.14) gives a contradiction and says that v is an undominated strategy, and this ends the proof of Lemma 2.11, for Case 1.

2.B.2 Case 2

Suppose that the second inequality of (2.15) does not hold, that is, $v(S') > \min_j\{v^{(j)}(S')\}$. Similarly as in Case 1, we let $J = \arg\min_j \left\{\widetilde{v^{(j)}}(S')\right\}$ be the set of minimizers, and let $j^* \in J$ be one of them. We can always choose some Δ such that

$$v(S') > \Delta > \widetilde{v^{(j^*)}}(S') \ , \tag{2.18}$$

and for every $j \notin J$:

$$\widetilde{v^{(j)}}(S') > \Delta \ . \tag{2.19}$$

Now, consider the following witness player, with $w(\overline{S'}) = H$ and $w([M]) = H + \Delta$, and $w(S) = 0$ everywhere else. Notice that unlike Case 1, $\Delta > 0$ is always positive. We also let H be sufficiently large when $\overline{S'} \neq \varnothing$. We choose $H = 0$ if $\overline{S'} = \varnothing$.

Claim 2.16 (A variant of Claim 2.12)**.** *If* $\overline{S'} \neq \varnothing$*,*

a. *The allocation of* $\mathsf{VCG}(v, w)$ *is* $\omega = (S', \overline{S'})$

b. *For all* $j^* \in J$*, the allocation of* $\mathsf{VCG}(v^{(j^*)}, w)$ *is* $\omega = (\varnothing, [M])$*.*

c. *For all* $j \notin J$*, the allocation of* $\mathsf{VCG}(v^{(j)}, w)$ *is* $\omega = (T, \overline{S'})$*, where* $T \in \arg\max_{T \subseteq S'} v^{(j)}(T)$ *(or a probabilistic distribution over them in case of ties).*

Proof. For any candidate allocation (S, T) of the VCG mechanism when the second player reports w, if $T \notin \{\overline{S'}, [M]\}$, then $\mathrm{SW}[(S, T)]$ does not contain the big term H and is thus smaller than any $\mathrm{SW}[\omega]$ in all three cases. Therefore, we only need to consider outcomes of the form $(S, \overline{S'})$ and $(\varnothing, [M])$.

a. In this case, $\mathrm{SW}[\omega] = v(S') + H$. If the allocation is of the form $(S, \overline{S'})$, by the strict monotonicity of v, $(S', \overline{S'}) = \omega$ must be the allocation with the best social welfare. If the allocation is $(\varnothing, [M])$ its social welfare $\mathrm{SW}[(\varnothing, [M])] = \Delta + H < v(S') + H = \mathrm{SW}[\omega]$, using (2.18). In sum, $\omega = (S', \overline{S'})$ must be the allocation of the VCG mechanism.

b. In this case, $\mathrm{SW}[\omega] = H + \Delta$. For the allocation of the form $(S, \overline{S'})$, $\mathrm{SW}[(S, \overline{S'})] \leq \widetilde{v^{(j^*)}}(S) + H < H + \Delta = \mathrm{SW}[\omega]$ (using (2.18)) is worse than the choice of ω.

c. In this case, $\mathrm{SW}[\omega] = \widetilde{v^{(j)}}(S') + H$. For the allocation of $(\varnothing, [M])$, we have that $\mathrm{SW}[(\varnothing, [M])] = H + \Delta < \widetilde{v^{(j)}}(S') + H = \mathrm{SW}[\omega]$ (using (2.19)) is worse than the choice of ω. For the allocation of the form $(S, \overline{S'})$, S must be a subset of S' and therefore $S \in \arg\max_{T \subseteq S'} v^{(j)}(T)$ as desired, since the VCG mechanism is outputting an allocation with the maximum reported social welfare. In sum, the allocation must be of the desired form.

□

Claim 2.17 (A variant of Claim 2.13)**.** *When* $\overline{S'} = \varnothing$ *(i.e.,* $S' = [M]$*), Claim 2.16 only requires the following small changes:*

> *at any time* $(T, \overline{S'})$ *is a possible allocation declared in Claim 2.16,* (T, R) *for* $R \subseteq \overline{T}$ *is now also possible.*[19]

Proof. Recall that, instead of choosing some sufficiently large H, we choose $H = 0$ in this case. The only place that we used H being sufficiently large, is where we declare that the only possible candidate allocation for $\mathsf{VCG}(\cdot, w)$ is of the form $S, \overline{S'})$ or $(\varnothing, [M])$. This is no longer true as we have to also consider (S, R) for $R \neq \overline{S'}$ or $[M]$. However, since $w(R) = 0$, $\mathrm{SW}[(S, R)] = \mathrm{SW}[(S, \varnothing)] = \mathrm{SW}[(S, \overline{S'})]$. This means, allocation (S, R) will be possible *only if* $(S, \overline{S'})$ is possible. □

[19] As a consequence, Claim 2.16(b) still holds, but Claim 2.16(a) and Claim 2.16(c) need small changes.

Now, we have some knowledge about what outcomes could be outputted by the VCG mechanism, on input (v, w) and on $(v^{(j)}, w)$. We now come to the final part that is to show that (2.14) holds. We first compute the utilities in all three cases:

Claim 2.18 (A variant of Claim 2.14)**.** *If we choose $\theta(S) = K^\top(S)$ for everything non-empty S:*

a. $U(\theta, \mathsf{VCG}(v, w)) = K^\top(S') + H - \max_S w(S)$,

b. $U(\theta, \mathsf{VCG}(v^{(j^*)}, w)) = H + \Delta - \max_S w(S)$ *for every* $j^* \in J$, *and*

c. $U(\theta, \mathsf{VCG}(v^{(j)}, w)) \leq K^\top(S') + H - \max_S w(S)$ *for every* $j \notin J$.

Proof.

a. We have proved in Claim 2.16(a) that $(S', \overline{S'})$ is the only possible allocation in this case, and therefore $U(\theta, \mathsf{VCG}(v, w)) = U(\theta, (S', \overline{S'})) = K^\top(S') + w(\overline{S'}) - \max_S w(S) = K^\top(S') + H - \max_S w(S)$.

   (In the case when $\overline{S'} = \varnothing$, the allocation might also be (S', R) for some $w(R) = 0$, and since we have chosen $H = 0$ this utility equation still holds.)

b. We have proved in Claim 2.16(b) that $(\varnothing, [M])$ is the only possible allocation in this case, and therefore $U(\theta, \mathsf{VCG}(v^{(j^*)}, w)) = 0 + w([M]) - \max_S w(S) = H + \Delta - \max_S w(S)$.

c. We have proved in Claim 2.16(c) that $(T, \overline{S'})$ is the only possible allocation in this case, and therefore $U(\theta, \mathsf{VCG}(v^{(j)}, w)) \leq K^\top(T) + w(\overline{S'}) - \max_S w(S) \leq K^\top(S') + w(\overline{S'}) - \max_S w(S) = K^\top(S') + H - \max_S w(S)$.

   (Here we used the weak monotonicity of $K^\top$, i.e., $K^\top(T) \leq K^\top(S')$. In the case when $\overline{S'} = \varnothing$, the allocation might also be (T, R) for some $w(R) = 0$, and since we have chosen $H = 0$ this utility equation still holds.)

□

Corollary 2.19. *(2.14) is satisfied.*

Proof. We recall that (2.11) and (2.18) tell us that $\Delta < v(S') \leq K^\top(S')$. Now, for every $j^* \in J$,

$$U(\theta, \mathsf{VCG}(v, w)) = K^\top(S') + H - \max_S w(S) > H + \Delta - \max_S w(S) = U(\theta, \mathsf{VCG}(v^{(j^*)}, w))$$

while for every $j \notin J$,

$$U(\theta, \mathsf{VCG}(v, w)) = K^\top(S') + H - \max_S w(S) \geq U(\theta, \mathsf{VCG}(v^{(j)}, w))$$

The combination of them immediately implies (2.14) □

We recall that (2.14) gives a contradiction and says that v is an undominated strategy, and this ends the proof of Lemma 2.11, for Case 2.

2.B.3 Case 3

Suppose that the second inequality of (2.15) does not hold, that is, $v(S'') < \max_j\{v^{(j)}(S'')\}$. Similarly as in Cases 1 and 2, we let $J = \arg\max_j \left\{\widetilde{v^{(j)}}(S'')\right\}$ be the set of maximizers, and let $j^* \in J$ be one of them. We can always choose some Δ such that

$$v(S'') < \Delta < \widetilde{v^{(j^*)}}(S'') \ , \tag{2.20}$$

and for every $j \notin J$:

$$\widetilde{v^{(j)}}(S'') < \Delta \ . \tag{2.21}$$

Now, consider the following witness player, with $w(\overline{S''}) = H$ and $w([M]) = H+\Delta$, and $w(S) = 0$ everywhere else. Notice that unlike Case 1, $\Delta > 0$ is always positive. We also let H be sufficiently large when $\overline{S''} \neq \varnothing$. We choose $H = 0$ if $\overline{S''} = \varnothing$.

Claim 2.20 (A variant of Claim 2.12)**.** *If $\overline{S''} \neq \varnothing$,*

a. *The allocation of $\mathsf{VCG}(v, w)$ is $\omega = (\varnothing, [M])$.*

b. *For all $j^* \in J$, the allocation of $\mathsf{VCG}(v^{(j^*)}, w)$ is $\omega = (T, \overline{S''})$, where $T \in \arg\max_{T\subseteq S''} v^{(j)}(T)$ (or a probabilistic distribution over them in case of ties).*

c. *For all $j \notin J$, the allocation of $\mathsf{VCG}(v^{(j)}, w)$ is $\omega = (\varnothing, [M])$.*

Proof. For any candidate allocation (S, T) of the VCG mechanism when the second player reports w, if $T \notin \{\overline{S''}, [M]\}$, then $\mathrm{SW}[(S, T)]$ does not contain the big term H and is thus smaller than any $\mathrm{SW}[\omega]$ in all three cases. Therefore, we only need to consider outcomes of the form $(S, \overline{S''})$ and $(\varnothing, [M])$.

a. In this case, $\mathrm{SW}[\omega] = H + \Delta$. If the allocation is of the form $(S, \overline{S''})$, by the strict monotonicity of v, $(S'', \overline{S''}) = \omega$ must be the allocation with the best social welfare. However, its social welfare $\mathrm{SW}[(S'', \overline{S''})] = v(S'') + H < H + \Delta = \mathrm{SW}[\omega]$, using (2.20). In sum, $(\varnothing, [M])$ must be the allocation of the VCG mechanism.

b. In this case, $\mathrm{SW}[\omega] = \widetilde{v^{(j^*)}}(S'')+H$. For the allocation of $(\varnothing, [M])$, we have that $\mathrm{SW}[(\varnothing, [M])] = H + \Delta < \widetilde{v^{(j^*)}}(S'') + H = \mathrm{SW}[\omega]$ (using (2.20)) is worse than the choice of ω. For the allocation of the form $(S, \overline{S''})$, S must be a subset of S'' and therefore $S \in \arg\max_{T\subseteq S''} v^{(j^*)}(T)$ as desired, since the VCG mechanism is outputting an allocation with the maximum reported social welfare. In sum, the allocation must be of the desired form.

c. In this case, $\mathrm{SW}[\omega] = H+\Delta$. For the allocation of the form $(S, \overline{S''})$, $\mathrm{SW}[(S, \overline{S''})] \leq \widetilde{v^{(j)}}(S) + H < H + \Delta = \mathrm{SW}[\omega]$ (using (2.21)) is worse than the choice of ω.

□

Claim 2.21 (A variant of Claim 2.13)**.** *When $\overline{S''} = \varnothing$ (i.e., $S'' = [M]$), Claim 2.20 only requires the following small changes:*

> *at any time $(T, \overline{S''})$ is a possible allocation declared in Claim 2.20, (T, R) for $R \subseteq \overline{T}$ is now also possible.*[20]

Proof. Recall that, instead of choosing some sufficiently large H, we choose $H = 0$ in this case. The only place that we used H being sufficiently large, is where we declare that the only possible candidate allocation for $\mathsf{VCG}(\cdot, w)$ is of the form $(S, \overline{S''})$ or $(\varnothing, [M])$. This is no longer true as we have to also consider (S, R) for $R \neq \overline{S''}$ or $[M]$. However, since $w(R) = 0$, $\mathrm{SW}[(S, R)] = \mathrm{SW}[(S, \varnothing)] = \mathrm{SW}[(S, \overline{S''})]$. This means, allocation (S, R) will be possible *only if* $(S, \overline{S''})$ is possible. □

Now, we have some knowledge about what outcomes could be outputted by the VCG mechanism, on input (v, w) and on $(v^{(j)}, w)$. We now come to the final part that is to show that (2.14) holds. We first compute the utilities in all three cases:

Claim 2.22 (A variant of Claim 2.14)**.** *If we choose $\theta(S) = K^{\perp}(S)$ for all non-empty S:*

a. $U(\theta, \mathsf{VCG}(v, w)) = H + \Delta - \max_S w(S)$,

b. $U(\theta, \mathsf{VCG}(v^{(j^*)}, w)) \leq H + K^{\perp}(S'') - \max_S w(S)$ *for every* $j^* \in J$, *and*

c. $U(\theta, \mathsf{VCG}(v^{(j)}, w)) = \Delta + H - \max_S w(S)$ *for every* $j \notin J$.

Proof.

a. We have proved in Claim 2.20(a) that $(\varnothing, [M])$ is the only possible allocation in this case, and therefore $U(\theta, \mathsf{VCG}(v, w)) = U(\theta, (\varnothing, [M])) = 0 + w(\overline{S''}) - \max_S w(S) = H + \Delta - \max_S w(S)$.

b. We have proved in Claim 2.20(b) that $(T, \overline{S''})$ is the only possible allocation in this case, and therefore $U(\theta, \mathsf{VCG}(v^{(j^*)}, w)) \leq K^{\perp}(T) + w(\overline{S''}) - \max_S w(S) \leq K^{\perp}(S'') + w(\overline{S''}) - \max_S w(S) = K^{\perp}(S'') + H - \max_S w(S)$.

   (Here we used the weak monotonicity of $K^{\perp}$, i.e., $K^{\perp}(T) \leq K^{\perp}(S'')$. In the case when $\overline{S''} = \varnothing$, the allocation might also be (T, R) for some $w(R) = 0$, and since we have chosen $H = 0$ this utility equation still holds.)

c. We have proved in Claim 2.20(c) that $(\varnothing, [M])$ is the only possible allocation in this case, and therefore $U(\theta, \mathsf{VCG}(v^{(j)}, w)) = 0 + w([M]) - \max_S w(S) = H + \Delta - \max_S w(S)$.

[20] As a consequence, Claim 2.20(a) and Claim 2.20(c) still hold, but Claim 2.20(b) needs small changes.

□

Corollary 2.23. *(2.14) is satisfied.*

Proof. We recall that (2.12) and (2.20) tell us that $\Delta > v(S'') \geq K^{\perp}(S'')$. Now, for every $j^* \in J$,

$$U(\theta, \mathsf{VCG}(v,w)) = H+\Delta-\max_S w(S) > H+K^{\perp}(S'')-\max_S w(S) = U(\theta, \mathsf{VCG}(v^{(j^*)},w))$$

while for every $j \notin J$,

$$U(\theta, \mathsf{VCG}(v,w)) = H + \Delta - \max_S w(S) = U(\theta, \mathsf{VCG}(v^{(j)},w))$$

The combination of them immediately implies (2.14) □

We recall that (2.14) gives a contradiction and says that v is an undominated strategy, and this ends the proof of Lemma 2.11, for Case 3.

2.C Proof of Theorem 2.1b

Theorem 2.1b (restated)**.** *In a combinatorial Knightian auction with* 2 *players and* m *goods, consider the VCG with any tie-breaking rule, then there exist products of* δ*-approximate candidate sets* $K = K_1 \times K_2$ *and profiles* $(v_1, v_2) \in \mathsf{UD}(K)$*, such that*

$$\textit{(best-case } \theta\textit{)} \quad \forall \theta \in K_1 \times K_2 \quad \mathrm{SW}\big(\theta, \mathsf{VCG}(v_1,v_2)\big) \leq \mathrm{MSW}(\theta) - (2^{m+1}-5)\delta \tag{2.22}$$

$$\textit{(worst-case } \theta\textit{)} \quad \exists \theta \in K_1 \times K_2 \quad \mathrm{SW}\big(\theta, \mathsf{VCG}(v_1,v_2)\big) \leq \mathrm{MSW}(\theta) - (2^{m+1}-3)\delta. \tag{2.23}$$

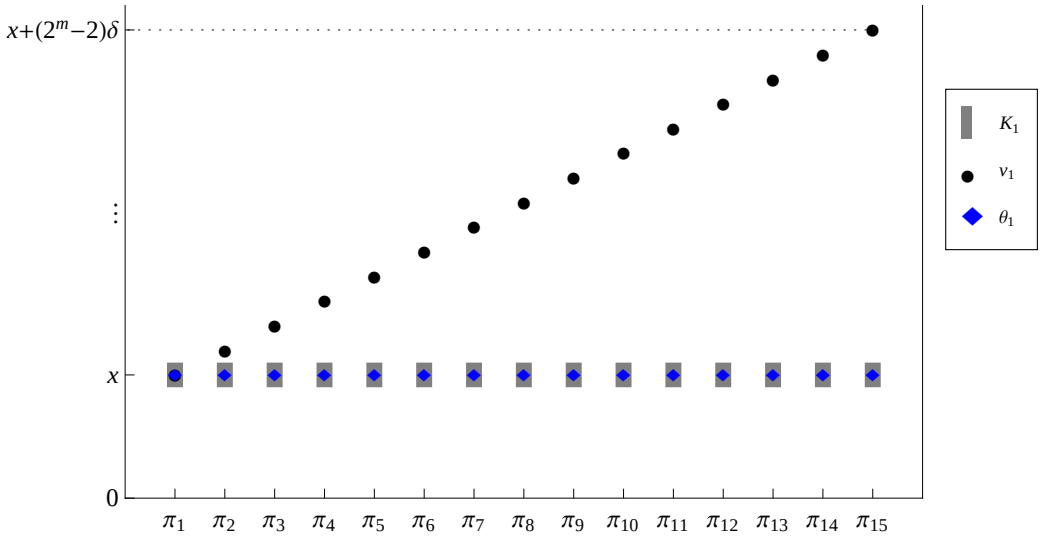
We prove the theorem in two steps.

Step 1 (Appendix 2.C.1). We construct a candidate hard instance for the VCG mechanism, by specifying two candidate sets K_1 and K_2 and two corresponding undominated strategies v_1 and v_2, for player 1 and player 2 respectively. To show that indeed $v_1 \in \mathsf{UD}_1(K_1)$ and $v_2 \in \mathsf{UD}_2(K_2)$, we prove that our choices of v_1 and v_2 do satisfy the requirements given in Lemma 2.11.

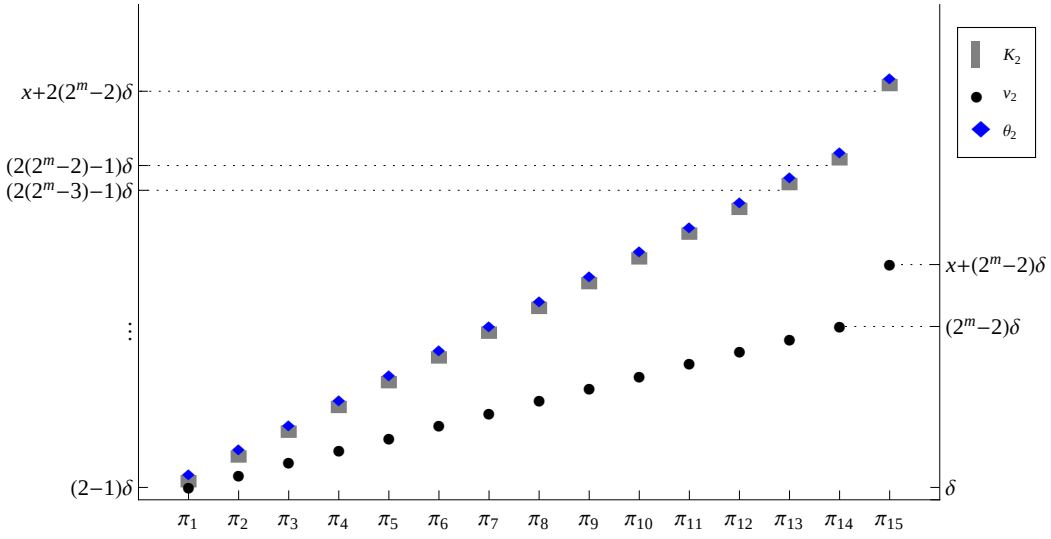
Step 2 (Appendix 2.C.2). We show that if player 1 has candidate set K_1 and reports v_1, and player 2 has candidate set K_2 and reports v_2 (while other players report 0), the fraction of the maximum social welfare that is guaranteed is at most the value stated in the theorem.

2.C.1 Construction of The Hard Instance

We construct two candidate sets K_1 and K_2 and two strategies v_1 and v_2 where, for $i = 1, 2$, K_i and v_i together satisfy the hypothesis of Lemma 2.11; we deduce that for our choices it holds that $v_1 \in \mathsf{UD}_1(K_1)$ and $v_2 \in \mathsf{UD}_2(K_2)$. These choices form our



(a) K_1 and v_1 with $v_1 \in \mathsf{UD}_1(K_1)$, and $\theta_1 \in K_1$



(b) K_2 and v_2 with $v_2 \in \mathsf{UD}_2(K_2)$, and $\theta_2 \in K_2$

Figure 2-2: The two hard instances constructed in Section 2.C.1 and the choice of true valuation made in Section 2.C.2, for the special case of $m = 4$.

candidate hard instance for the VCG mechanism. (We carry out the social welfare analysis in Section 2.C.2.)

Fix any labeling π over all $2^m - 1$ non-empty subsets of $[M]$ such that:

1. if $i < j$, then $\pi_i \not\supseteq \pi_j$ (i.e., π is proper, cf. Definition 2.9);
2. $\pi_i = \overline{\pi_{2^m-1-i}}$; and
3. $\pi_{2^m-1} = [M]$.

For instance, when $m = 3$ we can let $\pi = (\{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\})$. It is a simple exercise to prove that such a π exists for any $m \geq 2$.

Also fix any positive constant x (which should be thought of as a large constant).

We begin by choosing K_1 and v_1 (depending on π and x), and showing that $v_1 \in \mathsf{UD}_1(K_1)$:

Claim 2.24. *Choose:*

- *K_1 to be such that $K_1(\pi_i) = [x - \delta/2, x + \delta/2]$ for all $i \in \{1, \ldots, 2^m - 1\}$.*
- *v_1 to be such that $v_1(\pi_i) = x + (i-1)\delta$ for all $i \in \{1, \ldots, 2^m - 1\}$.*

(See Figure 2-2(a).) Then $v_1 \in \mathsf{UD}_1(K_1)$.

Proof. It suffices to verify that the assumptions in Lemma 2.11 hold. Indeed, $K_1^\perp$ and $K_1^\top$ are both weakly monotone because they are constant; v_1 is strictly monotonic since $v_1(\pi_i) < v_1(\pi_j)$ if $i < j$. If we choose $S' = S'' = \pi_1$, we definitely have $v_1(S') = x \leq x + \delta/2 = K_1^\top(S')$ and $v_1(S'') = x \geq x - \delta/2 = K_1^\perp(S'')$. Finally, we are left to verify (2.13), and we need a "witness labeling" for that. We simply choose π to be this labeling for which we have:

$$\forall\, i \in \{1, \ldots, 2^m - 2\}\,, \qquad v_1(\pi_i) - v_1(\pi_{i+1}) = -\delta = K_1^\perp(\pi_i) - K_1^\top(\pi_{i+1}) \ ,$$

and for $i = 2^m - 1$,

$$v_1(\pi_{2^m-1}) - v_1(\pi_1) > 0 > -\delta = K_1^\perp(\pi_{2^m-1}) - K_1^\top(\pi_1) \ .$$

This ends the proof that $v_1 \in \mathsf{UD}_1(K_1)$. □

Next, fixing any positive constant ε (which should be thought of as a small constant), we choose K_2 and v_2 (depending on π, x, and ε), and show that $v_2 \in \mathsf{UD}_2(K_2)$:

Claim 2.25. *Choose:*

- *K_2 to be such that*
$$K_2(\pi_i) = \big[(2i-1)\delta - \varepsilon, 2i\delta - \varepsilon\big]$$
for all $i \in \{1, \ldots, 2^m - 2\}$, and $K_2(\pi_{2^m-1})$ to be
$$K_2(\pi_{2^m-1}) = \big[x + 2(2^m - 2)\delta - \varepsilon, x + (2(2^m - 2) + 1)\delta - \varepsilon\big] \ .$$
- *v_2 to be such that $v_2(\pi_i) = i\delta - \varepsilon$ for all $i \in \{1, \ldots, 2^m - 2\}$, and $v_2(\pi_{2^m-1}) = x + (2^m - 2)\delta - \varepsilon$.*

(See Figure 2-2(b).) Then $v_2 \in \mathsf{UD}_2(K_2)$ owing to Lemma 2.11.

Proof. First, for sufficiently small ε, $K_2^{\perp}(\pi_i)$ and $K_2^{\top}(\pi_i)$ are both positive. Once again it suffices to verify that the assumptions in Lemma 2.11 hold. Indeed, $K_2^{\perp}$, $K_2^{\top}$, and v_2 are all strictly monotonic:

- $K_2^{\perp}(\pi_i) < K_2^{\perp}(\pi_j)$ for $i < j$,
- $K_2^{\top}(\pi_i) < K_2^{\top}(\pi_j)$ for $i < j$, and
- $v_2(\pi_i) < v_2(\pi_j)$ if $i < j$.

If we choose $S' = S'' = \pi_1$, we have $v_2(S') = \delta - \varepsilon < 2\delta - \varepsilon = K_2^{\top}(S')$ and $v_2(S'') = \delta - \varepsilon = K_2^{\perp}(S'')$. We are left to verify (2.13), and we need a "witness labeling" for that. We now choose the labeling that is the "reverse" of π, i.e., we let $\pi'_i = \pi_{2^m-i}$; for this choice of π' we have:

- for $2 \le i \le 2^m - 2$ and let $j = 2^m - 1 - i \in \{1, 2, \ldots, 2^m - 3\}$:

$$\begin{aligned} & v_2(\pi'_i) - v_2(\pi'_{i+1}) = v_2(\pi_{j+1}) - v_2(\pi_j) = \delta \\ = \ & \big((2(j+1)-1)\delta - \varepsilon\big) - \big(2j\delta - \varepsilon\big) \\ = \ & K_2^{\perp}(\pi_{j+1}) - K_2^{\top}(\pi_j) = K_2^{\perp}(\pi'_i) - K_2^{\top}(\pi'_{i+1}) \ , \end{aligned}$$

- for $i = 1$:

$$\begin{aligned} & v_2(\pi'_i) - v_2(\pi'_{i+1}) = v_2(\pi_{2^m-1}) - v_2(\pi_{2^m-2}) = x \\ = \ & K_2^{\perp}(\pi_{2^m-1}) - K_2^{\top}(\pi_{2^m-2}) = K_2^{\perp}(\pi'_i) - K_2^{\top}(\pi'_{i+1}) \ , \end{aligned}$$

- for $i = 2^m - 1$:

$$\begin{aligned} & v_2(\pi'_i) - v_2(\pi'_{i+1}) = v_2(\pi_1) - v_2(\pi_{2^m-1}) \\ = \ & -x - (2^m - 3)\delta > -x - (2(2^m - 2))\delta \\ = \ & K_2^{\perp}(\pi_1) - K_2^{\top}(\pi_{2^m-1}) = K_2^{\perp}(\pi'_i) - K_2^{\top}(\pi'_{i+1}) \ . \end{aligned}$$

This ends the proof that $v_2 \in \mathsf{UD}_2(K_2)$ owing to Lemma 2.11. □

2.C.2 Putting Things Together

Let the first two players respectively have candidate sets K_1 and K_2 and play the undominated strategies v_1 and v_2 (from Claim 2.24 and Claim 2.25, and see also Figure 2-2); let the rest of the players have valuation 0 and report 0 (which is an undominated strategy for each such player).

We make the following observations:

- When the players report $v \stackrel{\text{def}}{=} (v_1, v_2, 0, \ldots, 0)$, the VCG mechanism will always choose the allocation $A = ([M], \varnothing, \ldots, \varnothing)$.

  Indeed, the social welfare of A relative to v is

  $$v_1([M]) = v_1(\pi_{2^m-1}) = x + (2^m - 2)\delta \ .$$

  On the other hand, for any allocation giving $\pi_i \neq \varnothing$ to player 1 and $\pi_{2^m-1-i} = \overline{\pi_i}$ to player 2, the social welfare relative to v is equal to

  $$v_1(\pi_i) + v_2(\pi_{2^m-1-i}) = (x + (i-1)\delta) + (2^m - 1 - i)\delta - \varepsilon = x + (2^m - 2)\delta - \varepsilon \ ,$$

which is smaller than that achieved by A; furthermore, for any allocation giving $\varnothing$ to player 1 and $[M]$ to player 2, the social welfare relative to v is equal to

$$v_2([M]) = v_2(\pi_{2^m-1}) = x + (2^m - 2)\delta - \varepsilon \ ,$$

which again is also smaller than that achieved by A.

- Assume that we pick the true valuation $\theta_1 \in K_1$ for player 1 to be such that $\theta_1(S) = x$ for all non-empty S, and $\theta_2 \in K_2$ for player 2 to be such that $\theta_2(S) = K_2^\top(S)$. Of course, we can only choose $\theta_i(S) = 0$ for all other players $i > 2$. (See Figure 2-2)

- The true social welfare on allocation A is $\theta_1([M]) = x$.

- The maximum social welfare is instead the following:

$$\mathrm{MSW}(\theta) \geq \theta_2([M]) = K_2^\top(\pi_{2^m-1}) = x + (2(2^m - 2) + 1)\delta - \varepsilon \ .$$

- Hence, the obtained social welfare compared to the maximum social welfare in this case is

$$\mathrm{SW}(\theta, \mathsf{VCG}(v)) = x \leq \mathrm{MSW}(\theta) - (2(2^m - 2) + 1)\delta + \varepsilon \ .$$

By choosing $\varepsilon > 0$ sufficiently small, the social welfare guarantee of the VCG mechanism is at most

$$\mathrm{MSW}(\theta) - (2^{m+1} - 3)\delta \ .$$

This finishes the proof of (2.23), the worst-case choice of θ for Theorem 2.1b.

For the best-case choice of θ, we observe that for the same choice of v_1, v_2, K_1, K_2, A:

- The true social welfare on allocation A is $\theta_1([M]) \leq x + \delta/2$.

- The maximum social welfare is instead the following:

$$\mathrm{MSW}(\theta) \geq \theta_2([M]) \geq K_2^\perp(\pi_{2^m-1}) = x + 2(2^m - 2)\delta - \varepsilon \ .$$

- Hence, the obtained social welfare compared to the maximum social welfare in this case is

$$\mathrm{SW}(\theta, \mathsf{VCG}(v)) \leq x + \delta/2 \leq \mathrm{MSW}(\theta) - 2(2^m - 2)\delta + \delta/2 + \varepsilon \ .$$

By choosing $\varepsilon > 0$ sufficiently small, the social welfare guarantee of the VCG mechanism is at most

$$\mathrm{MSW}(\theta) - (2^{m+1} - 5)\delta \ .$$

This finishes the proof of (2.22), the best-case choice of θ for Theorem 2.1b. ■

2.D Theorem 2.2 with Mixed Strategies

In this section we prove an analogue of Theorem 2.2 for mixed strategies, as follows.

Theorem 2.2′. *In a combinatorial Knightian auction with n players and m goods, let the VCG mechanism break ties by preferring subsets with smaller cardinalities.*[21] *Then, for all δ, all products K of δ-approximate candidate sets, all profiles $\theta \in K$, all profiles of mixed strategies $\sigma \in \mathsf{RM}^{\mathsf{mix}}(K)$, and all $p \geq 1$, we have with probability at least $1 - 1/p$ over the choices of v from σ:*

$$\mathrm{SW}(\theta, \mathsf{VCG}(v)) \geq \mathrm{MSW}(\theta) - O(n^2 p) \cdot \delta \ .$$

(This result can be tightened to $O(n \log n \log(1/p) \cdot \delta)$ either when (1) players are restricted to consider only monotone valuations (i.e., $\theta_i(S) \leq \theta_i(T)$ for any $S \subseteq T$), or when (2) players are studying $\mathsf{RM}^{\mathsf{mix}}(\mathsf{UD}(K))$ strategies, rather than just $\mathsf{RM}^{\mathsf{mix}}(K)$.)

Before proving this theorem, we first illustrate why the result is very different from that of Theorem 2.2.

2.D.1 Why Allowing Mixed Strategies Yields a Different Result

When a regret-minimizing player considers mixed strategies, he may significantly deviate (in expectation) from his candidate set. (This stands in contrast to the pure-strategy case, where he may deviate by at most δ; cf. Claim 2.8.) In fact, deviating may happen even in a single-good auction.

An Example in a Single-Good Auction. Let i be a player with candidate set $K_i = [x, x + \delta]$ in a single-good (Knightian) auction. One can carefully verify that his minimum regret is at most $\frac{\delta}{4}$, obtained by a mixed strategy of bidding uniformly at random between x and $x + \delta$. However, we state without proof that the following mixed strategy σ_i also provides a regret of $\frac{\delta}{4}$:

$$\sigma_i = \begin{cases} \text{drawn uniformly at random from } [x, x + \frac{3}{4}\delta] & \text{w.p. } \frac{3}{4}; \\ x + t\delta, & \text{w.p. } \frac{1}{4}(\frac{1}{t} - \frac{1}{t+1}) \text{ where } t \in \mathbb{Z}_+. \end{cases} \tag{2.24}$$

Note that the expected bidding value $\mathbb{E}[\sigma_i] = +\infty$ is unbounded from above, and one can similarly construct a strategy in which player i arbitrarily (in expectation) underbids. This destroys the hope of using linearity of expectation to deduce the mixed-strategy case as a corollary of the pure-strategy one.

However, any such deviation always satisfies the probabilistic guarantee $\Pr[\sigma_i \geq x + t\delta] \leq \frac{1}{4t}$ for overbidding (and similarly, underbidding), resulting in the simple conclusion that, with constant probability, none of the n players over/underbids by

[21] If giving subsets A or $B \subsetneq A$ to player i provides the same social welfare, then the VCG will give B to player i.

more than $O(n\delta)$. The social welfare is therefore affected by at most $O(n^2\delta)$ in a single-good auction.[22]

A Harder Problem in Combinatorial Auctions. In combinatorial auctions with m goods, each player reports $2^m - 1$ values on each of the $2^m - 1$ non-empty subsets of $[m]$. Thus, a player may (in principle) choose to independently overbid or underbid each of his $2^m - 1$ coordinates, according to (2.24). If so, then, with constant probability, he may choose to (a) overbid by $O(2^m\delta)$ on one of his coordinates, and (b) underbid by $O(2^m\delta)$ on another.

This possibility complicates the analysis, because such a choice of strategy may lead to a social welfare loss of $O(2^m\delta)$. Interestingly, we show that (a) cannot happen, but (b) can. However, when (b) happens, the social welfare is not going to be affected much.

2.D.2 Proof of Theorem 2.2′

Proof. We begin by explicitly writing down the formulation of the (maximum) regret in (2.4) for mixed strategies. Given a candidate set K_i of player i, and a possibly mixed strategy σ_i from which his bidding strategy v_i is drawn, the (expected maximum) regret of σ_i for player i is

$$R_i(K_i, \sigma_i) = \max_{\theta_i \in K_i} \max_{v_{-i}} \Big(\mathrm{MSW}\big(\theta_i, v_{-i}\big) - \mathbb{E}_{v_i \sim \sigma_i} \big[\mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big)\big] \Big) \ . \tag{2.25}$$

We also recall the following notations. For each player i, each candidate set $K_i \subset \Theta_i$, and each subset $T \subseteq [m]$, we let

$$K_i(T) \stackrel{\text{def}}{=} \{\theta_i(T)\}_{\theta_i \in K_i}, \quad K_i^{\perp}(T) \stackrel{\text{def}}{=} \inf K_i(T),$$
$$K_i^{\top}(T) \stackrel{\text{def}}{=} \sup K_i(T), \quad K_i^{\mathrm{mid}}(T) \stackrel{\text{def}}{=} (K_i^{\perp}(T) + K_i^{\top}(T))/2 \ .$$

For the same reason as Footnote 13 on page 53 in the main paper, we assume without loss of generality that for each T, the minimum/maximum point in $K_i(T)$ exists. That is, $K_i^{\perp}(T) \stackrel{\text{def}}{=} \min K_i(T)$ and $K_i^{\top}(T) \stackrel{\text{def}}{=} \max K_i(T)$.

We first note that Claim 2.7 continues to hold:[23]

Claim 2.7. *Let v_i be a strategy of player i such that $v_i(T) = K_i^{\mathrm{mid}}(T)$ for each non-empty $T \subseteq [M]$. Then $R_i(K_i, v_i) \le \delta$.*

We now prove some properties about an arbitrary (possibly mixed) strategy σ_i of player i with regret $\le \delta$.

Player Underbidding

[22] A more careful analysis leads to $O(n \log n \cdot \delta)$.

[23] We note that when mixed strategies are allowed, one can find a strategy with regret $\delta/2$, therefore bidding the mid-points, having a regret δ, is no longer a regret-minimizing strategy. Since the remaining proof of Theorem 2.2′ only requires to know that 'the regret-minimizing strategy has a regret $O(\delta)$', it suffices to analyze the mid-points, losing a constant factor of 2.

We first show a variant of Claim 2.8a from the main paper. It is a probabilistic bound on how a player i may underbid on each of his $2^m - 1$ coordinates:

Claim 2.26 (player underbidding)**.** *Let σ_i be a (possibly mixed) strategy of player i such that $R_i(K_i, \sigma_i) \leq \delta$. Then, for any non-empty subset $T \subseteq [M]$, and any real number $t \geq 1$,*

$$\Pr_{v_i \sim \sigma_i}\left[K_i^\top(T) - \max_{T' \subseteq T} v_i(T') > t \cdot \delta\right] \leq \frac{1}{t} \ .$$

Proof. Suppose the claim is not true. Then, there exists T such that

$$\Pr_{v_i \sim \sigma_i}\left[K_i^\top(T) - \max_{T' \subseteq T} v_i(T') > t \cdot \delta\right] > \frac{1}{t} \ . \tag{2.26}$$

We contradict our assumption on v_i by showing $R_i(K_i, \sigma_i) > \delta$.

To show $R_i(K_i, \sigma_i) > \delta$, as per (2.25), we must find some v_{-i} and some θ_i so that

$$\mathrm{MSW}\big(\theta_i, v_{-i}\big) - \mathbb{E}_{v_i \sim \sigma_i}\big[\mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big)\big] > \delta \tag{2.27}$$

Let j be an arbitrary player other than i. We choose $\theta_i \in K_i$ such that $\theta_i(T) = K_i^\top(T)$ and v_{-i} as follows: for every $S \subseteq [m]$

$$v_j(S) \stackrel{\text{def}}{=} \begin{cases} H & \text{if } S = \overline{T} \\ H + (K_i^\top(T) - t \cdot \delta) & \text{if } S = [M] \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad v_k(S) \stackrel{\text{def}}{=} 0 \text{ for every } k \notin \{i, j\}.$$

Above, H is some huge real number (i.e., much bigger than $v_i(S)$ for any subset S).[24]

Recall that (2.26) tells us that, with probability more than $\frac{1}{t}$ over the choice of v_i from σ_i, the event $K_i^\top(T) - \max_{T' \subseteq T} v_i(T') > t \cdot \delta$ occurs. Let us denote by $\textsc{Event}(v_i)$ this event, and it is not hard to verify that $\textsc{Event}(v_i)$ implies that the outcome $\mathsf{VCG}(v_i, v_{-i})$ must allocate $\varnothing$ to player i, and $[M]$ to player j. Therefore, with probability more than $\frac{1}{t}$, we have

$$\mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) = \theta_i(\varnothing) + v_{-i}([m]) = H + K_i^\top(T) - t \cdot \delta \ .$$

On the other hand, $\mathrm{MSW}(\theta_i, v_{-i}) \geq \theta_i(T) + v_{-i}(\overline{T}) = K_i^\top(T) + H$, and therefore

$$\begin{aligned} &\mathbb{E}_{v_i \sim \sigma_i}\big[\mathrm{MSW}\big(\theta_i, v_{-i}\big) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big)\big] \\ &\geq \Pr_{v_i \sim \sigma_i}\left[\textsc{Event}(v_i)\right] \cdot \mathbb{E}_{v_i \sim \sigma_i}\left[\mathrm{MSW}(\theta_i, v_{-i}) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) \,\middle|\, \textsc{Event}(v_i)\right] \\ &> \frac{1}{t} \cdot \Big(K_i^\top(T) + H - \big(H + (K_i^\top(T) - t \cdot \delta)\big)\Big) = \delta \ . \end{aligned}$$

This proves (2.27) and concludes concludes the proof of Claim 2.26. □

We remark here that the above proof matches our high level description in Appendix 2.D.1. That is, since a player may have different valuations on all of his $2^m - 1$ coordinates,

[24] Notice that when $T = [M]$ we have $\overline{T} = \varnothing$, and one cannot assign $v_j(\varnothing)$ to be a nonzero number. In that case, we can choose $H = 0$ and v_j remains well-defined, since we must have $K_i^\top(T) - t \cdot \delta > 0$ (as otherwise $K_i^\top(T) - \max_{T' \subseteq T} v_i(T') > t \cdot \delta$ cannot hold, contradicting our assumption). The rest of the proof still goes through.

he may choose to independently underbid each of his $2^m - 1$ coordinates according to Claim 2.27 (which is tight, due to an example generalizing (2.24) to allow multiple goods). If so, with constant probability (using union bound), he may underbid by $O(2^m\delta)$ on one of his $2^m - 1$ coordinates.

Could this large underbidding destroy the social welfare by $O(2^m\delta)$? Our answer is No (as we shall formally explain later) because, if, in the maximum social welfare allocation, player i receives a subset $B_i \subseteq [M]$ of the goods, all we need to learn from the player's underbidding is: *how much will player i underbid on coordinate B_i?* Therefore, we do not care how much he underbids on other coordinates, and therefore this 2^m factor does not show up in the social welfare loss.

Player Overbidding

The overbidding case is much harder. In fact, one can (essentially) show a similar coordinate-wise argument as in Claim 2.26, and conclude that a player will overbid on each of his coordinates by at most $t \cdot \delta$, with probability at most $\frac{1}{t}$. Via a union bound, this implies that, with constant probability, he may overbid by $O(2^m\delta)$ on *one* of his $2^m - 1$ coordinates. If this happens, unlike the underbidding case, the social welfare performance will be very poor. The following example illustrates this point.

> EXAMPLE. Consider a 2-player auction with m goods, where m is even. The first player is only interested in the subsets of $[m]$ that have cardinality $m/2$, and his value for each such subset lies in the interval $[x, x+\delta]$. The second player is only interested in the set of all goods, $[m]$, which he values precisely $x + (\binom{m}{m/2} - 1)\delta$. Notice that the maximum social welfare in this setting is $x + (\binom{m}{m/2} - 1)\delta$. Also notice that, in such an auction, at most one player 'wins'. That is, at most one player can be allocated a subset of $[m]$ which he positively values.
>
> Now suppose that player 2 reports his true valuation, while player 1 overbids as follows. Let $t = \binom{m}{m/2}$. For each of the t subsets he is interested in, player 1 reports, independently and with probability $1/t$, the value $x + t \cdot \delta$, and x otherwise. (For each subset he is not interested in, player 1 reports 0.) Then, with constant probability, player 1 reports $x + t \cdot \delta$ on one of his coordinates, and thus 'wins' the auction. Note that, when player 1 'wins', the social welfare is at most $x + \delta$ and misses the maximum social welfare by $(t-2) \cdot \delta = \tilde{\Omega}(2^m\delta)$.

Therefore, to prove a good social-welfare performance, it is not advisable to bound a player's overbidding *coordinate-wise*. In fact, we prove the following claim, which is significantly different from what we showed in Claim 2.8b for the pure case. The new claim essentially bounds how a player i may overbid (on all coordinates) with respect to a given mixed strategy sub-profile σ_{-i} of his opponents. Since we will eventually be interested in only *one* particular σ_{-i} —namely, the one when all players other than i are playing regret-minimizing strategies— we do not need to pay for the extra $O(2^m\delta)$ loss in the union bound.

Claim 2.27 (player overbidding)**.** *Let σ_i be a (possibly mixed) strategy of player i such that $R_i(K_i, \sigma_i) \leq \delta$, σ_{-i} an arbitrary (possibly mixed) strategy sub-profile of his opponents, and $\theta_i \in K_i$ his possible true valuation. Then, for any real number $t \geq 1$,*

$$\Pr_{\substack{v_i \sim \sigma_i \\ v_{-i} \sim \sigma_{-i}}} \Big[v_i\big(\mathsf{VCG}(v_i, v_{-i})\big) > \theta_i\big(\mathsf{VCG}(v_i, v_{-i})\big) + 4t \cdot \delta \Big] \leq \frac{1}{t} \ . \tag{2.28}$$

Proof. Suppose the claim is not true and there are choices of σ_i, σ_{-i}, and θ_i, such that the above probability is strictly larger than $\frac{1}{t}$. We denote by $\textsc{Event}_1(v_i, v_{-i})$ the probabilistic event that $v_i\big(\mathsf{VCG}(v_i, v_{-i})\big) > \theta_i\big(\mathsf{VCG}(v_i, v_{-i})\big) + 4t \cdot \delta$, and we want to show that if $\Pr[\textsc{Event}_1] > \frac{1}{t}$, then $R_i(K_i, \sigma_i) > \delta$, contradicting our assumption on σ_i. To achieve this, we lower bound (2.25) (using the same choice of θ_i provided in the assumption of this claim) by a probabilistic form:

$$R_i(K_i, \sigma_i) \geq \mathbb{E}_{v^*_{-i} \sim \sigma^*_{-i}} \Big[\mathrm{MSW}\big(\theta_i, v^*_{-i}\big) - \mathbb{E}_{v_i \sim \sigma_i} \big[\mathrm{SW}\big((\theta_i, v^*_{-i}), \mathsf{VCG}(v_i, v^*_{-i})\big) \big] \Big] \ . \tag{2.29}$$

Now it suffices to choose a witness distribution σ^*_{-i} so that the right-hand side is larger than δ.

We choose σ^*_{-i} as follows. It is reconstructed from the distribution σ_{-i} given in the assumption, with every occurrence of $v_{-i} \sim \sigma_{-i}$ replaced by v^*_{-i} with the same probability, where v^*_{-i} is defined as:

$$\forall j \neq i \, \forall S \subseteq [m] \quad v^*_j(S) \stackrel{\text{def}}{=} \begin{cases} \mathrm{MSW}(\theta_i, v_{-i}) + 2t \cdot \delta & \text{if } S = [M] \\ v_j(S) & \text{otherwise} \end{cases} \ .$$

Now assuming, by way of contradiction, that the desired regret term $R_i(K_i, \sigma_i) \leq \delta$, which implies (using (2.25) for v_{-i} drawn from σ_{-i}):

$$\mathbb{E}_{v_{-i} \sim \sigma_{-i}} \Big[\mathrm{MSW}\big(\theta_i, v_{-i}\big) - \mathbb{E}_{v_i \sim \sigma_i} \big[\mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) \big] \Big] \leq R_i(K_i, \sigma_i) \leq \delta \ .$$

Using Markov bound, with probability at least $1/2t$ over the choices of $v_i \sim \sigma_i$ and $v_{-i} \sim \sigma_{-i}$, we have

$$\mathrm{MSW}\big(\theta_i, v_{-i}\big) - \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) \leq 2t \cdot \delta$$

We denote by $\textsc{Event}_2(v_i, v_{-i})$ the probabilistic event such that the above inequality is true. From (2.28), we know that with probability strictly larger than $1/t - 1/2t = 1/2t$ we have that both $\textsc{Event}_1$ and $\textsc{Event}_2$ happen, and therefore

$$\begin{aligned}
& v_i\big(\mathsf{VCG}(v_i, v_{-i})\big) > \theta_i\big(\mathsf{VCG}(v_i, v_{-i})\big) + 4t \cdot \delta && \text{(using } \textsc{Event}_1\text{)} \\
\implies \ & v_i\big(\mathsf{VCG}(v_i, v_{-i})\big) + v_{-i}\big(\mathsf{VCG}(v_i, v_{-i})\big) > \theta_i\big(\mathsf{VCG}(v_i, v_{-i})\big) + v_{-i}\big(\mathsf{VCG}(v_i, v_{-i})\big) + 4t \cdot \delta \\
\implies \ & \mathrm{SW}\big((v_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i}\big) > \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) + 4t \cdot \delta \\
\implies \ & \mathrm{MSW}(v_i, v_{-i}) > \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) + 4t \cdot \delta \\
\implies \ & \mathrm{MSW}(v_i, v_{-i}) > \mathrm{MSW}(\theta_i, v_{-i}) + 2t \cdot \delta && \text{(using } \textsc{Event}_2\text{)} \\
\implies \ & \mathrm{MSW}(v_i, v_{-i}) > v^*_j([M]) && (\forall j \neq i, \text{ using the definition of } v^*_{-i} \ .)
\end{aligned}$$

The last strict inequality implies that the allocation under $\mathsf{VCG}(v_i, v_{-i})$ must be the same as $\mathsf{VCG}(v_i, v^*_{-i})$. This is because v^*_{-i} is only different from v_{-i} on the coordinates

$[M]$ for players $j \neq i$, but those coordinates only incur a smaller social welfare than $\mathsf{VCG}(v_i, v_{-i})$ according to the last inequality above.

In sum, we have that $\mathrm{MSW}(\theta_i, v^*_{-i}) \geq v_{-i}([M]) = \mathrm{MSW}(\theta_i, v_{-i}) + 2t \cdot \delta$; however, under $\textsc{Event}_1 \wedge \textsc{Event}_2$, the obtained social welfare can be upper bounded as follows:

$$\begin{aligned}\mathrm{SW}\big((\theta_i, v^*_{-i}), \mathsf{VCG}(v_i, v^*_{-i})\big) &= \mathrm{SW}\big((\theta_i, v^*_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) = \mathrm{SW}\big((\theta_i, v_{-i}), \mathsf{VCG}(v_i, v_{-i})\big) \\ &\leq \mathrm{MSW}(\theta_i, v_{-i}) \leq \mathrm{MSW}(\theta_i, v^*_{-i}) - 2t \cdot \delta \ .\end{aligned}$$

Above, the first equality is because $\mathsf{VCG}(v_i, v_{-i})$ produces the same allocation as $\mathsf{VCG}(v_i, v^*_{-i})$; the second equality is because $\mathsf{VCG}(v_i, v^*_{-i})$ never gives all the goods to a player $j \neq i$; and the first inequality is because, by definition, the VCG maximizes social welfare.

Now we go back to (2.29), and show that $R_i(K_i, \sigma_i) > \delta$:

$$\begin{aligned}R_i(K_i, \sigma_i) &\geq \mathbb{E}_{\substack{v_i \sim \sigma_i \\ v^*_{-i} \sim \sigma^*_{-i}}} \Big[\mathrm{MSW}\big(\theta_i, v^*_{-i}\big) - \mathrm{SW}\big((\theta_i, v^*_{-i}), \mathsf{VCG}(v_i, v^*_{-i})\big)\Big] \\ &\geq \Pr_{\substack{v_i \sim \sigma_i \\ v^*_{-i} \sim \sigma^*_{-i}}} [\textsc{Event}_1 \wedge \textsc{Event}_2] \times \\ &\quad \mathbb{E}_{\substack{v_i \sim \sigma_i \\ v^*_{-i} \sim \sigma^*_{-i}}} \Big[\mathrm{MSW}\big(\theta_i, v^*_{-i}\big) - \mathrm{SW}\big((\theta_i, v^*_{-i}), \mathsf{VCG}(v_i, v^*_{-i})\big) \Big| \textsc{Event}_1 \wedge \textsc{Event}_2\Big] \\ &> \frac{1}{2t} \times 2t \cdot \delta = \delta \ .\end{aligned}$$

The above conclusion contradicts our assumption that the regret of the mixed strategy σ_i is at most δ. This concludes the proof of the claim. □

Putting It All Together

Now we go back to the proof of Theorem 2.2. Let $\sigma = (\sigma_1, \ldots, \sigma_n) \in \mathsf{RM}^{\mathsf{mix}}(K)$ be a profile of regret-minimizing mixed strategies, and let $\theta \in K$ be any valuation profile. Since there exists a strategy with regret $\leq \delta$ for each player (see Claim 2.7), we must have $R_i(K_i, \sigma_i) \leq \delta$ to satisfy the assumption of Claim 2.26 and 2.27.

Now, letting $(B_0, B_1, \ldots, B_n)$ be the allocation that maximizes the social welfare under θ, we are ready to compute the social welfare guarantee. For any choice of $v \sim \sigma$, let X_i denote the non-negative probabilistic variable equal to the difference $v_i\big(\mathsf{VCG}(v)\big) - \theta_i\big(\mathsf{VCG}(v)\big)$; according to Claim 2.27, we have $\Pr[X_i > 4t\delta] < \frac{1}{t}$. Also let Y_i denote the non-negative probabilistic variable equal to the difference $K_i^\top(B_i) - \max_{T' \subseteq B_i} v_i(T')$, and, according to Claim 2.26 (for the choice of $T = B_i$), we have $\Pr[Y_i > t\delta] \leq \frac{1}{t}$.

$$\begin{aligned}&\mathrm{SW}(\theta, \mathsf{VCG}(v)) = \sum_{i=1}^n \theta_i(\mathsf{VCG}(v)) = \sum_{i=1}^n v_i(\mathsf{VCG}(v)) - \sum_{i=1}^n X_i \\ \geq\ & \sum_{i=1}^n \max_{T' \subseteq B_i} v_i(T') - \sum_{i=1}^n X_i \quad \text{(because the VCG maximizes social welfare under } v\text{)}\end{aligned}$$

$$= \sum_{i=1}^{n} K_i^{\top}(B_i) - \sum_{i=1}^{n}(X_i + Y_i)$$
$$\geq \sum_{i=1}^{n} \theta_i(B_i) - \sum_{i=1}^{n}(X_i + Y_i) = \mathrm{MSW}(\theta) - \sum_{i=1}^{n}(X_i + Y_i) \ .$$

We are now left to bound $\sum_{i=1}^{n}(X_i + Y_i)$. For any $p \geq 1$ and each choice of $i \in [n]$, with probability at least $1 - \frac{1}{2np}$, we have that $X_i \leq (8np)\delta$, and, with probability at least $1 - \frac{1}{2np}$, $Y_i \leq (2np)\delta$. Using union bound, with a total probability of at least $1 - \frac{1}{p}$ (over the choices of v from σ), we have $X_i \leq (8np)\delta$ and $Y_i \leq (2np)\delta$ *for all* $i \in [n]$. In such a case the above difference satisfies

$$\mathrm{SW}(\theta, \mathsf{VCG}(v)) \geq \mathrm{MSW}(\theta) - O(n^2 p) \cdot \delta \ .$$

This concludes the proof of Theorem 2.2′. ■

Chapter 3

Bridging Utility Maximization and Regret Minimization

We relate the strategy sets that a player ends up with after refining his own strategies according to two very different models of rationality: namely, utility maximization and regret minimization.

3.1 Introduction

Rational players have been modeled in two main ways.

- A utility-maximizing player $\mathcal{U}$ eliminates all his dominated strategies to compute his set of undominated ones, UD. Notice that $\mathcal{U}$ cannot further refine UD based on utility maximization. If UD consists of a single strategy s (necessarily a dominant one), then $\mathcal{U}$ of course chooses s. *But, if* UD *contains multiple strategies, which one should* $\mathcal{U}$ *choose?*
- A regret-minimizing player $\mathcal{R}$ eliminates all his non regret-minimizing strategies so as to compute his set of regret-minimizing strategies, RM. He might even continue this process k times, until he is satisfied or no further elimination is possible. Let us denote the final set of strategies he obtains this way by RM^k. If RM^k consists of a single strategy s, $\mathcal{R}$ of course chooses s. *But, if* RM^k *contains multiple strategies, which one should* $\mathcal{R}$ *choose?*

In both cases, "a random strategy" or "the lexicographic first strategy" are certainly possible answers. But another answer is that, when he is 'no longer able to apply his favorite way of reasoning', even a die-hard utility maximizer $\mathcal{U}$ will resort to regret minimization to refine UD, and even a die-hard regret minimizer $\mathcal{R}$ will resort to utility maximization to refine RM^k. In principle, the two final sets of strategies obtained by such different refinement procedures could be vastly different. Our next structural theorem, however, guarantees that they coincide.

Abusing notation a bit, consider UD and RM also to be "operators" acting on sets of strategies. In this case $\mathsf{UD}(\mathsf{UD}) = \mathsf{UD}$, while $\mathsf{RM}^2 \stackrel{\text{def}}{=} \mathsf{RM}(\mathsf{RM})$ may be a strict

subset of RM. Then, we prove that the set of strategies obtained after applying, in arbitrary order, k times the operator RM and at least once the operator UD coincides with $\mathsf{RM}^k \cap \mathsf{UD}$. For instance,

$$\mathsf{RM}(\mathsf{RM}(\mathsf{UD}(\mathsf{RM}(\mathsf{RM}(\mathsf{UD}))))) = \mathsf{RM}^4(\mathsf{UD}) = \mathsf{RM}^4 \cap \mathsf{UD}.$$

After recalling the relevant notions, we prove our theorem for pure strategies, and then point out its simple but interesting implications for mechanism design. Finally, we point out that our result extends to mixed strategies as well.

We recall that regret-minimizing strategies are also known as regret-minimax strategies. The suggestion of adopting regret-minimizing strategies traces back to Savage's reading [138] of the work of Wald [161], and has been axiomatized by Milnor [109]. The notion of regret has been treated differently in different settings. A unified axiomatic characterization of minimax regret has been recently given by Stoye [155].

Many empirical studies compare utility maximizers and regret minimizers, see for instance Chorus, Arentze and Timmermans [45], and Hensher, Greene and Chorus [75]. Recently, Engelbrecht-Wiggans and Katok [59] and Filiz and Ozbay [62] provide experimental evidence for regret in first- and second-price auctions.

To the best of our knowledge, we are the first to study players who use regret for refining their sets of undominated strategies.

3.2 Basic Notions

To state and prove our result, we use the language of decision theory: namely, envisaging "a single player against Nature".[1]

Let $\mathcal{S}$ be a compact set of (pure) strategies of the player, and T a compact set of states of Nature.[2] We denote by U the (continuous) utility function of the player, where $U(s,t)$ is the utility under strategy $s \in \mathcal{S}$ when Nature's state is $t \in T$. Regret-minimizing strategies and undominated strategies are defined as follows:

- Given a *menu* $S \subseteq \mathcal{S}$ of strategies, the player's (maximum) regret for a strategy $s \in S$ in menu S, denoted by $R_S(s)$, is the maximum difference, taken over all possible Nature's states $t \in T$, between the utility the player gets by playing s, and that he could have gotten by "best responding" to t; formally,

$$R_S(s) \stackrel{\text{def}}{=} \max_{t \in T} \big(\max_{s^* \in S} U(s^*, t) - U(s,t) \big).$$

Therefore, the *regret-minimizing strategies* with respect to a menu $S \subseteq \mathcal{S}$, denoted

[1] Results for n-player (strategic or pre-Bayesian) games follow as corollaries. This is because the definitions of dominance and regret are universally quantified over other players' strategies, which can be treated as Nature's strategies.

[2] Both $\mathcal{S}$ and T may be infinite, and $\mathcal{S}$ may be convex in order to allow arbitrary mixed strategies to be considered.

by $\mathsf{RM}(S)$, is the set of strategies that minimize the regret:

$$\mathsf{RM}(S) \stackrel{\text{def}}{=} \arg\min_{s \in S} R_S(s).$$

- Given two strategies $s, s' \in \mathcal{S}$, by definition s' *weakly dominates* s, denoted by $s' \succ s$, if

$$\forall t \in T,\ U(s',t) \geq U(s,t) \quad \text{and} \quad \exists t \in T,\ U(s',t) > U(s,t)\ .$$

Given a *menu* $S \subseteq \mathcal{S}$ of strategies, the player's *undominated strategies* consist of those that are not weakly dominated by any weakly undominated strategy.[3] Formally,

$$\begin{aligned}\mathsf{UD}(S) &\stackrel{\text{def}}{=} S \setminus \{s \in S\ :\ \exists s' \in S \text{ s.t. } (s' \succ s) \wedge (\nexists s'' \in S, s'' \succ s')\} \\ &= \quad \{s \in S\ :\ \nexists s' \in S \text{ s.t. } (s' \succ s) \wedge (\nexists s'' \in S, s'' \succ s')\}.\end{aligned}$$

We now state two simple facts which easily follow from the above definitions:

Fact 3.1. *For any menu $\tilde{S} \subseteq \mathcal{S}$,*

(a) if $s \prec s'$ for some $s, s' \in \tilde{S}$, then $R_{\tilde{S}}(s) \geq R_{\tilde{S}}(s')$, and

(b) the regret values of a strategy with respect to $\tilde{S}$ and $\mathsf{UD}(\tilde{S})$ are the same, namely:[4]

$$R_{\tilde{S}}(s) = \max_{t \in T}\Big(\max_{s^* \in \tilde{S}} U(s^*,t) - U(s,t)\Big) = \max_{t \in T}\Big(\max_{s^* \in \mathsf{UD}(\tilde{S})} U(s^*,t) - U(s,t)\Big) = R_{\mathsf{UD}(\tilde{S})}(s)\ .$$

Note that regret minimization is mostly studied when a player has beliefs about his opponents. In particular, the notions from Hyafil and Boutilier [76] and Renou and Schlag [134] coincide with ours when the players do not form beliefs about their opponents —or, in our language, Nature.

3.3 Result

Established our language, we prove our theorem as a corollary of the following lemma.

Lemma 3.2. *For any menu $S \subseteq \mathcal{S}$, $\mathsf{UD}(\mathsf{RM}(S)) = \mathsf{RM}(\mathsf{UD}(S)) = \mathsf{RM}(S) \cap \mathsf{UD}(S)$.*

Proof. We divide the proof into six steps:

1. $\mathsf{RM}(\mathsf{UD}(S)) \subseteq \mathsf{RM}(S)$.

[3] In general, weakly undominated strategies do not coincide with undominated ones. As argued by Jackson [79], it may happen that every pure strategy is weakly dominated by another one in an infinite chain, and in such a case all strategies are undominated but weakly dominated. However, in many cases of interest (e.g., when the set of pure strategies is finite, or when the mechanism is *bounded*), weakly undominated strategies coincide with undominated ones.

[4] The equality in the middle is since any strategy $s^* \in \tilde{S} \setminus \mathsf{UD}(\tilde{S})$ must be weakly dominated by some $s^{**} \in \tilde{S}$, giving at least as good utilities as s^* for *any* $t \in T$. Therefore, such choices of s^{**} can be ignored in the inner max.

For any $s \in \mathsf{RM}(\mathsf{UD}(S))$, we show that $s \in \mathsf{RM}(S)$ by proving that s has minimum regret among all strategies in S. Indeed:

- For any other strategy $s' \in \mathsf{UD}(S)$, it holds that $R_{\mathsf{UD}(S)}(s) \leq R_{\mathsf{UD}(S)}(s')$. By Fact 3.1b, we deduce that $R_S(s) \leq R_S(s')$.
- For any other strategy $s' \in S \setminus \mathsf{UD}(S)$, it holds that $s' \prec s''$ for some $s'' \in \mathsf{UD}(S)$ and $R_S(s) \leq R_S(s'')$. By Fact 3.1a, we deduce that $R_S(s) \leq R_S(s'') \leq R_S(s')$.

2. $\mathsf{RM}(\mathsf{UD}(S)) \subseteq \mathsf{UD}(\mathsf{RM}(S))$.

   Given that $\mathsf{RM}(\mathsf{UD}(S)) \subseteq \mathsf{RM}(S)$ (proved above), if there is some $s \in \mathsf{RM}(\mathsf{UD}(S))$ with $s \notin \mathsf{UD}(\mathsf{RM}(S))$, then s must be weakly dominated by some other strategy $s' \in \mathsf{RM}(S)$, namely $s \prec s'$, but s' cannot be weakly dominated by any other strategy in $\mathsf{RM}(S)$, by definition of UD.

   Now we show that s' cannot be weakly dominated by any strategy in S as well. Suppose not, that is $s' \prec s''$ where $s'' \in S$. Then $s'' \notin \mathsf{RM}(S)$ as we have just argued. However, using Fact 3.1a we have $R_S(s') \geq R_S(s'')$, implying that $s'' \in \mathsf{RM}(S)$ since $s' \in \mathsf{RM}(S)$, giving a contradiction to $s'' \notin \mathsf{RM}(S)$.

   In sum, we showed that s is weakly dominated by $s' \in S$, and in addition s' cannot be weakly dominated by any strategy in S, contradicting the fact that $s \in \mathsf{UD}(S)$.

3. $\mathsf{UD}(\mathsf{RM}(S)) \subseteq \mathsf{UD}(S)$.

   Suppose not, that is, there exists some $s \in \mathsf{UD}(\mathsf{RM}(S))$ that is not in $\mathsf{UD}(S)$. By the definition of $\mathsf{UD}(S)$, the strategy s must be weakly dominated by some $s' \in S$, and in addition s' cannot be weakly dominated by any other strategy in S. There are two cases here.

   - The first case is when $s' \in \mathsf{RM}(S)$. This case is impossible because $s \in \mathsf{UD}(\mathsf{RM}(S))$ implies that if s is weakly dominated by $s' \in \mathsf{RM}(S)$, then s' must also be weakly dominated, contradicting the fact that s' cannot be weakly dominated by any strategy in S.
   - The second case is when $s' \notin \mathsf{RM}(S)$. Since $s \prec s'$, by Fact 3.1a we have $R_S(s) \geq R_S(s')$. However, because $s \in \mathsf{UD}(\mathsf{RM}(S))$ implies that $s \in \mathsf{RM}(S)$, it must hold that s' is a regret minimizer with respect to S, contradicting the fact that $s' \notin \mathsf{RM}(S)$.

4. $\mathsf{UD}(\mathsf{RM}(S)) \subseteq \mathsf{RM}(\mathsf{UD}(S))$.

   Given that $\mathsf{UD}(\mathsf{RM}(S)) \subseteq \mathsf{UD}(S)$ (proved above), consider any strategy $s \in \mathsf{UD}(\mathsf{RM}(S))$, and suppose that $s \notin \mathsf{RM}(\mathsf{UD}(S))$. Then there exists some $s' \in \mathsf{UD}(S)$ satisfying $R_{\mathsf{UD}(S)}(s) > R_{\mathsf{UD}(S)}(s')$. This implies, through Fact 3.1b, that $R_S(s) > R_S(s')$, contradicting the fact that $s \in \mathsf{RM}(S)$.

5. $\mathsf{RM}(\mathsf{UD}(S)) \subseteq \mathsf{RM}(S) \cap \mathsf{UD}(S)$.

   Trivial given the previous steps: $\mathsf{RM}(\mathsf{UD}(S)) \subseteq \mathsf{UD}(S)$ and $\mathsf{RM}(\mathsf{UD}(S)) = \mathsf{UD}(\mathsf{RM}(S)) \subseteq \mathsf{RM}(S)$.

6. $\mathsf{RM}(S) \cap \mathsf{UD}(S) \subseteq \mathsf{RM}(\mathsf{UD}(S))$.

   Take any strategy $s \in \mathsf{RM}(S) \cap \mathsf{UD}(S)$, and suppose that $s \notin \mathsf{RM}(\mathsf{UD}(S))$. Then there exists some $s' \in \mathsf{UD}(S)$ satisfying $R_{\mathsf{UD}(S)}(s) > R_{\mathsf{UD}(S)}(s')$. This implies, through Fact 3.1b, that $R_S(s) > R_S(s')$, contradicting the fact that $s \in \mathsf{RM}(S)$. □

It is not hard to see that Lemma 3.2 implies our theorem. That is,

Theorem 3.3. *From any menu $S \subseteq \mathcal{S}$, the set of strategies obtained by applying,* in arbitrary order*, i times the operator* RM *and at least once the operator* UD*, is:*

$$\mathsf{RM}^i(S) \cap \mathsf{UD}(S) \ .$$

3.4 Implications for Mechanism Design

Mechanism design enables a social planner to generate a desirable outcome by leveraging the rationality (and the beliefs) of the players. Most works in mechanism designs assume the players to be utility maximizers. In particular, implementation in undominated strategies traces back to Jackson [79]. However, mechanism design also considers regret minimizers. In particular, Linhart and Radner [98] study regret-minimizing strategies in a sealed-bid mechanism for bilateral bargaining under complete information. Engelbrecht-Wiggans [58] and Selten [140] analyze first- and second-price sealed-bid auctions by incorporating regret for the bidders. Halpern and Pass [71] propose the solution concept of iterated regret minimization using beliefs, and argue that it actually is the only one capable of explaining the actual behavior of the players in some settings.

If a mechanism ensures that each player has a unique undominated strategy, then that strategy is also dominant, and thus the only regret-minimizing one. However, it is not always possible to design such mechanisms. The designer of a new mechanism M may never be sure that M will be played solely by utility-maximizing players, nor that it will be played solely by regret-minimizing players. In principle, if he designs M so that it implements a social choice correspondence f in undominated strategies, then M might produce a non desired outcome when one of the players is a regret minimizer, and viceversa.

We wish to quickly point out that Theorem 3.3 has an immediate but reassuring consequence for mechanism design.

Assume that a mechanism M implements a social choice correspondence f whenever each player chooses a strategy is a strategy subset that coincides either with RM *or with* UD*. Then M is automatically guaranteed to implement f whenever each player chooses a strategy in his set* RM(UD)*.*

For instance, a mechanism implementing f for regret minimizers continues to implement f when the players are utility maximizers who resort to regret only for further refining, if needed, their sets of undominated strategies.

3.5 Pure vs. Mixed Strategies

So far we have been ambiguous, when discussing undominated strategies and regret-minimizing ones, about whether or not the players consider only pure strategies or also mixed ones. When only pure strategies are allowed, a utility maximizer compares only between his pure strategies for the notion of dominance and plays a pure undominated one, while a regret minimizer picks a pure strategy that minimizes regret among his pure strategies.

Our theorem and lemma are stated for pure strategies.

When mixed strategies are allowed, the definitions of UD and RM need more careful attention. It is easy to see that, when considering mixed strategies for regret minimizers, the only change needed is to allow such a minimizer to choose a mixed strategy that minimizes his expected regret among all his mixed ones (see e.g., [76, 71]). Note that, it is easy to construct examples in which a mixed strategy yields strictly smaller regret than any pure strategy.

It is important to realize, however, that if we allow regret minimizers to consider mixed strategies, we *should* also allow utility maximizers to consider mixed strategies. For instance, our structural lemma (Lemma 3.2) would have difficulty to equate a set of pure strategies and a set of mixed ones. A utility maximizer may consider mixed strategies when determining that a strategy s is weakly dominated by another strategy s'. The two interesting cases to consider are (1) s is pure and s' is mixed; and (2) both s and s' are mixed. Traditionally, most attention has been devoted to the first case, but the second has been studied too (see for instance [48, 134]). Clearly, UD can be defined in both cases, and yields a more "refined" set of strategies in the second case.[5] It is actually under this more refined case that our structural lemma holds. In a sense, we have nothing to lose and something to gain by adopting a more flexible definition, after all the right notions are those yielding the right theorems.

[5]Let $\mathsf{UD}^{\mathsf{pure}}$ be the set of (pure) undominated strategies in the first case, and UD be the set of (possibly mixed) undominated strategies in the second case. Then, UD is a more "refined" notion of undominated strategies than $\mathsf{UD}^{\mathsf{pure}}$ because $\mathsf{UD}^{\mathsf{pure}} \subseteq \mathsf{UD} \subseteq \Delta(\mathsf{UD}^{\mathsf{pure}})$, i.e., $\mathsf{UD}^{\mathsf{pure}}$ coincides with the support of UD. For this reason, there is no difference in choosing between the two notions in most of the literature (see [48, footnote 2]).

Part II

Novel Frameworks for Optimization

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent

This chapter is based on the result published in [5], and its further edits can be found at:

http://arxiv.org/abs/1407.1537.

First-order methods play a central role in large-scale convex optimization. Even though many variations exist, each suited to a particular problem form, almost all such methods fundamentally rely on two types of algorithmic steps and two corresponding types of analysis: gradient-descent steps, which yield primal progress, and mirror-descent steps, which yield dual progress. In this paper, we observe that the performances of these two types of step are complementary, so that faster algorithms can be designed by linearly coupling the two steps.

In particular, we obtain a simple accelerated gradient method for the class of smooth convex optimization problems. The first such method was proposed by Nesterov back to 1983 [116, 117, 118], but to the best of our knowledge, the proof of the fast convergence of accelerated gradient methods has not found a clear interpretation and is still regarded by many as crucially relying on "algebraic tricks"[87]. We apply our novel insights to construct a new accelerated gradient method as a natural linear coupling of gradient descent and mirror descent and to write its proof of convergence as a simple combination of the convergence analyses of the two underlying descent steps.

We believe that the complementary view and the linear coupling technique in this paper will prove very useful in the design of first-order methods as it allows us to design fast algorithms in a conceptually easier way. For instance, our technique greatly facilitates the recent breakthroughs in solving packing and covering linear programs [7, 6].

4.1 Introduction

The study of fast iterative methods for approximately solving linear programs and, more generally, convex programming problems is a central focus of research in convex optimization, with important applications in Machine Learning, Combinatorial Optimizations and many other areas of Computer Science and Mathematics. The crowning jewel of this field of research has been the development of interior point methods, iterative methods that produce ε-additive approximations to the optimum with a small number of iterations and a logarithmic $\log\left(\frac{1}{\varepsilon}\right)$ dependence on the accuracy ε.

The fast rate of convergence of interior point methods comes at the cost of more expensive iterations, typically requiring the solution of a system of linear equations in the input variables. As a consequence, the cost of each iteration typically grows at least quadratically with the problem dimension, making interior point methods impractical for very-large-scale convex programs where the problem dimension is on the magnitude of millions or billions [27]. In such a regime, the methods of choice are first-order algorithms. These are modeled as accessing the target convex-optimization problem $\min_{x \in Q} f(x)$ in a black-box fashion: the algorithm queries a point $y \in Q$ at every iteration and receives the pair $\big(f(y), \nabla f(y)\big)$.[1] The convergence of the algorithm is measured in the number of queries necessary to produce a feasible solution which achieves an additive ε-approximation to the optimum.

Because of the restricted interaction with the input, first-order methods only require very cheap and often highly parallelizable iterations, which makes them well-suited to massive optimization problems. At the same time, first-order methods often require a number of iterations inversely polynomial to the accuracy ε, i.e. exponentially larger than required by interior-point algorithms.

Recently, first-order methods have experienced a renaissance in the design of fast algorithms for fundamental combinatorial problems. In particular, gradient-descent techniques play a crucial role in recent breakthroughs on the complexity of approximate maximum flow problems [94, 150, 88, 104]. At the same time, multiplicative weight updates, another first-order method and a cornerstone technique in online learning, have become a standard tool in the design of fast algorithms and have been applied with success to a variety of problems, including approximately solving linear and semidefinite relaxations of fundamental combinatorial problems [131, 65, 9, 10] as well as spectral algorithms for graph problems [46, 126].

Despite the myriad of applications, first-order methods with provable convergence guarantees can be mostly classified as instantiations of two fundamental algorithmic

[1] Here, variable x is constrained to lie in a convex set $Q \subseteq \mathbb{R}^n$, which is known as the *constraint set* of the problem.

ideas: *gradient descent* and the *mirror descent*.[2]

A method with provable guarantees must provide both a solution x_{out} and an implicit or explicit certificate that x_{out} in the form of a lower bound on the optimum. We refer to the task of constructing a solution x_{out} of small objective as the primal side of the problem and to that of constructing a lower bound on the optimum as the dual side.

We will argue that gradient descent takes a fundamentally primal approach, while mirror descent follows a complementary dual approach. In our main result, we will show how these two approaches blend in a natural manner to yield a new and simple accelerated gradient method for smooth convex optimization problems.

4.1.1 Understanding First-Order Methods: Gradient Descent and Mirror Descent

In this section, we provide high-level descriptions of the gradient-descent and the mirror-descent algorithms and their analysis. While much of this material is classical in the field of optimization, our intuitive presentation of these ideas forms the basis for our main result. For a more detailed survey of gradient descent and mirror descent, we recommend the textbooks [117, 27].

For the purpose of this section, we only consider the case of unconstrained minimization (i.e. $Q = \mathbb{R}^n$), but, as we will see in Section 4.2, the same intuition and a similar analysis extend to the constrained case. In the following, we will also be using generic dual norms $\|\cdot\|$ and $\|\cdot\|_*$. At a first reading, they can be both replaced with the Euclidean norm $\|\cdot\|_2$.

Primal Approach: Gradient Descent for Smooth Convex Optimization

A natural approach to iterative optimization is to decrease the objective function as much as possible at every iteration. To formalize the effectiveness of this idea, one has to introduce an additional smoothness assumption on the objective function $f(x)$; specifically, this is achieved by considering the class of objectives that are L-smooth (i.e., that have L-Lipschitz continuous gradient):

$$\forall x, y, \quad \|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\| \ .$$

The smoothness condition immediately yields a global quadratic upper bound on the function around a query point x:

$$\forall y, \quad f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \ . \tag{4.1}$$

[2]We emphasize here that these two terms are sometimes used ambiguously in the literature; in this paper, we attempt to stick as close as possible to the conventions of the Optimization community and in particular in the textbooks [117, 27] with one exception: we extend the definition of gradient descent to non-Euclidean norms in a natural way, following [88].

The gradient-descent algorithm exploits this bound by taking a step that maximizes the guaranteed objective decrease (i.e., the primal progress) $f(x_k) - f(x_{k+1})$ at every iteration k. More precisely,

$$x_{k+1} \leftarrow \arg\min_y \left\{ \frac{L}{2}\|y - x_k\|^2 + \langle \nabla f(x), y - x_k \rangle \right\} .$$

Notice that here $\|\cdot\|$ is a generic norm. When this is the Euclidean ℓ_2-norm, the step takes the familiar additive form $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$. However, in other cases, e.g., for the non-Euclidean ℓ_1 or ℓ_∞ norms, the update step will not follow the direction of the gradient $\nabla f(x_k)$ (see for instance [118, 88]).

Under the smoothness assumption above, the magnitude of this primal progress is at least

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L}\|\nabla f(x_k)\|_*^2 . \tag{4.2}$$

In general, this quantity will be larger when the gradient $\nabla f(x_k)$ has large norm.

Inequality (4.2) ensures that at every iteration the objective value of the current solution x_k decreases by at least $\frac{1}{2L}\|\nabla f(x_k)\|_*^2$. The proof of convergence of gradient descent is completed by using a basic convexity argument to relate $f(x_k) - f(x^*)$ and $\|\nabla f(x_k)\|_*$ (where x^* is the minimizer of $f(x)$). The final bound shows that the algorithm converges to an ε-approximate solution in $O\left(\frac{L}{\varepsilon}\right)$ iterations [117]. More details on the gradient-descent algorithm and its analysis are given in Section 4.2.1 and in Nesterov's book [117].

In conclusion, it is useful to think of gradient descent as choosing query points in a greedy way to ensure the largest possible primal progress at every iteration. The limitation of this strategy is that it does not make any attempt to construct a good lower bound to the optimum value, i.e., it essentially ignores the dual problem. In the next subsection, we will see a method that takes the opposite approach by focusing completely on the dual side. This method is suitable when there is no guarantee on the smoothness of the objective.

Dual Approach: Mirror Descent for Nonsmooth Convex Optimization

In non-smooth convex optimization, we are given an upper bound ρ on the Lipschitz constant of $f(x)$, rather than $\nabla f(x)$. When f is differentiable, this means that the gradient could change arbitrarily fast, but its norm remains bounded, i.e., $\|\nabla f(x)\| \leq \rho$ for every $x \in Q$. The possibility that the gradient varies quickly seriously undermines the performance of gradient descent, which relies on making a certain amount of primal progress at every iteration. In this case, it is not possible to guarantee that an update step of a predetermined length would result in an improved objective value, as the gradient may be very large even at points very near the optimum. At the same time, we cannot afford to take too small steps as this limits our rate of convergence.

Dual-averaging methods (see for instance [114, 119, 56, 163, 27]) bypass this obstacle by tackling the dual problem of constructing a lower bound to the optimum. They

interpret each queried gradient as a hyperplane lower bounding the objective function $f(x)$ and attempt to carefully construct a convex combination of these hyperplanes that yields a stronger lower bound. Intuitively, the flatter the queried gradients are (i.e. the smaller $\|\nabla f(x_k)\|_* \leq \rho$ is), the fewer iterations will be needed to combine them into an approximately optimal solution.

Formally, at each iteration k, using the convexity of $f(x)$, we can consider the following lower bound implied by the gradient $\nabla f(x_k)$:

$$\forall u, \quad f(u) \geq f(x_k) + \langle \nabla f(x_k), u - x_k \rangle \ .$$

To get a stronger lower bound, we can form a linear combination of the lower bounds given by all the queried gradients, and obtain[3]

$$\forall u, \quad f(u) \geq \tfrac{1}{T} \textstyle\sum_{t=0}^{T-1} f(x_k) + \tfrac{1}{T} \sum_{t=0}^{T-1} \langle \nabla f(x_k), u - x_k \rangle \ . \tag{4.3}$$

On the upper bound side, we consider the point $\bar{x} = \frac{1}{T} \sum_{k=0}^{T-1} x_k$, i.e., the mean of the queried points. By straightforward convexity argument, we have $f(\bar{x}) \leq \frac{1}{T} \sum_{k=0}^{T-1} f(x_k)$. As a result, we can upper bound the distance between $f(\bar{x})$ and $f(u)$ for any arbitrary u using (4.3):

$$\forall u, \quad f(\bar{x}) - f(u) \leq \tfrac{1}{T} \textstyle\sum_{k=0}^{T-1} \langle \nabla f(x_k), x_k - u \rangle \stackrel{\text{def}}{=} R_T(u) \ . \tag{4.4}$$

Borrowing terminology from online learning, the righthand side $R_T(u)$ is known as the *regret* of the sequence $(x_k)_{k=0}^{T-1}$ with respect to point u.

Dual Averaging via Regularization: Mirror Descent. We are aware of two main algorithmic instantiations of dual averaging: *Nemirovski's mirror descent* [114] and *Nesterov's dual averaging* [119].[4] Both these algorithm make use of a *regularizer* $w(\cdot)$, also known as the distance-generating function (DGF), which is a strongly convex function over Q with respect to some norm $\|\cdot\|$. The two methods are very similar, differing only in how the constraint set is integrated in the update step [106]. In fact, they are exactly identical in the unconstrained case $Q = \mathbb{R}^n$ and, more generally, when $w(\cdot)$ enjoys some nice properties (see Appendix 4.A.3). Below, we focus on the unconstrained case.

Both algorithms consider a regularized version $\tilde{R}_k$ of the regret in (4.4):

$$\tilde{R}_k(u) \stackrel{\text{def}}{=} \frac{1}{\alpha k} \cdot \Big(-w(u) + \alpha \sum_{i=0}^{k-1} \langle \nabla f(x_i), x_i - u \rangle \Big) \ ,$$

where $\alpha > 0$ is a trade-off parameter. Notice that an upper bound on $\tilde{R}_k(u)$ can be simply converted into one for $R_k(u)$ with an additive loss: $R_k(u) \leq \tilde{R}_k(u) + \frac{w(u)}{\alpha k}$. Both Nemirovski's mirror descent and Nesterov's dual averaging attempt to

[3]For simplicity, we choose uniform weights here. For the purpose of proving convergence results, the weights of individual hyperplanes are typically uniform or only dependent on k.

[4]Several other update rules can be viewed as specializations or generalizations of the mentioned instantiations. For instance, the follow-the-regularized-leader (FTRL) step is a generalization of Nesterov's dual averaging step where the regularizers are allowed to be adaptively and incrementally selected (see [107]).

minimize the maximum regularized regret at the next iteration (i.e., $\max_u \tilde{R}_{k+1}(u)$), by choosing the next query point x_k to be the maximizer of the current regularized regret (i.e., $\arg\max_u \tilde{R}_k(u)$). It turns out that this choice of query point successfully drives $\max_u \tilde{R}_{k+1}(u)$ down. In fact, the smaller the queried gradient $\nabla f(x_k)$ is, the smaller the new maximum regularized regret $\max_u \tilde{R}_{k+1}(u)$ will be. In general, one can show that:

$$\max_u \tilde{R}_{k+1}(u) \leq \frac{k}{k+1} \max_u \tilde{R}_k(u) + O\Big(\frac{\alpha}{k+1} \|\nabla f(x_k)\|_*^2\Big). \tag{4.5}$$

This bound can then be turned into a convergence proof requiring $T = O(\rho^2/\varepsilon^2)$ iterations.

We remark that the convergence argument sketched here crucially relies on the use of the regularized regret (instead of the original regret). In particular, Inequality (4.5) directly follows from a smoothness property of the maximum regularized regret with respect to the addition of new gradient hyperplanes, which only holds when the regularizer $w(u)$ is strongly convex. For more details of this view of dual averaging and the proof of (4.5), see Appendix 4.A.4.

This paper. In this paper, we adopt *mirror descent* as our dual algorithm of choice, as it is more familiar to the Theoretical Computer Science audience. Indeed, the most common instantiation of mirror descent is perhaps the multiplicative-weight-update algorithm, which has become a standard tool in the design of algorithms [10] (see Appendix 4.A.2 for this relationship). We describe the mirror descent step for the constrained case and its analysis in Section 4.2.2. A great resource for an in-depth description of mirror descent is the textbook by Ben-Tal and Nemirovski [27].

Remark: A Few Exceptions

One may occasionally find analyses that do not immediately fall into the above two categories. To name a few, Dekel *et al.* [52] have applied dual averaging steps to a *smooth* objective, and shown that the convergence rate is the same as that of gradient descent. Shamir and Zhang [148] have studied non-smooth objectives and obtained an algorithm that converges slightly slower than dual averaging, but has an error guarantee on the last iterate, rather than the average history.

4.1.2 Our Conceptual Question

Following this high level description of gradient and mirror descent, it is useful to pause and observe the complementary nature of the two procedures. Gradient descent relies on primal progress, uses local steps and makes faster progress when the norms of the queried gradients $\nabla f(x_k)$ are large. In contrast, mirror descent works by ensuring dual progress, uses global steps and converges faster when the norms of the queried gradients are small.

This interpretation immediately leads to the question that inspires our work:

Can Gradient Descent and Mirror Descent be combined to obtain faster first-order algorithms?

In this paper, we initiate the formal study of this key conceptual question. We believe that the techniques and insights to answer this question have the potential to lead to faster and better motivated algorithms for many more computational problems.

4.1.3 Accelerated Gradient Method From Linear Coupling

In the seminal work [116, 117], Nesterov has designed an accelerated gradient method for the class of L-smooth functions with respect to ℓ_2 norms, and this method performs quadratically faster than gradient descent —requiring $\Omega(L/\varepsilon)^{0.5}$ rather than $\Omega(L/\varepsilon)$ iterations. This is also shown to be asymptotically tight [117]. Later in 2005, Nesterov himself generalizes this method to allow non-Euclidean norms in the definition of smoothness [118]. All these versions of methods are referred to as *accelerated gradient methods*, or sometimes as Nesterov's accelerated methods.

Although accelerated gradient methods have been widely applied (to mention a few, see [146, 147] for regularized optimizations, [121, 93] for composite optimization, [120] for cubic regularization, [122] for universal method, and [94] for an application on maxflow), little geometric explanation is known. For instance, Juditsky [87] has mentioned that Nesterov's method "looks as an analytical trick."

In this paper, we provide a simple, alternative, but **complete** version of the accelerated gradient method. Here, by 'complete' we mean our method works for any norm, and for both the constrained and unconstrained case. This is in contrast with the (perhaps better-known) version of Nesterov [117] that only works with the ℓ_2 Euclidean norm.[5]

Instead of using the *estimation sequence* technique provided in the original proof of Nesterov, we take a different path. Our key observation is to construct two sequences of updates: one sequence of gradient steps and one sequence of mirror steps. Recall that, according to the gradient-descent and mirror-descent analyses described above, the gradient steps perform well whenever the observed gradients are large; the mirror steps perform well whenever the observed gradients are small. Thus, intuitively, we hope to *couple* these two steps together, and choose the better method 'adaptively' according to the size of the gradient. We begin with a thought experiment.

Thought Experiment. Consider the case when the smooth property is with respect to the ℓ_2-norm, and the objective $f(x)$ is unconstrained. Suppose that $\|\nabla f(x)\|_2$, the

[5]Some authors have regarded the result in [117] as the 'momentum analysis' or 'momentum method' [123, 156]. To the best of our knowledge, all the momentum analysis only applies to Euclidean spaces. We point out the importance of allowing non-Euclidean norms in Appendix 4.A.1. (Our proof also extends to the proximal version of first-order methods, but for simplicity, we choose to include only the constrained version.)

size of the observed gradient, is *either* always $\geq K$, or always $\leq K$, where the cut-off value K is determined later. If $\|\nabla f(x)\|_2$ is always $\geq K$, we perform T gradient steps; otherwise we perform T mirror steps. Suppose in addition that we start with some $f(x_0)$ whose distance to $f(x^*)$ is at most 2ε, and we want to obtain some x so that $f(x) - f(x^*) \leq \varepsilon$.[6]

If T gradient steps are conducted, in each step the objective decreases by at least $\frac{\|\nabla f(\cdot)\|_2^2}{2L} \geq \frac{K^2}{2L}$ according to (4.2), and thus we only need to choose $T \geq \Omega(\frac{\varepsilon L}{K^2})$ steps in order to achieve an ε accuracy. On the other hand, if T mirror steps are conducted, we need $T \geq \Omega(\frac{K^2}{\varepsilon^2})$ steps according to the mirror-descent convergence. In sum, in this thought experiment, we need $T \geq \Omega\big(\max\big\{\frac{\varepsilon L}{K^2}, \frac{K^2}{\varepsilon^2}\big\}\big)$ steps to achieve a solution ε-close to the optimum.

Now, setting K to be the 'magic number' so that the two terms in the max function equal, we obtain $T \geq \Omega\big(\frac{L}{\varepsilon}\big)^{1/2}$. This is a quadratic improvement over $T \geq \Omega(\frac{L}{\varepsilon})$ from the gradient descent.

Towards the Actual Proof. To turn this thought experiment into an actual proof, we are facing the following obstacles. The gradient steps always decrease the objective, while the mirror step may very often increase the objective, cancelling the effect of the gradient steps. On the other hand, the mirror steps are *only* useful when a large number of iterations are performed in a row, and the performance guarantee is on the average of these iterations; if any primal step stands in the middle, this guarantee is destroyed.

Therefore, it is natural to design an algorithm that, in every single iteration k, performs *both* a gradient and a mirror step, and somehow ensure that the two steps are coupled together. However, the following additional difficulty arises: if from some starting point x_k, the gradient step instructs us to go to y_k, while the mirror step instructs us to go to z_k, then how do we continue? Do we look at the gradient at $\nabla f(y_k)$ or $\nabla f(z_k)$? In particular, if $\|\nabla f(y_k)\|_2$ is large, we can continue performing gradient steps from y_k; or if $\|\nabla f(z_k)\|_2$ is small, we can continue performing mirror steps from z_k. However, what if $\|\nabla f(y_k)\|_2$ is small but $\|\nabla f(z_k)\|_2$ is large?

This problem is implicitly solved by Nesterov using the following simple idea[7]: in the k-th step, we can choose a linear combination $x_{k+1} \leftarrow \tau z_k + (1-\tau) y_k$, and use this same gradient $\nabla f(x_{k+1})$ to continue the gradient and mirror steps. Whenever τ is carefully chosen (just like the 'magic number' K being selected), the two descent sequences provide a coupled bound on the error guarantee, and we recover the method of [118].

Finally, we point out that our method also recovers the strong convexity version

[6] It is worth noting that for first-order methods, the heaviest computation always happens in this 2ε to ε procedure.

[7] We wish to point out that Nesterov has phrased his method differently from ours, and little is known on why this linear combination is needed from his proof, except for being used as an algebraic trick to cancel specific terms.

of [117], and therefore is a full proof to all existing versions of accelerated gradient methods for smooth convex optimization problems.

4.1.4 Conclusion

We provide a simple variant of the accelerated gradient method with a reinterpretation of its convergence analysis. Providing such an intuitive, yet formal interpretation has been a long-open question in Optimization [87]. We believe that our interpretation is one important step towards this general goal, and may facilitate the study of accelerated gradient methods in a white-box manner, so as to apply them to problems outside its original scope.

In addition, we believe that our complementary view of gradient descent and mirror descent is a very fundamental (and to the best of our knowledge, new!) conceptual message in the design of first-order methods. This has the potential to lead to faster and better motivated algorithms for many more computational problems. Indeed, we have already succeeded in this direction in our separate papers [7, 6], where we have proposed faster nearly-linear-time algorithms for approximately solving positive linear programs, both in parallel and in sequential.[8]

4.2 Preliminaries

4.2.1 Review of Primal Descent

Consider a function $f(x)$ that is convex and differentiable on a closed convex set $Q \subseteq \mathbb{R}^n$,[9] and assume that f is *L-smooth* (or has L-Lipschitz continuous gradient) with respect to $\|\cdot\|$, that is

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|, \quad \forall x, y \in Q$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.[10]

Definition 4.1. *For any $x \in Q$, the* gradient (descent) step *(with step length $\frac{1}{L}$) is*

$$\tilde{x} = \mathsf{Grad}(x) \stackrel{\text{def}}{=} \arg\min_{y \in Q} \left\{ \tfrac{L}{2}\|y - x\|^2 + \langle \nabla f(x), y - x \rangle \right\}$$

[8] In our paper [7] (see Chapter 5), we have designed an iterative algorithm whose update steps can be viewed both as gradient and as mirror steps, therefore allowing us to apply two complementary analyses to support each other; this breaks the $O(1/\varepsilon^4)$ barrier in the parallel packing/covering LP solver running time since [101].

In our paper [6] (see Chapter 6), we have designed algorithms whose update steps can be viewed as linear couplings of (the coordinates version of) gradient and mirror steps; this breaks the $O(1/\varepsilon^2)$ barrier in the sequential packing/covering LP solver running time since [24, 165, 25].

Neither of the two papers is any direct variant of accelerated gradient methods, and their objectives are not even smooth.

[9] In most of the applications, Q is simple enough so that the gradient steps (and mirror steps as well) can be computed explicitly and efficiently. For instance, one may use the positive orthant, $Q = \{x \in \mathbb{R}^n : x \geq 0\}$, the unit sphere, $Q = \{x \in \mathbb{R}^n : \|x\|_2 = 1\}$, and many others.

[10] $\|\xi\|_* \stackrel{\text{def}}{=} \max\{\langle \xi, x \rangle : \|x\| \leq 1\}$. For instance, ℓ_p norm is dual to ℓ_q norm if $\frac{1}{p} + \frac{1}{q} = 1$.

and we let $\mathsf{Prog}(x) \stackrel{\text{def}}{=} -\min_{y\in Q}\{\frac{L}{2}\|y-x\|^2 + \langle \nabla f(x), y-x\rangle\} \geq 0$.

In particular, when $\|\cdot\| = \|\cdot\|_2$ is the ℓ_2-norm and $Q = \mathbb{R}^n$ is unconstrained, the gradient step can be simplified as $\mathsf{Grad}(x) = x - \frac{1}{L}\nabla f(x)$. Or, slightly more generally, when $\|\cdot\| = \|\cdot\|_2$ is the ℓ_2-norm but Q may be constrained, we have $\mathsf{Grad}(x) = x - \frac{1}{L}g_Q(x)$ where $g_Q(x)$ is the gradient mapping of f at x (see [117, Chapter 2.2.3]).

The classical theory on smooth convex programming gives rise to the following lower bound on the amount of objective decrease (whose proof is provided in Appendix 4.B for completeness).

Gradient Descent Guarantee

$$f(\mathsf{Grad}(x)) \leq f(x) - \mathsf{Prog}(x) \qquad (4.6)$$

or in the special case when $Q = \mathbb{R}^n$ $\qquad f(\mathsf{Grad}(x)) \leq f(x) - \frac{1}{2L}\|\nabla f(x)\|_*^2$.

From the above descent guarantee, one can deduce the convergence rate of the gradient descent steps. In particular, if $\|\cdot\| = \|\cdot\|_2$ is the Euclidean norm, and the gradient step $x_{k+1} = \mathsf{Grad}(x_k)$ is applied T times, we obtain the following convergence guarantee (see [117, Chapter 2.1.5])

$$f(x_T) - f(x^*) \leq O\Big(\frac{L\|x_0 - x^*\|_2^2}{T}\Big) \quad \text{or equivalently}$$
$$T \geq \Omega\Big(\frac{L\|x_0 - x^*\|_2^2}{\varepsilon}\Big) \Rightarrow f(x_T) - f(x^*) \leq \varepsilon \ .$$

Here, x^* is any minimizer of $f(x)$. If $\|\cdot\|$ is a general norm, but $Q = \mathbb{R}^n$ is unconstrained, the above convergent rate becomes $f(x_T) - f(x^*) \leq O\big(\frac{LR^2}{T}\big)$, where $R = \max_{x:f(x)\leq f(x_0)}\|x - x^*\|$. We provide the proof of this later case in Appendix 4.B because it is less known and we cannot find it in the optimization literature.

Note that, we are unaware of any universal convergence proof for both the general norm and the unconstrained case. As we shall see later in Section 4.4, this convergence rate can be improved by accelerated gradient methods, even for the general norm $\|\cdot\|$ and the constrained case.

4.2.2 Review of Mirror Descent

Consider some function $f(x)$ that is convex on a closed convex set $Q \subseteq \mathbb{R}^n$, and assume that f is ρ-*Lipschitz continuous* with respect to norm $\|\cdot\|$, that is

$$|f(x) - f(y)| \leq \rho\|x - y\|, \quad \forall x, y \in Q \ .$$

Notice that this is equivalent to saying that f admits a subgradient $\partial f(x)$ at every point $x \in Q$, and satisfies $\|\partial f(x)\|_* \leq \rho$ for all x. (Recall that $\partial f(x) = \nabla f(x)$ if f is differentiable.)

The mirror descent method requires one to choose a distance generating function.

Definition 4.2. *We say that* $w(x)\colon \mathbb{R}^n \to \mathbb{R}$ *is a* distance generating function (DGF)*, if* w *is* 1*-strongly convex with respect to* $\|\cdot\|$*, or in symbols*

$$w(y) \geq w(x) + \langle \nabla w(x), y - x\rangle + \frac{1}{2}\|x-y\|^2 \quad \forall x \in Q \setminus \partial Q,\, \forall y \in Q \ .^{11}$$

Accordingly, the Bregman divergence *(or* prox-term*) is given as*

$$V_x(y) \stackrel{\text{def}}{=} w(y) - \langle \nabla w(x), y - x\rangle - w(x) \quad \forall x \in Q \setminus \partial Q,\, \forall y \in Q \ .$$

The property of DGF ensures that $V_x(x) = 0$ *and* $V_x(y) \geq \frac{1}{2}\|x-y\|^2 \geq 0$*.*

Common examples of DGFs include `(i)` $w(y) = \frac{1}{2}\|y\|_2^2$, which is strongly convex with respect to the ℓ_2-norm over any convex set Q, and the corresponding $V_x(y) = \frac{1}{2}\|x-y\|_2^2$, and `(ii)` the entropy function $w(y) = \sum_i y_i \log y_i$, which is strongly convex with respect to the ℓ_1-norm over any $Q \subseteq \Delta \stackrel{\text{def}}{=} \{x \geq 0 \,:\, \mathbb{1}^T x = 1\}$, and the corresponding $V_x(y) = \sum_i y_i \log(y_i/x_i) \geq \frac{1}{2}\|x-y\|_1^2$.

Definition 4.3. *The* mirror (descent) step *with step length* α *can be described as*

$$\tilde{x} = \mathsf{Mirr}_x(\alpha \cdot \partial f(x)) \quad \textit{where} \quad \mathsf{Mirr}_x(\xi) \stackrel{\text{def}}{=} \arg\min_{y \in Q} \big\{V_x(y) + \langle \xi, y - x\rangle\big\}$$

The core lemma of mirror descent is the following inequality. (Its proof can be found in Appendix 4.B for completeness.)

Mirror Descent Guarantee

If $x_{k+1} = \mathsf{Mirr}_{x_k}\big(\alpha \cdot \partial f(x_k)\big)$, then

$$\forall u \in Q, \quad \alpha(f(x_k) - f(u)) \leq \alpha\langle \partial f(x_k), x_k - u\rangle \leq \frac{\alpha^2}{2}\|\partial f(x_k)\|_*^2 + V_{x_k}(u) - V_{x_{k+1}}(u) \ . \tag{4.7}$$

The term $\langle \partial f(x_k), x_k - u\rangle$ features prominently in online optimization (see for instance the survey [143]), where it is known as the *regret* at iteration k with respect to u.[12] It is not hard to see that, after telescoping (4.7) for $k = 0, \ldots, T-1$, letting $\bar{x} \stackrel{\text{def}}{=} \frac{1}{T}\sum_{k=0}^{T-1} x_k$ be the average of the x_k's, and letting x^* be the minimizer of $f(x)$, we have

$$\alpha T(f(\bar{x}) - f(x^*)) \leq \sum_{k=0}^{T-1} \alpha\langle \partial f(x_k), x_k - x^*\rangle \leq \frac{\alpha^2}{2}\sum_{k=0}^{T-1}\|\partial f(x_k)\|_*^2 + V_{x_0}(x^*) - V_{x_T}(x^*) \ . \tag{4.8}$$

Finally, letting Θ be any upper bound on $V_{x_0}(x^*)$, and $\alpha = \frac{\sqrt{2\Theta}}{\rho\cdot\sqrt{T}}$ be the step length, inequality (4.7) ensures that

$$f(\bar{x}) - f(x^*) \leq \frac{\sqrt{2\Theta}\cdot\rho}{\sqrt{T}} \quad \text{or equivalently} \quad T \geq \frac{2\Theta\cdot\rho^2}{\varepsilon^2} \Rightarrow f(\bar{x}) - f(x^*) \leq \varepsilon \ . \tag{4.9}$$

Notice that $\Theta = \frac{1}{2}\|x_0 - x^*\|_2^2$ when $\|\cdot\|$ is the Euclidean norm.

[11] One can in fact only require w to have subgradients at all $x \in Q \setminus \partial Q$.

[12] The notion of *regret* is especially used in the language of multiplicative weight update methods, which can be viewed as mirror descent, see Appendix 4.A.2.

4.2.3 Remark

While their analyses share some similarities, mirror and gradient steps are often very different. This is particularly true when working with non-Euclidean norms. For example, if we consider an optimization problem over the simplex with underlying norm ℓ_1-norm, the gradient step gives $x' \leftarrow \arg\min_y\{\frac{1}{2}\|y-x\|_1^2 + \alpha\langle\nabla f(x), y-x\rangle\}$, while the mirror step with entropy regularizer gives $x' \leftarrow \arg\min_y\{\sum_i y_i \log(y_i/x_i) + \alpha\langle\nabla f(x), y-x\rangle\}$. We shall point out in Appendix 4.A.1 that non-Euclidean norms are very important for certain applications.

In the special case of $w(x) = \frac{1}{2}\|x\|_2^2$ and $\|\cdot\| = \|\cdot\|_2$, gradient and mirror steps are indistinguishable from each other. However, as we have discussed earlier, these two update rules are often equipped with very different convergence analyses, even if they 'look the same'.

4.3 Warm-Up Accelerated Gradient Method with Fixed Step Length

We adopt the same setting as in Section 4.2.1: that is, $f(x)$ is convex and differentiable on its domain Q, and is L-smooth with respect to some norm $\|\cdot\|$. (Note that $f(x)$ may not have a good Lipschitz continuity parameter ρ, but we do not need such a property.)

In this section, we focus on the unconstrained case of $Q = \mathbb{R}^n$, and wish to combine gradient descent and mirror descent to produce a very simple accelerated method, which matches the running time of Nesterov's. We choose to explain this method first because it avoids the mysterious choice of the step lengths in the full accelerated gradient methods, and carries our conceptual message in a very clean way.

As argued in Section 4.1.3, it is desirable to design an algorithm that, in every single step k, performs *both* a gradient and a mirror step, and ensures that the two steps are linearly coupled. In particular, we consider the following steps: starting from $x_0 = y_0 = z_0$, in each step $k = 0, 1, \ldots, T-1$, we first compute $x_{k+1} \leftarrow \tau z_k + (1-\tau) y_k$ and then

- perform a gradient step $y_{k+1} \leftarrow \mathsf{Grad}(x_{k+1})$, and
- perform a mirror step $z_{k+1} \leftarrow \mathsf{Mirr}_{z_k}\big(\alpha\nabla f(x_{k+1})\big)$.[13]

Above, α is the (fixed) step length of the mirror step, while τ is the parameter controlling our coupling. The choices of α and τ will become clear at the end of this section, but from a high level,

- α will be determined from the mirror-descent analysis, similar to that in (4.8), and

[13]Here, the mirror step Mirr is defined by specifying any DGF $w(\cdot)$ that is 1-strongly convex over Q.

- τ will be determined as the best parameter to balance the gradient and mirror steps, similar to the 'magic number' K in our thought experiment discussed in Section 4.1.3.

The classical gradient-descent and mirror-descent analyses immediately imply the following

Lemma 4.4. *For every $u \in Q = \mathbb{R}^n$,*

$$\begin{aligned}\alpha\langle \nabla f(x_{k+1}), z_k - u\rangle &\overset{①}{\le} \frac{\alpha^2}{2}\|\nabla f(x_{k+1})\|_*^2 + V_{z_k}(u) - V_{z_{k+1}}(u) \\ &\overset{②}{\le} \alpha^2 L\big(f(x_{k+1}) - f(y_{k+1})\big) + V_{z_k}(u) - V_{z_{k+1}}(u)\ . \end{aligned} \tag{4.10}$$

Proof. To deduce ①, we note that our mirror step $z_{k+1} = \mathsf{Mirr}_{z_k}(\alpha\nabla f(x_{k+1}))$ is essentially identical to that of $x_{k+1} = \mathsf{Mirr}_{x_k}(\alpha\nabla f(x_k))$ in (4.7), with only changes of variable names. Therefore, inequality ① is a simple copy-and-paste from (4.7) after changing the variable names (see the proof of (4.7) for details). The second inequality ② is from the gradient step guarantee $f(x_{k+1}) - f(y_{k+1}) \ge \frac{1}{2L}\|\nabla f(x_{k+1})\|_*^2$ in (4.6). □

One can already see from the above Lemma 4.4 that, although the mirror step introduces an error of $\frac{\alpha^2}{2}\|\nabla f(x_{k+1})\|_*^2$, this error is proportional to the amount of the gradient step progress $f(x_{k+1}) - f(y_{k+1})$. To be clear, this captures the observation we have stated in the introduction: if $\|\nabla f(x_{k+1})\|_*$ is large, we can make a large gradient step, or if $\|\nabla f(x_{k+1})\|_*$ is small, the mirror step suffers from a small loss.

At this moment, if we choose $\tau = 1$ or equivalently $x_{k+1} = z_k$, the left hand side of inequality (4.10) gives us $\langle \nabla f(x_{k+1}), x_{k+1} - u\rangle$, the regret at iteration x_{k+1}. We therefore wish to telescope it for all choices of k in the spirit as mirror descent (see (4.8)); however, we face the problem that the terms $f(x_{k+1}) - f(y_{k+1})$ do not telescope.[14] On the other hand, if we choose $\tau = 0$ or equivalently $x_{k+1} = y_k$, then the terms $f(x_{k+1}) - f(y_{k+1}) = f(y_k) - f(y_{k+1})$ telescope, but the left hand side of (4.10) is no longer the regret.[15]

To overcome this issue, we need the linear coupling. We compute and upper bound the difference between the left hand side of (4.10) and the real 'regret':

$$\begin{aligned}&\alpha\langle \nabla f(x_{k+1}), x_{k+1} - u\rangle - \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \\ &= \alpha\langle \nabla f(x_{k+1}), x_{k+1} - z_k\rangle = \frac{(1-\tau)\alpha}{\tau}\langle \nabla f(x_{k+1}), y_k - x_{k+1}\rangle \le \frac{(1-\tau)\alpha}{\tau}(f(y_k) - f(x_{k+1})).\end{aligned} \tag{4.11}$$

[14] In other words, although a gradient step may decrease the objective from $f(x_{k+1})$ to $f(y_{k+1})$, it may also get the objective increased from $f(y_k)$ to $f(x_{k+1})$.

[15] Indeed, our "thought experiment" in the introduction is conducted *as if* we both had $x_{k+1} = z_k$ and $x_{k+1} = y_k$, and therefore we could arrive at the desired (4.12) directly.

Above, we have used the choice of x_{k+1} that satisfies $\tau(x_{k+1}-z_k)=(1-\tau)(y_k-x_{k+1})$, as well as the convexity of $f(\cdot)$.

It is now clear that by choosing $\frac{1-\tau}{\tau}=\alpha L$ and combining (4.10) and (4.11), we immediately have

Lemma 4.5 (Coupling)**.** *Letting $\tau\in(0,1)$ satisfy that $\frac{1-\tau}{\tau}=\alpha L$, we have that*

$$\forall u\in Q=\mathbb{R}^n,\quad \alpha\langle\nabla f(x_{k+1}),x_{k+1}-u\rangle\le\alpha^2L\big(f(y_k)-f(y_{k+1})\big)+\big(V_{z_k}(u)-V_{z_{k+1}}(u)\big)\ .$$

It is clear from the above proof that τ is introduced to precisely balance the objective decrease $f(x_{k+1})-f(y_{k+1})$, and the (possible) objective increase $f(y_k)-f(x_{k+1})$. This is similar to the 'magic number' K discussed in the introduction.

Convergence Rate. Finally, we only need to telescope the inequality in Lemma 4.5 for $k=0,1,\dots,T-1$. Letting $\bar{x}\overset{\text{def}}{=}\frac{1}{T}\sum_{k=0}^{T-1}x_k$ and $u=x^*$, we have

$$\alpha T(f(\bar{x})-f(x^*))\le\sum_{k=0}^{T-1}\alpha\langle\partial f(x_k),x_k-x^*\rangle\le\alpha^2L\big(f(y_0)-f(y_T)\big)+V_{x_0}(x^*)-V_{x_T}(x^*)\ . \tag{4.12}$$

Suppose that our initial point y_0 is of error at most d (i.e., $f(y_0)-f(x^*)\le d$), and $V_{x_0}(x^*)\le\Theta$, then (4.12) gives that

$$f(\bar{x})-f(x^*)\le\frac{1}{T}\big(\alpha Ld+\Theta/\alpha\big)\ .$$

Choosing $\alpha=\sqrt{\Theta/Ld}$ to be the value that balances the above two terms,[16] we obtain that $f(\bar{x})-f(x^*)\le\frac{2\sqrt{L\Theta d}}{T}$. In other words,

in $T=4\sqrt{L\Theta/d}$ steps, we can obtain some $\bar{x}$ satisfying $f(\bar{x})-f(x^*)\le d/2$,

halving the distance to the optimum. If we restart this entire procedure a few number of times, halving the distance for every run, then we obtain an ε-approximate solution in

$$T=O\big(\sqrt{L\Theta/\varepsilon}+\sqrt{L\Theta/2\varepsilon}+\sqrt{L\Theta/4\varepsilon}+\cdots\big)=O\big(\sqrt{L\Theta/\varepsilon}\big)$$

iterations, matching the same guarantee of Nesterov's accelerated methods [116, 117, 118].

It is important to note here that $\alpha=\sqrt{\Theta/Ld}$ increases as time goes (i.e., as d goes down), and therefore $\tau=\frac{1}{\alpha L+1}$ decreases as time goes. This lesson instructs us that gradient steps should be given more weights than mirror steps, when it is closer to the optimum.[17]

[16] We remark here that this is essentially the way to choose α in mirror descent, see (4.8).

[17] One may find this counter-intuitive because when it is closer to the optimum, the observed gradients will become smaller, and therefore mirror steps should perform well due to our conceptual message in the introduction. This understanding is incorrect for two reasons. First, when it is closer to the optimum, the threshold between 'large' and 'small' gradients also become smaller, so one cannot rely only on mirror steps. Second, when it is closer to the optimum, mirror steps are more 'unstable' and may increase the objective more (in comparison to the current distance to the optimum), and thus should be given less weight.

Algorithm 1 $\texttt{AGM}(f, w, x_0, T)$

Input: f a differentiable and convex function on Q that is L-smooth with respect to $\|\cdot\|$;

w the DGF function that is 1-strongly convex with respect to the same $\|\cdot\|$ over Q;

x_0 some initial point; and T the number of iterations.

Output: y_T such that $f(y_T) - f(x^*) \leq \frac{4\Theta L}{T^2}$.

$V_x(y) \stackrel{\text{def}}{=} w(y) - \langle \nabla w(x), y - x\rangle - w(x)$.

$y_0 \leftarrow x_0, \quad z_0 \leftarrow x_0$.

for $k \leftarrow 0$ **to** $T-1$ **do**

  $\alpha_{k+1} \leftarrow \frac{k+2}{2L}$, and $\tau_k \leftarrow \frac{1}{\alpha_{k+1}L} = \frac{2}{k+2}$.

  $x_{k+1} \leftarrow \tau_k z_k + (1-\tau_k) y_k$.

  $y_{k+1} \leftarrow \mathsf{Grad}(x_{k+1}) \quad \triangleright = \arg\min_{y \in Q} \left\{ \frac{L}{2}\|y - x_{k+1}\|^2 + \langle \nabla f(x_{k+1}), y - x_{k+1}\rangle \right\}$

  $z_{k+1} \leftarrow \mathsf{Mirr}_{z_k}\big(\alpha_{k+1} \nabla f(x_{k+1})\big)$ $\triangleright$
  $= \arg\min_{z \in Q} \left\{ V_{z_k}(z) + \langle \alpha_{k+1} \nabla f(x_{k+1}), z - z_k\rangle \right\}$

end for

return y_T.

Conclusion. Equipped with the basic knowledge of gradient descent and mirror descent, the above proof is quite straightforward and also gives intuition to how the two 'magic numbers' α and τ are selected. We are unaware of any similar accelerated gradient method that uses fixed step length like ours (when the objective is not known to be strongly convex).

However, this simple algorithm has several caveats. First, the value α depends on the knowledge of Θ; second, a good initial distance bound d has to be specified; and third, the algorithm has to be restarted. In the next section, we choose α and τ differently between iterations, in order to extend the above analysis to allow Q to be constrained, as well as overcome the mentioned caveats.

4.4 Final Accelerated Gradient Method with Variable Step Lengths

In this section, we recover the main result of [118] in the unconstrained case, that is

Theorem 4.6. *If $f(x)$ is L-smooth with respect to $\|\cdot\|$ on Q, and $w(x)$ is 1-strongly convex with respect to the same $\|\cdot\|$ on Q, the algorithm $\texttt{AGM}(f, w, x_0, T)$ in Algorithm 1 ensures*

$$f\big(y_T\big) - f(x^*) \leq \frac{4\Theta L}{T^2} \ .$$

Here, recall from Section 4.2.2 that Θ is any upper bound on $V_{x_0}(x^*)$.

We remark here that it is very important to allow the norm $\|\cdot\|$ to be general, rather than focusing on the ℓ_2-norm as in [117]. See our discussion in Appendix 4.A.1.

This time, we start from $x_0 = y_0 = z_0$, and in each step $k = 0, 1, \ldots, T-1$, we first compute $x_{k+1} \leftarrow \tau_k z_k + (1-\tau_k) y_k$ and then (as illustrated in Algorithm 1)

- perform a gradient step $y_{k+1} \leftarrow \mathsf{Grad}(x_{k+1})$, and
- perform a mirror step $z_{k+1} \leftarrow \mathsf{Mirr}_{z_k}\big(\alpha_{k+1} \nabla f(x_{k+1})\big)$.

Here, α_{k+1} is the step length of the mirror descent and its choice will become clear at the end of this section (and indeed increasing as time goes, similar to the warm-up case). The value of τ_k is chosen as $\frac{1}{\alpha_{k+1} L}$ comparing to $\frac{1}{\alpha_{k+1} L + 1}$ in the warm-up case, in order to capture the constrained case $Q \neq \mathbb{R}^n$. Our eventual choice of α_{k+1} will ensure that $\tau_k \in (0, 1]$ for each k.

We state the counterpart of Lemma 4.4, whose proof can be found in Appendix 4.C:

Lemma 4.7. *If $\tau_k = \frac{1}{\alpha_{k+1} L}$, then it satisfies that for every $u \in Q$,*

$$\begin{aligned} \alpha_{k+1} \langle \nabla f(x_{k+1}), z_k - u \rangle &\overset{①}{\leq} \alpha_{k+1}^2 L \mathsf{Prog}(x_{k+1}) + V_{z_k}(u) - V_{z_{k+1}}(u) \\ &\overset{②}{\leq} \alpha_{k+1}^2 L \big(f(x_{k+1}) - f(y_{k+1}) \big) + V_{z_k}(u) - V_{z_{k+1}}(u) \ . \end{aligned}$$

We state the counterpart of Lemma 4.5, whose proof is only slightly different from Lemma 4.5 because we are using $\tau_k = \frac{1}{\alpha_{k+1} L}$ rather than $\tau = \frac{1}{\alpha L + 1}$, and can be found in Appendix 4.C:

Lemma 4.8 (Coupling)**.** *For any $u \in Q$,*

$$\big(\alpha_{k+1}^2 L\big) f(y_{k+1}) - \big(\alpha_{k+1}^2 L - \alpha_{k+1}\big) f(y_k) + \big(V_{z_{k+1}}(u) - V_{z_k}(u)\big) \leq \alpha_{k+1} f(u) \ .$$

Finally, we only need to set the sequence of α_k so that $\alpha_k^2 L \approx \alpha_{k+1}^2 L - \alpha_{k+1}$ as well as $\tau_k = 1/\alpha_{k+1} L \in (0, 1]$. For instance, we can let $\alpha_k = \frac{k+1}{2L}$ so that $\alpha_k^2 L = \alpha_{k+1}^2 L - \alpha_{k+1} + \frac{1}{4L}$.

Proof of Theorem 4.6. After telescoping Lemma 4.8 with $k = 0, 1, \ldots, T-1$ we obtain that

$$\alpha_T^2 L f(y_T) + \textstyle\sum_{k=1}^{T-1} \frac{1}{4L} f(y_k) + \big(V_{z_T}(u) - V_{z_0}(u)\big) \leq \sum_{k=1}^{T} \alpha_k f(u) \ .$$

By choosing $u = x^*$, we notice that $\sum_{k=1}^{T} \alpha_k = \frac{T(T+3)}{4L}$, $f(y_k) \geq f(x^*)$, $V_{z_T}(u) \geq 0$ and $V_{z_0}(x^*) \leq \Theta$. Therefore, we obtain

$$\tfrac{(T+1)^2}{4L^2} L f(y_T) \leq \Big(\tfrac{T(T+3)}{4L} - \tfrac{T-1}{4L} \Big) f(x^*) + \Theta \ ,$$

which after simplification implies $f(y_T) \leq f(x^*) + \frac{4\Theta L}{(T+1)^2}$. □

Let us make two remarks.

- First, our accelerated method `AGM` is almost the same to that of Nesterov [118], with the following (minor) differences: (1) we use mirror steps instead of dual

averaging steps,[18] (2) we allow arbitrary starting points x_0, and (3) we use $\tau_k = \frac{2}{k+2}$ rather than $\tau_k = \frac{2}{k+3}$.

- This method is very different from the (perhaps better-known) version of Nesterov [117], which is only applicable to the ℓ_2 Euclidean case, and is known by some authors as the 'momentum analysis' or 'momentum method' [123, 156]. To the best of our knowledge, the momentum analysis does not apply to non-Euclidean spaces.

4.5 Strong Convexity Version of Accelerated Gradient Method

When the objective $f(\cdot)$ is both σ-strongly convex and L-smooth with respect to the same norm $\|\cdot\|_2$, another version of accelerated gradient method exists and achieves a $\log(1/\varepsilon)$ convergence rate [117, Theorem 2.2.2]. We show in this section that, our method $\texttt{AGM}(f, w, x_0, T)$ can be used to recover that strong-convexity accelerated method in one of the two ways. Therefore, the gradient-mirror coupling interpretation behind our paper still applies to the strong-convexity accelerated method.

One way to recover the strong-convexity accelerated method is to replace the use of the mirror-descent analysis on the regret term by its strong-convexity counterpart (also known as logarithmic-regret analysis, see for instance [73, 144]). This would incur some different parameter choices on α_k and τ_k, and results in an algorithm similar to that of [117].

Another, but simpler way is to recursively apply Theorem 4.6. In light of the definition of strong convexity and Theorem 4.6, we have

$$\tfrac{\sigma}{2}\|y_T - x^*\|_2^2 \le f(y_T) - f(x^*) \le \tfrac{4\cdot\frac{1}{2}\|x_0-x^*\|_2^2\cdot L}{T^2} \ .$$

In particular, in every $T = T_0 \stackrel{\text{def}}{=} \sqrt{8L/\sigma}$ iterations, we can halve the distance $\|y_T - x^*\|_2^2 \le \frac{1}{2}\|x_0 - x^*\|_2^2$. If we repeatedly invoke $\texttt{AGM}(f, w, \cdot, T_0)$ a sequence of ℓ times, each time feeding the initial vector x_0 with the previous output y_{T_0}, then in the last run of the T_0 iterations, we have

$$f(y_{T_0}) - f(x^*) \le \tfrac{4\cdot\frac{1}{2^\ell}\|x_0-x^*\|_2^2\cdot L}{T_0^2} = \tfrac{1}{2^{\ell+1}}\|x_0 - x^*\|_2^2\cdot\sigma \ .$$

By choosing $\ell = \log\big(\frac{\|x_0-x^*\|_2^2\cdot\sigma}{\varepsilon}\big)$, we conclude that

Corollary 4.9. *If $f(\cdot)$ is both σ-strongly convex and L-smooth with respect to $\|\cdot\|_2$, in a total of $T = O\big(\sqrt{\frac{L}{\sigma}}\cdot\log\big(\frac{\|x_0-x^*\|_2^2\cdot\sigma}{\varepsilon}\big)\big)$ iterations, we can obtain some x such that $f(x) - f(x^*) \le \varepsilon$.*

This is slightly better than the result $O\big(\sqrt{\frac{L}{\sigma}}\cdot\log\big(\frac{\|x_0-x^*\|_2^2\cdot L}{\varepsilon}\big)\big)$ in [117, Theorem 2.2.2].

[18] We are unaware of the existence of this mirror-descent version of Nesterov's accelerated method recorded anywhere.

We remark here that O'Donoghue and Candès [123] have studied some heuristic adaptive restarting techniques which suggest that the above (and other) restarting version of the accelerated method practically outperforms the original method of Nesterov.

Acknowledgements

We thank Jon Kelner and Yin Tat Lee for helpful conversations, and Aaditya Ramdas for pointing out a typo in the previous version of this paper.

This material is based upon work partly supported by the National Science Foundation under Grant CCF-1319460 and by a Simons Graduate Student Award under grant no. 284059.

Appendix

4.A Several Remarks on First-Order Methods

4.A.1 Importance of Non-Euclidean Norms

Let us use a simple example to illustrate the importance of allowing arbitrary norms in studying first-order methods.

Consider the saddle point problem of $\min_{x\in\Delta_n}\max_{y\in\Delta_m} y^T Ax$, where A is an $m\times n$ matrix, $\Delta_n = \{x\in\mathbb{R}^n : x\geq 0 \wedge \mathbb{1}^T x = 1\}$ is the unit simplex in $\mathbb{R}^n$, and $\Delta_m = \{y\in\mathbb{R}^m : y\geq 0 \wedge \mathbb{1}^T y = 1\}$. This problem is important to study because it captures packing and covering linear programs that have wide applications in many areas of computer science, see the discussion in [7] or Chapter 5 of this thesis.

Letting $\mu = \frac{\varepsilon}{2\log m}$, Nesterov [118] has shown that the following objective

$$f_\mu(x) \stackrel{\text{def}}{=} \mu\log\big(\frac{1}{m}\sum_{j=1}^{m}\exp^{\frac{1}{\mu}(Ax)_j}\big) \ ,$$

when optimized over $x\in\Delta_n$, can yield an additive $\varepsilon/2$ solution to the original saddle point problem.

This $f_\mu(x)$ is proven to be $\frac{1}{\mu}$-smooth with respect to the ℓ_1-norm over Δ_n, if all the entries of A are between $[-1,1]$. Instead, $f_\mu(x)$ is $\frac{1}{\mu}$-smooth with respect to the ℓ_2-norm over Δ_n, *only if* the sum of squares of every row of A is at most 1. This ℓ_2 condition is certainly stronger and less natural than the ℓ_1 condition, and the ℓ_1 condition one leads to the fastest (approximate) width-dependent positive LP solver (see the discussion in [7] or Chapter 5 of this thesis).

Different norm conditions also yield different gradient and mirror descent steps. For instance, in the ℓ_1-norm case, the gradient step is $x'\leftarrow\arg\min_{x'\in\Delta_n}\big\{\frac{1}{2}\|x'-x\|_1^2+\alpha\langle\nabla f_\mu(x),x'-x\rangle\big\}$, and the mirror step is $x'\leftarrow\arg\min_{x'\in\Delta_n}\big\{\sum_{i\in[n]}x'_i\log\frac{x'_i}{x_i}+\alpha\langle\nabla f_\mu(x),x'-x\rangle\big\}$. In the ℓ_2-norm case, gradient and mirror steps are both of the form $x'\leftarrow\arg\min_{x'\in\Delta_n}\big\{\frac{1}{2}\|x'-x\|_2^2+\alpha\langle\nabla f_\mu(x),x'-x\rangle\big\}$.

One can find other applications as well in [118] for the use of non-Euclidean norms, and an interesting example of ℓ_∞-norm gradient descent for nearly-linear time maximum flow in [88].

It is now important to note that, the methods in [116, 117] work only for the ℓ_2-norm case, and it is not clear how the proof can be generalized to other norms until [118]. Some other proofs (such as Fercoq and Richtárik [61]) only work for the ℓ_2-norm because the mirror steps are described as (a scaled version of) gradient steps.

4.A.2 Multiplicative Weight Updates as Mirror Descent

The multiplicative weight update (MWU) method (see the survey of Arora, Hazan and Kale [10]) is a simple method that has been repeatedly discovered in theory of computation, machine learning, optimization, and game theory. The setting of this method is the following.

Let $\Delta_n = \{x \in \mathbb{R}^n : x \geq 0 \wedge \mathbb{1}^T x = 1\}$ be the unit simplex in $\mathbb{R}^n$, and we call any vector in Δ_n an *action*. A player is going to play T actions $x_0, \dots, x_{T-1} \in \Delta_n$ in a row; only after playing x_k, the player observes a loss vector $\ell_k \in \mathbb{R}^n$ that may depend on x_k, and suffers from a loss value $\langle \ell_k, x_k \rangle$. The MWU method ensures that, if $\|\ell_k\|_\infty \leq \rho$ for all $k \in [T]$, then the player has an (adaptive) strategy to choose the actions such that the average *regret* is bounded:

$$\frac{1}{T}\Big(\sum_{i=0}^{T-1} \langle \ell_k, x_k \rangle - \min_{u \in \Delta_n} \sum_{i=0}^{T-1} \langle \ell_k, u \rangle\Big) \leq O\Big(\frac{\rho\sqrt{\log n}}{\sqrt{T}}\Big) \ . \tag{4.13}$$

The left hand side is called the average regret because it is the (average) difference between the suffered loss $\sum_{i=0}^{T-1} \langle \ell_k, x_k \rangle$, and the loss $\sum_{i=0}^{T-1} \langle \ell_k, u \rangle$ of the best action $u \in \Delta_n$ in hindsight. Another way to interpret (4.13) is to state that we can obtain an average regret of ε using $T = O(\frac{\rho^2 \log n}{\varepsilon^2})$ rounds.

The above result can be proven directly using mirror descent. Letting $w(x) \stackrel{\text{def}}{=} \sum_i x_i \log x_i$ be the entropy DGF over the simplex $Q = \Delta_n$, and its corresponding Bregman divergence $V_x(x') \stackrel{\text{def}}{=} \sum_{i \in [n]} x'_i \log \frac{x'_i}{x_i}$, we consider the following update rule.

Start from $x_0 = (1/n, \dots, 1/n)$, and update $x_{k+1} = \mathsf{Mirr}_{x_k}\big(\alpha \ell_k\big)$, or equivalently, $x_{k+1,i} = x_{k,i} \cdot \exp^{-\alpha \ell_{k,i}} / Z_k$, where $Z_k > 0$ is the normalization factor that equals to $\sum_{i=1}^n x_{k,i} \cdot \exp^{-\alpha \ell_{k,i}}$.[19] Then, the mirror-descent guarantee (4.7) implies that[20]

$$\forall u \in \Delta_n, \quad \alpha \langle \ell_k, x_k - u \rangle \leq \frac{\alpha^2}{2} \|\ell_k\|_\infty^2 + V_{x_k}(u) - V_{x_{k+1}}(u) \ .$$

After telescoping the above inequality for all $k = 0, 1, \dots, T-1$, and using the upper

[19] This version of the MWU is often known as the Hedge rule [65]. Another commonly used version is to choose $x_{k+1,i} = \frac{x_{k,i}(1-\alpha \ell_{k,i})}{Z_k}$. Since $e^{-t} \approx 1 - t$ whenever $|t|$ is small and our choice of α will make sure that $|\alpha \ell_{k,i}| \ll 1$, this is essentially identical to the Hedge rule.

[20] To be precise, we have replaced $\partial f(x_k)$ with ℓ_k. It is easy to see from the proof of (4.7) that this loss vector ℓ_k does not need to come from the subgradient of some objective $f(\cdot)$.

bounds $\|\ell(x_k)\|_\infty \le \rho$ and $V_{x_0}(u) \le \log n$, we obtain that for all $u \in \Delta_n$,

$$\frac{1}{T}\sum_{k=0}^{T-1}\langle \ell_k, x_k - u\rangle \le \frac{\alpha\rho^2}{2} + \frac{\log n}{\alpha T} \ .$$

Setting $\alpha = \frac{\sqrt{\log n}}{\rho\sqrt{T}}$ we arrive at the desired average regret bound (4.13).

In sum, we have re-deduced the MWU method from mirror descent, and the above proof is quite different from most of the classical analysis of MWU (e.g., [131, 65, 9, 10]). It can be generalized to solve the matrix version of MWU [126, 10], as well as to incorporate the width-reduction technique [131, 10]. We ignore such extensions here because they are outside the scope of this paper.

4.A.3 Partial Equivalence Between Mirror Descent and Dual Averaging

In this section, we show the (folklore) equivalence between mirror descent and dual averaging in two special cases: i) when $Q = \mathbb{R}^\kappa$ and w is a general regularizer, and ii) when $Q = \{x \ge 0 : \mathbb{1}^T x = 1\}$ is the n-dimensional simplex and w is the entropy regularizer. In fact, this equivalence holds more generally for all regularizers $w(\cdot)$ that are convex function of Legendre type with domain Q (see for instance [22, 136]).

Letting $\xi_i = \alpha_i \nabla f(x_i)$ be the observed (scaled) gradient at step i, the dual averaging method can be described as

$$\forall k \in [T], \quad x_k = \arg\min_{y\in Q}\left\{ w(y) + \sum_{i=0}^{k-1}\langle \xi_i, y - x_i\rangle \right\} \ . \tag{4.14}$$

The mirror descent method (with starting point $\tilde{x}_0 = \arg\min_{y\in Q}\{w(y)\}$) can be described as

$$\forall k \in [T], \quad \tilde{x}_k = \arg\min_{y\in Q}\left\{ V_{\tilde{x}_{k-1}}(y) + \langle \xi_{k-1}, y - \tilde{x}_{k-1}\rangle \right\} \ , \tag{4.15}$$

where as before, $V_x(y) \stackrel{\text{def}}{=} w(y) - \langle \nabla w(x), y - x\rangle - w(x)$ is the Bregman divergence of $w(\cdot)$.

Unconstrained Case. If $Q = \mathbb{R}^n$, by taking the derivative from (4.14), we obtain that $\nabla w(x_k) = -\sum_{i=0}^{k-1}\xi_i$. On the other hand, by taking the derivative from (4.15), we obtain that

$$\nabla V_{\tilde{x}_{k-1}}(\tilde{x}_k) = -\xi_{k-1} \iff \nabla w(\tilde{x}_k) - \nabla w(\tilde{x}_{k-1}) = -\xi_{k-1} \ .$$

Combining this with the fact that $\nabla w(\tilde{x}_0) = 0$, we conclude that $\nabla w(\tilde{x}_k) = -\sum_{i=0}^{k-1}\xi_i$. This finishes the proof of $\tilde{x}_k = x_k$ in the unconstrained $Q = \mathbb{R}^n$ case, because the solution x to $\nabla w(x) = -\sum_{i=0}^{k-1}\xi_i$ must be unique for a strongly convex function $w(\cdot)$.

Simplex Case. If $Q = \{x \ge 0 : \mathbb{1}^T x = 1\}$ is the simplex, $\|\cdot\| = \|\cdot\|_1$ is the ℓ_1-norm, $w(x) = \sum_i x_i \log x_i$ is the entropy regularizer, we can precisely compute according to

(4.14) and (4.15) that for every iteration k and coordinate $j \in [n]$,

$$x_{k,j} = \frac{\exp^{-\sum_{i=0}^{k-1} \ell_{i,j}}}{Z_k} \quad \text{and} \quad \tilde{x}_{k,j} = \frac{\tilde{x}_{k-1,j} \cdot \exp^{-\ell_{k,j}}}{\tilde{Z}_k} \ ,$$

where Z_k and $\tilde{Z}_k$ are normalization constants that ensure $\mathbb{1}^T x_k = \mathbb{1}^T \tilde{x}_k = 1$. It is a simple exercise to verify that $x_k = \tilde{x}_k$ for every k.

4.A.4 Deducing the Mirror-Descent Guarantee via Gradient Descent

In this section, we re-deduce the convergence rate of mirror descent from gradient descent. In particular, we show that the dual averaging steps are equivalent to gradient steps on the Fenchel dual of the regularized regret, and deduce the same convergence bound as (4.9). (Similar proof can also be obtained for mirror steps but is notationally more involved.)

Given a sequence of points $x_0, \ldots, x_{T-1} \in Q$, the (scaled) regret with respect to any point $u \in Q$ is $R(x_0, \ldots, x_{T-1}, u) \stackrel{\text{def}}{=} \sum_{i=0}^{T-1} \alpha \langle \partial f(x_i), x_i - u \rangle$. Since it satisfies that $\alpha T \cdot (f(\bar{x}) - f(u)) \leq R(x_0, \ldots, x_{T-1}, u)$, the average regret (after scaling) upper bounds on the distance between any point $f(u)$ and the average $\bar{x} = \frac{1}{T}(x_0 + \cdots + x_{T-1})$. Consider now the regularized regret

$$\hat{R}(x_0, \ldots, x_{T-1}) \stackrel{\text{def}}{=} \max_{u \in Q} \Big\{ \sum_{i=0}^{T-1} \alpha \langle \partial f(x_i), x_i - u \rangle - w(u) \Big\} \ ,$$

and we can rewrite it using the Fenchel dual $w^*(\lambda) \stackrel{\text{def}}{=} \max_{u \in Q} \{\langle \lambda, u \rangle - w(u)\}$ of $w(\cdot)$:

$$\hat{R}(x_0, \ldots, x_{T-1}) = w^* \Big(-\alpha \sum_{i=0}^{T-1} \partial f(x_i) \Big) + \sum_{i=0}^{T-1} \alpha \langle \partial f(x_i), x_i \rangle \ .$$

The classical theory of Fenchel duality tells us that $w^*(\lambda)$ is 1-smooth with respect to the dual norm $\|\cdot\|_*$, because $w(\cdot)$ is 1-strongly convex with respect to $\|\cdot\|$. We also have $\nabla w^*(\lambda) = \arg\max_{u \in Q} \{\langle \lambda, u \rangle - w(u)\}$. (See for instance [143].)

With enough notations introduced, let us now minimize $\hat{R}$ by intelligently selecting $x_0, \ldots, x_{T-1}$. Perhaps a little counter-intuitively, we start from $x_0 = \cdots = x_{T-1} = x^*$ and accordingly $\partial f(x^*) = 0$ (if there are multiple subgradients at x^*, choose the zero one). This corresponds to a regret value of zero and a regularized regret $\hat{R}(x^*, \ldots, x^*) = w^*(0) = -\min_{u \in Q} \{w(u)\}$.

Next, we choose the values of $x_0, \ldots, x_{T-1}$ one by one. We choose $x_0 = \arg\min_{u \in Q} \{w(u)\}$ as the starting point.[21] Suppose that the values of $x_0, \ldots, x_{k-1}$ are already determined, and we are ready to pick $x_k \in Q$. Let us compute the changes in the regular-

[21] Dual averaging steps typically demand the first point x_0 to be at the minimum of the regularizer $w(\cdot)$, because that leads to the cleanest analysis. This can be relaxed to allow an arbitrary starting point.

ized regret as a function of x_k:

$$\begin{aligned}\Delta\hat{R} &= \hat{R}(x_0,\dots,x_k,x^*,\dots,x^*) - \hat{R}(x_0,\dots,x_{k-1},x^*,\dots,x^*)\\ &= w^*\Big(-\alpha\sum_{i=0}^{k}\partial f(x_i)\Big) - w^*\Big(-\alpha\sum_{i=0}^{k-1}\partial f(x_i)\Big) + \alpha\langle\partial f(x_k),x_k\rangle\\ &\le \Big\langle\nabla w^*\Big(-\alpha\sum_{i=0}^{k-1}\partial f(x_i)\Big), -\alpha\partial f(x_k)\Big\rangle + \frac{1}{2}\big\|\alpha\partial f(x_k)\big\|_*^2 + \alpha\langle\partial f(x_k),x_k\rangle\ .\end{aligned} \tag{4.16}$$

Here, the last inequality is because $w^*(a)-w^*(b)\le\langle\nabla w^*(b),a-b\rangle+\frac{1}{2}\|a-b\|_*^2$, owing to the smoothness of $w^*(\cdot)$. At this moment, it is clear to see that if one chooses

$$x_k = \nabla w^*\Big(-\alpha\sum_{i=0}^{k-1}\partial f(x_i)\Big) = \arg\min_{u\in Q}\Big\{w(u)+\sum_{i=0}^{k-1}\alpha\langle\partial f(x_i),u\rangle\Big\}\ ,$$

the first and third terms in (4.16) cancel out, and we obtain $\Delta\hat{R}\le\frac{1}{2}\big\|\alpha\partial f(x_k)\big\|_*^2$. [22] In other words, the regularized regret increases by no more than $\frac{1}{2}\big\|\alpha\partial f(x_k)\big\|_*^2\le\alpha^2\rho^2/2$ in each step, so in the end we have $\hat{R}(x_0,\dots,x_{T-1})\le -w(x_0)+\alpha^2\rho^2T/2$.

In sum, by the definition of the regularized regret, we have

$$\begin{aligned}\alpha T\cdot(f(\bar{x})-f(x^*)) - w(x^*) &\le \sum_{i=0}^{T-1}\alpha\langle\partial f(x_i),x_i-x^*\rangle - w(x^*) \le \hat{R}(x_0,\dots,x_{T-1})\\ &\le -w(x_0)+\frac{\alpha^2\rho^2T}{2}\ .\end{aligned}$$

This implies the following upper bound on the optimality of $f(\bar{x})$

$$f(\bar{x})-f(x^*)\le\frac{\alpha\rho^2}{2}+\frac{w(x^*)-w(x_0)}{\alpha T} = \frac{\alpha\rho^2}{2}+\frac{V_{x_0}(x^*)}{\alpha T}\le\frac{\alpha\rho^2}{2}+\frac{\Theta}{\alpha T}\ .$$

Finally, choosing $\alpha=\frac{\sqrt{2\Theta}}{\rho\cdot\sqrt{T}}$ to be the step length, we arrive at $f(\bar{x})-f(x^*)\le\frac{\sqrt{2\Theta}\cdot\rho}{\sqrt{T}}$, which is the same convergence rate as (4.9).

4.B Missing Proof of Section 4.2

For the sake of completeness, we provide self-contained proofs of the mirror descent and mirror descent guarantees in this section.

4.B.1 Missing Proof for Gradient Descent

[22] This essentially proves (4.5) in the introduction after scaling: $\Delta\hat{R}=\alpha(k+1)\max_u\tilde{R}_{k+1}(u)-\alpha k\max_u\tilde{R}_k(u)$.

Gradient Descent Guarantee

$$f(\mathsf{Grad}(x)) \le f(x) - \mathsf{Prog}(x) \qquad (4.6)$$

or in the special case when $Q = \mathbb{R}^n$ $\qquad f(\mathsf{Grad}(x)) \le f(x) - \frac{1}{2L}\|\nabla f(x)\|_*^2 \ .$

Proof. [23] Letting $\tilde{x} = \mathsf{Grad}(x)$, we prove the first inequality by

$$\mathsf{Prog}(x) = -\min_{y\in Q}\Big\{\frac{L}{2}\|y-x\|^2 + \langle \nabla f(x), y-x\rangle\Big\} = -\Big(\frac{L}{2}\|\tilde{x}-x\|^2 + \langle \nabla f(x), \tilde{x}-x\rangle\Big)$$
$$= f(x) - \Big(\frac{L}{2}\|\tilde{x}-x\|^2 + \langle \nabla f(x), \tilde{x}-x\rangle + f(x)\Big) \le f(x) - f(\tilde{x}) \ .$$

Here, the last inequality is a consequence of the smoothness assumption: for any $x, y \in Q$,

$$f(y) - f(x) = \int_{\tau=0}^{1} \langle \nabla f(x+\tau(y-x)), y-x\rangle d\tau$$
$$= \langle \nabla f(x), y-x\rangle + \int_{\tau=0}^{1} \langle \nabla f(x+\tau(y-x)) - \nabla f(x), y-x\rangle d\tau$$
$$\le \langle \nabla f(x), y-x\rangle + \int_{\tau=0}^{1} \|\nabla f(x+\tau(y-x)) - \nabla f(x)\|_* \cdot \|y-x\| d\tau$$
$$\le \langle \nabla f(x), y-x\rangle + \int_{\tau=0}^{1} \tau L\|y-x\| \cdot \|y-x\| d\tau = \langle \nabla f(x), y-x\rangle + \frac{L}{2}\|y-x\|^2$$

The second inequality follows because in the special case of $Q = \mathbb{R}^n$, we have

$$\mathsf{Prog}(x) = -\min_{y\in Q}\Big\{\frac{L}{2}\|y-x\|^2 + \langle \nabla f(x), y-x\rangle\Big\} = \frac{1}{2L}\|\nabla f(x)\|_*^2 \ . \qquad \square$$

Fact 4.10 (Gradient Descent Convergence)**.** *Let $f(x)$ be a convex, differentiable function that is L-smooth with respect to $\|\cdot\|$ on $Q = \mathbb{R}^n$, and x_0 any initial point in Q. Consider the sequence of T gradient steps $x_{k+1} \leftarrow \mathsf{Grad}(x_k)$, then the last point x_T satisfies that*

$$f(x_T) - f(x^*) \le O\big(\frac{LR^2}{T}\big) \ ,$$

where $R = \max_{x:f(x)\le f(x_0)} \|x - x^*\|$, and x^* is any minimizer of f.

Proof. [24] Recall that we have $f(x_{k+1}) \le f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|_*^2$ from (4.6). Furthermore, by the convexity of f and Cauchy-Schwarz we have

$$f(x_k) - f(x^*) \le \langle \nabla f(x_k), x_k - x^*\rangle \le \|\nabla f(x_k)\|_* \cdot \|x_k - x^*\| \le R \cdot \|\nabla f(x_k)\|_* \ .$$

Letting $D_k = f(x_k) - f(x^*)$ denote the distance to the optimum at iteration k, we now obtain two relationships $D_k - D_{k+1} \ge \frac{1}{2L}\|\nabla f(x_k)\|_*^2$ as well as $D_k \le R \cdot \|\nabla f(x_k)\|_*$.

[23] This proof can be found for instance in the textbook [117].

[24] Our proof follows almost directly from Nesterov [117], but he only uses the Euclidean ℓ_2 norm.

Combining these two, we get

$$D_k^2 \le 2LR^2(D_k - D_{k+1}) \implies \frac{D_k}{D_{k+1}} \le 2LR^2\Big(\frac{1}{D_{k+1}} - \frac{1}{D_k}\Big) .$$

Noticing that $D_k \ge D_{k+1}$ because our objective only decreases at every round, we obtain that $\frac{1}{D_{k+1}} - \frac{1}{D_k} \ge \frac{1}{2LR^2}$. Finally, we conclude that at round T, we must have $\frac{1}{D_T} \ge \frac{T}{2LR^2}$, finishing the proof that $f(x_T) - f(x^*) \le \frac{2LR^2}{T}$. □

4.B.2 Missing Proof for Mirror Descent

Mirror Descent Guarantee

If $x_{k+1} = \mathsf{Mirr}_{x_k}\big(\alpha \cdot \partial f(x_k)\big)$, then

$$\forall u \in Q, \quad \alpha(f(x_k) - f(u)) \le \alpha\langle \partial f(x_k), x_k - u\rangle \le \frac{\alpha^2}{2}\|\partial f(x_k)\|_*^2 + V_{x_k}(u) - V_{x_{k+1}}(u) \ . \tag{4.7}$$

Proof. [25] we compute that

$$\begin{aligned}
\alpha\langle \partial f(x_k), x_k - u\rangle &= \langle \alpha\partial f(x_k), x_k - x_{k+1}\rangle + \langle \alpha\partial f(x_k), x_{k+1} - u\rangle \\
&\overset{①}{\le} \langle \alpha\partial f(x_k), x_k - x_{k+1}\rangle + \langle -\nabla V_{x_k}(x_{k+1}), x_{k+1} - u\rangle \\
&\overset{②}{=} \langle \alpha\partial f(x_k), x_k - x_{k+1}\rangle + V_{x_k}(u) - V_{x_{k+1}}(u) - V_{x_k}(x_{k+1}) \\
&\overset{③}{\le} \Big(\langle \alpha\partial f(x_k), x_k - x_{k+1}\rangle - \frac{1}{2}\|x_k - x_{k+1}\|^2\Big) + \big(V_{x_k}(u) - V_{x_{k+1}}(u)\big) \\
&\overset{④}{\le} \frac{\alpha^2}{2}\|\partial f(x_k)\|_*^2 + \big(V_{x_k}(u) - V_{x_{k+1}}(u)\big)
\end{aligned}$$

Here, ① is due to the minimality of $x_{k+1} = \arg\min_{x\in Q}\{V_{x_k}(x) + \langle \alpha\partial f(x_k), x\rangle\}$, which implies that $\langle \nabla V_{x_k}(x_{k+1}) + \alpha\partial f(x_k), u - x_{k+1}\rangle \ge 0$ for all $u \in Q$. ② is due to the triangle equality of Bregman divergence.[26] ③ is because $V_x(y) \ge \frac{1}{2}\|x - y\|^2$ by the strong convexity of the DGF $w(\cdot)$. ④ is by Cauchy-Schwarz. □

4.C Missing Proofs of Section 4.4

Lemma 4.7. *If $\tau_k = \frac{1}{\alpha_{k+1}L}$, then it satisfies that for every $u \in Q$,*

$$\alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u\rangle \overset{①}{\le} \alpha_{k+1}^2 L\mathsf{Prog}(x_{k+1}) + V_{z_k}(u) - V_{z_{k+1}}(u)$$

[25] This proof can be found for instance in the textbook [27].

[26] That is,

$$\begin{aligned}
\forall x, y \ge 0, \quad \langle -\nabla V_x(y), y - u\rangle &= \langle \nabla w(x) - \nabla w(y), y - u\rangle \\
&= (w(u) - w(x) - \langle \nabla w(x), u - x\rangle) - (w(u) - w(y) - \langle w(y), u - y\rangle) \\
&\quad - (w(y) - w(x) - \langle \nabla w(x), y - x\rangle) \\
&= V_x(u) - V_y(u) - V_x(y) \ .
\end{aligned}$$

$$\overset{②}{\leq} \alpha_{k+1}^2 L\big(f(x_{k+1}) - f(y_{k+1})\big) + V_{z_k}(u) - V_{z_{k+1}}(u) \ .$$

Proof. The second inequality ② is again from the gradient descent guarantee $f(x_{k+1}) - f(y_{k+1}) \geq \mathsf{Prog}(x_{k+1})$. To prove ①, we first write down the key inequality of mirror-descent analysis (whose proof is identical to that of (4.7))

$$\begin{aligned}
&\alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u\rangle = \langle \alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle + \langle \alpha_{k+1}\nabla f(x_{k+1}), z_{k+1} - u\rangle \\
&\overset{①}{\leq} \langle \alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle + \langle -\nabla V_{z_k}(z_{k+1}), z_{k+1} - u\rangle \\
&\overset{②}{=} \langle \alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle + V_{z_k}(u) - V_{z_{k+1}}(u) - V_{z_k}(z_{k+1}) \\
&\overset{③}{\leq} \Big(\langle \alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2\Big) + \big(V_{z_k}(u) - V_{z_{k+1}}(u)\big)
\end{aligned}$$

Here, ① is due to the minimality of $z_{k+1} = \arg\min_{z\in Q}\{V_{z_k}(z) + \langle \alpha_{k+1}\nabla f(x_{k+1}), z\rangle\}$, which implies that $\langle \nabla V_{z_k}(z_{k+1}) + \alpha_{k+1}\nabla f(x_{k+1}), u - z_{k+1}\rangle \geq 0$ for all $u \in Q$. ② is due to the triangle equality of Bregman divergence (see Footnote 26 in Appendix 4.B). ③ is because $V_x(y) \geq \frac{1}{2}\|x-y\|^2$ by the strong convexity of the $w(\cdot)$.

If one stops here and uses Cauchy-Shwartz $\langle \alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2 \leq \frac{\alpha_{k+1}^2}{2}\|\nabla f(x_{k+1})\|_*^2$, he will get the desired inequality in the special case of $Q = \mathbb{R}^n$, because $\mathsf{Prog}(x_{k+1}) = \frac{1}{2L}\|\nabla f(x_{k+1})\|_*^2$ from (4.6).

For the general unconstrained case, we need to use the special choice of $\tau_k = 1/\alpha_{k+1}L$ follows. Letting $v \overset{\text{def}}{=} \tau_k z_{k+1} + (1-\tau_k)y_k \in Q$ so that $x_{k+1} - v = (\tau_k z_k + (1-\tau_k)y_k) - v = \tau_k(z_k - z_{k+1})$, we have

$$\begin{aligned}
&\langle \alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2 \\
&= \langle \frac{\alpha_{k+1}}{\tau_k}\nabla f(x_{k+1}), x_{k+1} - v\rangle - \frac{1}{2\tau_k^2}\|x_{k+1} - v\|^2 \\
&= \alpha_{k+1}^2 L\left(\langle \nabla f(x_{k+1}), x_{k+1} - v\rangle - \frac{L}{2}\|x_{k+1} - v\|^2\right) \leq \alpha_{k+1}^2 L\mathsf{Prog}(x_{k+1})
\end{aligned}$$

where the last inequality is from the definition of $\mathsf{Prog}(x_{k+1})$. □

Lemma 4.8 (Coupling)**.** *For any $u \in Q$,*

$$\big(\alpha_{k+1}^2 L\big)f(y_{k+1}) - \big(\alpha_{k+1}^2 L - \alpha_{k+1}\big)f(y_k) + \big(V_{z_{k+1}}(u) - V_{z_k}(u)\big) \leq \alpha_{k+1}f(u) \ .$$

Proof. We deduce the following sequence of inequalities

$$\begin{aligned}
&\alpha_{k+1}\big(f(x_{k+1}) - f(u)\big) \\
&\leq \alpha_{k+1}\langle \nabla f(x_{k+1}), x_{k+1} - u\rangle \\
&= \alpha_{k+1}\langle \nabla f(x_{k+1}), x_{k+1} - z_k\rangle + \alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u\rangle \\
&\overset{①}{=} \frac{(1-\tau_k)\alpha_{k+1}}{\tau_k}\langle \nabla f(x_{k+1}), y_k - x_{k+1}\rangle + \alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u\rangle \\
&\overset{②}{\leq} \frac{(1-\tau_k)\alpha_{k+1}}{\tau_k}(f(y_k) - f(x_{k+1})) + \alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u\rangle
\end{aligned}$$

$$\overset{③}{\leq} \frac{(1-\tau_k)\alpha_{k+1}}{\tau_k}(f(y_k) - f(x_{k+1})) + \alpha_{k+1}^2 L\big(f(x_{k+1}) - f(y_{k+1})\big) + V_{z_k}(u) - V_{z_{k+1}}(u)$$

$$\overset{④}{=} \big(\alpha_{k+1}^2 L - \alpha_{k+1}\big) f(y_k) - \big(\alpha_{k+1}^2 L\big) f(y_{k+1}) + \alpha_{k+1} f(x_{k+1}) + \big(V_{z_k}(u) - V_{z_{k+1}}(u)\big)$$

Here, ① uses the choice of x_{k+1} that satisfies $\tau_k(x_{k+1} - z_k) = (1-\tau_k)(y_k - x_{k+1})$; ② is by the convexity of $f(\cdot)$ and $1 - \tau_k \geq 0$; ③ uses Lemma 4.7; and ④ uses the choice of $\tau_k = 1/\alpha_{k+1}L$. □

Chapter 5

Using Optimization to Solve Positive LPs Faster in Parallel

This chapter is based on the result published in [7], and its further edits can be found at:

http://arxiv.org/abs/1407.1925.

Positive linear programs (LP), also known as packing and covering linear programs, are an important class of problems that bridges computer science, operations research, and optimization. Despite the consistent efforts on this problem, all known nearly-linear-time algorithms require $\tilde{O}(\varepsilon^{-4})$ iterations to converge to $1 \pm \varepsilon$ approximate solutions. This ε^{-4} dependence has not been improved since 1993, and limits the performance of parallel implementations for such algorithms. Moreover, previous algorithms and their analyses rely on update steps and convergence arguments that are combinatorial in nature and do not seem to arise naturally from an optimization viewpoint.

In this paper, we leverage new insights from optimization theory to construct a novel algorithm that breaks the longstanding ε^{-4} barrier. Our algorithm has a simple analysis and a clear motivation. Our work introduces a number of novel techniques, such as the combined application of gradient descent and mirror descent, and a truncated, smoothed version of the standard multiplicative weight update, which may be of independent interest.

5.1 Introduction

Fractional packing and covering linear programs (LP) are described with non-negative matrices, non-negative constraints, and non-negative variables. They are also known as positive linear programs as originally studied by Luby and Nisan [101].

A generic packing LP takes the form $\max\{c^T x : Ax \leq b\}$ where $c \in \mathbb{R}^n_{\geq 0}$, $b \in \mathbb{R}^m_{\geq 0}$, and $A \in \mathbb{R}^{m \times n}_{\geq 0}$; similarly, a covering LP can be written as $\min\{b^T y : A^T y \geq c\}$, with

the same requirements on A, b, and c. As in other works, we assume without loss of generality that the LP is in its *standard form*: $b = \mathbb{1}$ and $c = \mathbb{1}$:[1]

$$\text{Packing LP:} \qquad \max_{x \geq 0}\{\mathbb{1}^T x \,:\, Ax \leq \mathbb{1}\}\ , \tag{5.1}$$

$$\text{Covering LP:} \qquad \min_{y \geq 0}\{\mathbb{1}^T y \,:\, A^T y \geq \mathbb{1}\}\ . \tag{5.2}$$

Since the two programs are dual to each other, we denote by OPT their shared optimal value. We say that x is a $(1-\varepsilon)$-approximation for the packing LP if $Ax \leq \mathbb{1}$ and $\mathbb{1}^T x \geq (1-\varepsilon)\mathsf{OPT}$, and y a $(1+\varepsilon)$-approximation for the covering LP if $A^T y \geq \mathbb{1}$ and $\mathbb{1}^T y \leq (1+\varepsilon)\mathsf{OPT}$.

Of course, it is possible to adopt the general Interior Point or Ellipsoid Methods to obtain approximate solvers with a $\log(1/\varepsilon)$ dependence on the number of iterations. However, the computational cost of such algorithms is typically very high, as each iteration requires the solution of a system of linear equations in $A^T A$. As a consequence, this approach is simply not suitable to the solution of large-scale problems.

To address this issue, researchers have developed iterative *approximate* solvers that achieve a better dependence on the problem size at the cost of having a $\mathsf{poly}(1/\varepsilon)$ dependence on the approximation parameter ε. These algorithms rely crucially on the power of *multiplicative weight update methods* (see the survey by Arora, Hazan and Kale [10]). Multiplicative weight update methods can be viewed as special cases of the *mirror descent* method, a widely-used first-order method in optimization (see for instance [5] or Chapter 4 for this relationship). Such methods achieve fast running times by eschewing any structure in the problem and only accessing the instance in a restricted, quick fashion through the computation of gradients of the objective.

As a result, iterative approximate solvers often require a larger number of iterations, i.e., one that depends on $\mathsf{poly}(1/\varepsilon)$, but each iteration consists only of a small number of simple steps (such as matrix-vector multiplications or sorting operations) and requires only nearly-linear work in N and $O(\log N)$ depth, even in the weak EREW model of the Parallel Random Access Machine (PRAM).

Such fast approximate positive-LP solvers have been widely used in approximation algorithms (e.g., MinSetCover [101], MaxSet, MaxDiCut, Max-k-CSP [158], bipartite matching), probabilistic checkable proofs [158], zero-sum matrix games [118], scheduling [131], graph embedding [131], flow controls [24, 25], auction mechanisms [169], wireless sensor networks [38], and many other areas. In addition, techniques developed in this line of research have also inspired many other important results, most notably regarding fast algorithms for multi-commodity flow problems [131, 63, 68, 103, 20].

Previous approximate solvers can be further divided into two classes.

[1] This can be achieved simply by scaling.

Width-Dependent Solvers. These algorithms[2] require a number of iterations that is at least linearly dependent on $\rho \cdot \mathsf{OPT}$, where ρ is the largest entry, i.e. the *width*, of matrix A. Since $\mathsf{OPT} \geq 1/\rho$, this value $\rho \cdot \mathsf{OPT}$ is at least 1. However, since OPT can easily be as large as 1 or even more than n, the resulting running time is not polynomial, but only pseudo-polynomial. In particular, positive LPs can be solved in $O(\frac{\rho^2 \mathsf{OPT}^2 \log m}{\varepsilon^2})$ iterations [131], or $O(\frac{\rho \mathsf{OPT} \log m}{\varepsilon^2})$ iterations using negative-width techniques [10]. These algorithms strongly rely on multiplicative weight updates and only require "oracle-access" to the matrix A.

When A is given explicitly like in this paper, the number of iterations can be reduced to $O(\frac{\rho \mathsf{OPT} \log m}{\varepsilon})$ by deploying more advanced optimization tools such as Nesterov's accelerated gradient method [118], or Nemirovski's mirror prox method [113]. It is also worth noting that Bienstock and Iyengar [32] have converted this dependence on $\rho\mathsf{OPT}$ into a more benign, yet linear dependence on n. More specifically, their iteration count is $O(\varepsilon^{-1}\sqrt{Kn \log m})$ where K is the maximum number of non-zeros per row of A. This is $O(\varepsilon^{-1} n\sqrt{\log m})$ in the worst case.

Width-Independent Solvers. In this paper, we are interested in a second, more efficient class of methods, i.e. *width-independent*,[3] truly polynomial-time approximate solvers (see Table 5.1).

This line of research was initiated by a seminal paper of Luby and Nisan [101], who were able to remove the dependence from the width and give an algorithm running in $O\big(\frac{\log^2 N}{\varepsilon^4}\big)$ iterations. Theirs is the first *nearly-linear-time* approximate solver for positive LPs and also the first to run in parallel in nearly-linear-work and polylogarithmic depth. This algorithm was later simplified and made explicit for parallelization by Bartal, Byers and Raz [24], improved to allow mixed packing and covering by Young [165], and generalized by Awerbuch and Khandekar [17] to the computational model where processors are restricted to be 'stateless'. These solvers are *parallelizable* because they only require $O(\mathsf{polylog}(N)/\varepsilon^{O(1)})$ iterations to converge to $1 \pm \varepsilon$ approximate solutions. They are nearly-linear time because each iteration runs in nearly-linear time.

A separate line of work starting from Bartal, Byers and Raz [24, 25] eschews the parallelization constraint to design *sequential* width-independent solvers with a better ε dependence. At high level, these algorithms modify the candidate LP solutions

[2]Note that most width-dependent solvers are studied under the minmax form of positive LPs:

$$\min_{\substack{x \geq 0 \\ \mathbb{1}^T x = 1}} \max_{\substack{y \geq 0 \\ \mathbb{1}^T y = 1}} y^T A x \ ,$$

whose optimal value equals $1/\mathsf{OPT}$. Their approximation guarantees are often written in terms of the *additive* error. We have translated their performances to the multiplicative error for a fair comparison.

[3]Some of these solvers may still have a $\mathsf{polylog}(\rho)$ dependence. Since each occurrence of $\log(\rho)$ can typically be replaced with $\log(nm)$ after slightly modifying the instance matrix A, we have done so in Table 5.1 for a fair comparisons.

Problem	Paper	Total Work	Number of Iterations[a]	Notes
p/c LP	[101]	$\frac{\log^2 N}{\varepsilon^4} \times (N \log n)$	$\frac{\log^2 N}{\varepsilon^4}$	
p/c LP	[24, 25]	$\frac{\log^3 N}{\varepsilon^4} \times N$	$\frac{\log^3 N}{\varepsilon^4}$	
p/c LP	[165]	$\frac{\log^3 N}{\varepsilon^4} \times N$	$\frac{\log^3 N}{\varepsilon^4}$	mixed p/c
p/c LP	[17]	$\frac{\log^4 N}{\varepsilon^5} \times N$	$\frac{\log^4 N}{\varepsilon^5}$	stateless
p/c LP	**[this paper]**	$\frac{\log^2 N}{\varepsilon^3} \times N$	$\frac{\log^2 N}{\varepsilon^3}$	semi-stateless
p/c LP	[165]	$\frac{\log N}{\varepsilon^2} \times (md + N)$ [b]	$\frac{\log N}{\varepsilon^2} \times (n+m)$	not parallelizable
p/c LP	[166]	$\frac{\log N}{\varepsilon^2} \times N$	$\frac{\log N}{\varepsilon^2} \times (n+m)$	not parallelizable
p/c LP	[92]	$\frac{\log N}{\varepsilon^2} \times (n+m) + N$	$\frac{\log N}{\varepsilon^2} \times (n+m)$	not parallelizable
p LP	[6]	$\frac{\log N \log \varepsilon^{-1}}{\varepsilon} \times N$	$\frac{\log N \log \varepsilon^{-1}}{\varepsilon} \times n$	not parallelizable
c LP	[6]	$\frac{\log N \log \varepsilon^{-1}}{\varepsilon^{1.5}} \times N$	$\frac{\log N \log \varepsilon^{-1}}{\varepsilon^{1.5}} \times n$	not parallelizable

Table 5.1: Comparisons among width-independent approximate solvers for positive LPs.

[a] For most parallelizable solvers, an iteration is dominated by a matrix-vector multiplicative that can be implemented in $O(N)$ total work. However, an iteration of Luby-Nisan is more complicated, and to the best of our knowledge, we only know how to implement it in $O(nm)$ or $O(N \log n)$ total work, rather than $O(N)$.

[b] d is the maximum number of constraints each variable is in; md may be larger than N.

coordinate by coordinate and therefore require at least a linear number of iterations to converge. For instance, the algorithm of Koufogiannakis and Young [92] runs in nearly-linear total time $O\big(N + \frac{\log N}{\varepsilon^2} \times (n+m)\big)$, but requires $O(\frac{\log N}{\varepsilon^2}(n+m))$ iterations to converge to $1 \pm \varepsilon$ approximate solutions. In contrast, as we shall discuss later in Section 5.1.1, parallelizable solvers modify all coordinates of the candidate LP solution *at once* per iteration, thus converging in a much smaller polylogarithmic number of iterations. For this reason, the design of parallelizable solvers faces different technical challenges from that of sequential ones, because the update rules are much more restrictive. We have summarized prior results on sequential solvers in Table 5.1.

To sum up, despite the amount of work in this area, the $O\big(\frac{\log^2 N}{\varepsilon^4}\big)$-iteration-count has not been improved since the original paper of Luby and Nisan. This lack of progress constitutes a significant limitation, as the ε^{-4}-dependence on the approximation parameter ε is particularly pour. The question of how to go beyond ε^{-4} has been raised by Young [165] and remained open until now. In this paper, we give an answer to this question and provide a brief empirical evaluation supporting the idea that the performance gains achieved by our algorithm in the worst-case actually translate into practice.

5.1.1 Our Results

In this paper, we present an algorithm $\texttt{PosLPSolver}(A, \varepsilon)$ that runs only in $O(\frac{\log n \cdot \log(nm/\varepsilon)}{\varepsilon^3})$ iterations, and each iteration consists mostly of a matrix-vector multiplication so can be implemented in $O(\log N)$ parallel depth. This is a total work of $O\big(\frac{\log n \cdot \log(nm/\varepsilon)}{\varepsilon^3} \cdot N\big)$. (See a full comparison between our and previous results in Table 5.1.) Besides being the fastest parallel algorithm for solving positive LPs to date, our method also is surprisingly simple and enjoys a 'semi-stateless' property, i.e. is stateless except for requiring a global clock (see Appendix 5.B).

Our algorithms works by optimizing a relaxation of the original packing LP (see Definition 5.1), where the hard constraint $Ax \leq 1$ is replaced by an exponential penalty function for violating the constraint.[4] This initial step ensures that our candidate iterative solutions remain approximately feasible throughout the evolution of the algorithm. It also leads us to optimize our modified objective by updating our current iterate $x^{(k)}$ using gradient information. This is done by computing a feedback vector v so that $v_i \stackrel{\text{def}}{=} \sum_{j=1}^{m} A_{i,j} \cdot \exp^{\frac{1}{\mu}((Ax)_j - 1)} - 1 \in [-1, \infty)$ for each variable $i \in [n]$, and performing a multiplicative update $x_i \leftarrow x_i \cdot \exp^{-\alpha \cdot \mathbb{T}(v_i)}$. Here, our thresholding function $\mathbb{T}(v) = v$ for $v \in [-1, 1] \setminus [-\varepsilon, \varepsilon]$, $\mathbb{T}(v) = 0$ for $v \in [-\varepsilon, \varepsilon]$, and $\mathbb{T}(v) = 1$ for $v > 1$; and $\alpha = \frac{\varepsilon\mu}{4}$ is some fixed constant.

Our Techniques. Our result fundamentally differs from all previous width-independent solvers both in the algorithm specification and in its analysis. Like previous works, we also update the coordinates of x simultaneously and multiplicatively. However, previous methods treat all relevant coordinates alike, multiplying each of them either by $1 + \alpha$ or $1 - \alpha$, for some fixed constant α. Instead, our use of the feedback vector v (along with the thresholding function) allows us to update the coordinates by a factor between $e^{\pm\alpha} \approx 1 \pm \alpha$ and $e^{\pm\varepsilon\alpha} \approx 1 \pm \varepsilon\alpha$. This *discriminative multiplicative update* rule is a key step in overcoming the $1/\varepsilon^4$ barrier.

More importantly, our work introduces a completely novel way of analyzing the performance of our algorithm. More specifically, previous methods [101, 24, 165, 68] fall into the following framework: the method is divided into $\tilde{\Omega}(\frac{1}{\varepsilon^2})$ *phases*, with each phase having a different parameter setting. Each phase itself consists of $\tilde{\Omega}(\frac{1}{\varepsilon^2})$ iterations. This immediately prevents their analyses from breaking the $\frac{1}{\varepsilon^4}$ barrier[5].

In contrast, we interpret the packing LP problem as a purely optimization question, i.e., to minimize $f(x)$ for some convex function f. Next, in each iteration of the algorithm, we interpret the feedback vector v as the gradient $\nabla f(x) \in [-1, \infty)^n$, and divide it into two components, the large component $\eta \in [0, \infty)^n$ and the small (and truncated) component $\xi \in [-1, 1]^n$, satisfying $\nabla f(x) \approx \eta + \xi$. The key observation now is to interpret our update $x_i \leftarrow x_i \cdot \exp^{-\alpha \cdot \mathbb{T}(v_i)}$ as performing two different kind

[4] This standard technique in optimization is used explicitly in [17] and implicitly in [101] and [165].

[5] Although the algorithm in [17] does not explicitly require phases, its convergence analysis divides the iterations into $\Omega(\frac{\log^2 N}{\varepsilon^3})$ phases each with $\Omega(\frac{\log^2 N}{\varepsilon^2})$ iterations.

of steps at the same time:

- a *"gradient descent"*[6] step (on η), to ensure that $f(x)$ decreases by a large amount at each step; and
- a *mirror descent* step (on ξ), to ensures that the average *regret* of the history of the steps is small.

Both gradient and mirror descent are well-known tools from optimization (see for instance [117, 27] and, for starters, mirror descent is a generalization of multiplicative weight updates). This 'duality' view allows us to combine the analysis of both gradient and mirror descent for a faster algorithm, and is the key to bypass the combinatorial/phaseful analysis used by all previous results. More generally, the same authors of this paper observed that gradient and mirror descent have complementary performances, and *coupling* these two methods often leads to better running times [5] (see also Chapter 4).

We develop two more techniques that may be of independent interests, one for the gradient descent analysis and one for the mirror descent analysis. In our gradient descent view, since $f(x)$ does not satisfy any Lipschitz gradient property, the classical convergence analysis of gradient descent (see [117]) no longer applies.[7] Instead, we adopt a *multiplicative Lipschitz gradient property*: if each coordinate of x changes multiplicatively by a little, the gradient does not change too much multiplicatively as well. This property enables us to produce a promise on the decrease of the objective $f(x)$ in each step.

In our mirror descent analysis, we have developed a *gradient truncation* technique that removes large components from the gradient, delegating their contribution to the gradient descent analysis. This effectively reduces the width experienced by our mirror descent algorithm.

Finally, we emphasize that our optimization view for solving positive LPs should be seen as yet another example on *designing combinatorial algorithms based on insights from optimization*. Before our work, the updates on x are maximally aggressive, since they arise naturally from a combinatorial approach to the solution of the original LP program. In our algorithm, we have smoothed out the updates on x so that, for coordinates whose absolute feedbacks $|v_i|$ are small, we perform less aggressive steps. While one may find such intuition very legitimate, without the optimization interpretation behind it, it is very hard to analyze the resulting algorithm or even to find the *right* step length. For instance, the algorithm of [17] is similar to ours

[6]It is important to note here that we have generalized the notion of "gradient descent" to indicate any descent step that is guaranteed to decrease the objective. This is in contrast to mirror descent, that does not necessarily decrease the objective at each iteration.

[7]The Lipschitz gradient property (also known as Lipschitz smooth property in the literature) says that $\|\nabla f(x_1) - \nabla f(x_2)\| \leq L \cdot \|x_1 - x_2\|$ for some constant L and some special choice of norm. If one forces $f(x)$ to satisfy this property, the algorithm falls into the category of [118] and becomes width-dependent.

in terms of the updates on x. However, the simple difference between the choices of step length makes our algorithm faster than theirs, $\log^2 N/\varepsilon^3$ vs. $\log^4 N/\varepsilon^5$. Moreover, our step lengths are in fact *less aggressive* than theirs in terms of decreasing the objective $f(x)$. We also provide an empirical evaluation in Appendix 5.A to support this comparison.

The Stateless Feature. Some parallelizable algorithms enjoy a desirable *stateless* feature. Informally, this feature requires that the updates of each processor only depend on the current feedback, and not on the history or on any global variable. The only known stateless solver for positive LPs is due to Awerbuch and Khandekar [17], but their method is much slower than that of Luby and Nisan (see Table 5.1). Stateless algorithms enjoy a number of features (P1) *self-stabilization*, (P2) *robustness against incremental adjustments*, and (P3) *no global clock*. We point out that our algorithm is 'semi-stateless' (introduced in Appendix 5.B): that is, it exhibits properties (P1) and (P2). Unfortunately, our current proof technique requires the use of a global clock for the parallelized algorithm. Instead, [17] only requires that the desired number of iterations are performed synchronously with the global clock, while between consecutive iterations each processor can run on its own arbitrarily without synchronization.

5.1.2 Roadmap

We transfer the positive LP problem into an optimization question in Section 5.2, provide our packing LP solver in Section 5.3, and turn the same algorithm into a covering LP solver in Section 5.4. We also provide a brief empirical evaluation comparing the performance of our algorithm against previous ones in Appendix 5.A. We defer the argument of the semi-statelessness of our LP solver to Appendix 5.B. Some missing proofs are included in the appendix.

5.2 Smoothing the Positive LP Objective

In this section we introduce the smoothed objective $f_\mu(x)$ that we are going to minimize in order to approximately solve the packing LP, by turning each row of the LP constraint $Ax \leq \mathbb{1}$ into an exponential penalty function so that we only need to require $x \geq 0$ throughout the algorithm.

Let x^* be any optimal solution of the packing LP (5.1). Throughout this paper, we use indices $i \in [n]$ for the columns of A, and $j \in [m]$ for the rows of A. We denote by $A_{\diamond i}$ the i-th column vector of A, and $A_{j\diamond}$ the j-th row vector of A. We assume without loss of generality that

$$\min_{i\in[n]}\{\|A_{\diamond i}\|_\infty\} = 1 \ , \tag{5.3}$$

since otherwise one can scale A by a constant factor, and the solution OPT as well as x^* are only affected by this same constant factor.

We now introduce our smoothed objective $f_\mu(x)$.

Definition 5.1. *Letting parameter* $\mu \stackrel{\text{def}}{=} \frac{\varepsilon}{4\log(nm/\varepsilon)}$*, we define the smoothed objective* $f_\mu(x)$ *as*

$$f_\mu(x) \stackrel{\text{def}}{=} \mu \sum_{j=1}^{m} \exp^{\frac{1}{\mu}((Ax)_j - 1)} - \mathbb{1}^T x \ .$$

We wish to study the *minimization* problem on $f_\mu(x)$, subject to the constraint that each coordinate $x_i \geq 0$ is non-negative. We denote by $x \geq 0$ this positive orthant.

Intuitively this objective $f_\mu(x)$ should capture the original packing LP (5.1) approximately as follows. On one hand, we want to maximize $\mathbb{1}^T x$ so the negative term $-\mathbb{1}^T x$ shows up in $f_\mu(x)$. On the other, if $(Ax)_j \geq 1 + \varepsilon$ for some j, the exponential penalty in $f_\mu(x)$ introduces a value that is at least $\exp^{\varepsilon/\mu} = (nm/\varepsilon)^4$ and very large. This means $Ax \leq (1+\varepsilon)\mathbb{1}$ must be true if the objective $f_\mu(x)$ is small.

We wish to point out that this is very different from the softmax function implicitly used in [165], and is used as a potential function in [17]. More precisely, the standard softmax function can be seen to arise as the Legendre dual of the negative entropy over the simplex, while our potential function is actually the Legendre dual of the negative *generalized* entropy over the positive quadrant. Our specific choice of this objective enables us to deduce what we call the multiplicative Lipschitz gradient property, described in (5.7).

We begin with several simple but important properties about OPT and $f_\mu(x)$. In short, they together imply that the minimum of $f_\mu(x)$ is around $-\mathsf{OPT}$, and if one can approximately find the minimum of $f_\mu(x)$ (up to an error $O(\varepsilon\mathsf{OPT})$), this corresponds to a $(1 - O(\varepsilon))$-approximate solution to the packing LP (5.1). Notice that we will not be able to directly obtain a covering solution from this objective, and thus more techniques will be introduced in Section 5.4.

Proposition 5.2.

(a) $\mathsf{OPT} \in [1, n]$.
(b) Letting $x = (1 - \varepsilon/2)x^* \geq 0$*, we have* $f_\mu(x) \leq -(1-\varepsilon)\mathsf{OPT}$.
(c) Letting $x^{(0)} \geq 0$ *be such that* $x_i^{(0)} = \frac{1-\varepsilon/2}{n\|A_{\circ i}\|_\infty}$ *for each* $i \in [n]$*, we have* $f_\mu(x^{(0)}) \leq -\frac{1-\varepsilon}{n}$.
(d) For any $x \geq 0$ *satisfying* $f_\mu(x) \leq 0$*, we must have* $Ax \leq (1+\varepsilon)\mathbb{1}$*, and thus* $\mathbb{1}^T x \leq (1+\varepsilon)\mathsf{OPT}$.
(e) If $x \geq 0$ *satisfies* $f_\mu(x) \leq -(1 - O(\varepsilon))\mathsf{OPT}$*, then* $\frac{1}{1+\varepsilon}x$ *is a* $(1 - O(\varepsilon))$*-approximate solution to the packing LP.*
(f) The gradient of $f_\mu(x)$ *can be written as*

$$\nabla f_\mu(x) = A^T y(x) - \mathbb{1} \quad \textit{where} \quad y_j(x) \stackrel{\text{def}}{=} \exp^{\frac{1}{\mu}((Ax)_j - 1)} \ . \tag{5.4}$$

(The proofs are straightforward and can be found in Appendix 5.C.)

Algorithm 2 `PosLPSolver`(A, ε)

Input: $A \in \mathbb{R}_{\geq 0}^{m \times n}, \varepsilon \in (0, 1/10]$.
Output: $x \in \mathbb{R}_{\geq 0}$ and $\bar{y} \in \mathbb{R}_{\geq 0}^{m}$.

$\mu \leftarrow \frac{\varepsilon}{4\log(nm/\varepsilon)}$ and $\alpha \leftarrow \frac{\varepsilon\mu}{4}$. ▷ parameters
$x_i^{(0)} \leftarrow \frac{1-\varepsilon/2}{n\|A_{\diamond i}\|_\infty}$ for all $i \in [n]$. ▷ initial vector $x^{(0)}$
$T \leftarrow \frac{6\log(2n)}{\alpha\varepsilon}$. ▷ number of iterations
for $k \leftarrow 0$ **to** $T-1$ **do**
    for $i \leftarrow 1$ **to** n **do**
        Compute the feedback $v_i \leftarrow \sum_{j=1}^{m} A_{i,j} \cdot \exp^{\frac{1}{\mu}((Ax)_j - 1)} - 1$
        ▷ in fact, $v_i = \nabla_i f_\mu(x^{(k)}) = \langle A_{\diamond i}, y(x^{(k)})\rangle - 1 \in [-1, \infty)$.
        Update: $x_i^{(k+1)} \leftarrow x_i^{(k)} \cdot \exp^{-\alpha \cdot \mathbb{T}(v_i)}$. ▷ see Definition 5.3 for the definition of $\mathbb{T}(v)$
    end for
end for
return $\frac{x^{(T)}}{1+\varepsilon}$ and $\bar{y} = \sum_{i=0}^{T-1} y(x^{(k)})$. ▷ recall that $y_j(x) \stackrel{\text{def}}{=} \exp^{\frac{1}{\mu}((Ax)_j - 1)}$

5.3 Parallelizable Packing LP Solver

In this section we prove the approximation and convergence guarantee on our packing LP algorithm. Although the same algorithm also produces a good covering LP solution, we defer such analysis to Section 5.4 because different techniques are required.

To describe our algorithm we first make the following choice of thresholding function

Definition 5.3. *The thresholding function* $\mathbb{T}\colon [-1, \infty) \to [-1, 1]$ *is defined as follows*

$$\mathbb{T}(v) \stackrel{\text{def}}{=} \begin{cases} 0, & v \in [-\varepsilon, \varepsilon]; \\ v, & v \in [-1, 1] \setminus [-\varepsilon, \varepsilon]; \\ 1, & v > 1. \end{cases}$$

Our algorithm is presented in Algorithm 2, and each of its iterations can be described with $x_i^{(k+1)} \leftarrow x_i^{(k)} \cdot \exp^{-\alpha \cdot \mathbb{T}(v_i)}$, where we choose $\alpha = \varepsilon\mu/4$ to be the step length. (Throughout this paper, we use superscript $x^{(k)}$ to represent vector x at iteration k, and subscript x_i to represent the i-th coordinate of vector x.)

Our proof of the correctness of `PosLPSolver` is divided into three steps.

Step I: Gradient Descent. We interpret (see Section 5.3.1 for details) each update $x_i^{(k+1)} \leftarrow x_i^{(k)} \cdot \exp^{-\alpha \cdot \mathbb{T}(v_i)}$ as a gradient descent step,[8] and show that the objective $f_\mu(x)$ does not increase, or more strongly, always decreases by at least the following amount:

[8] To be clear, in some literature, the gradient descent is referred only to $x \leftarrow x - c \cdot \nabla f(x)$ for some constant c. In this paper, we adopt the more general notion, and refer it to any step that directly decreases $f(x)$.

Lemma 5.4 (Gradient Descent)**.** *For any step k in* `PosLPSolver`*, letting $B^{(k)} \subseteq [n]$ be the set of indices i such that $\nabla_i f_\mu(x^{(k)}) \geq 1$, the objective $f_\mu(x)$ decreases by at least*

$$f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) \geq \frac{\alpha}{4} \cdot \textstyle\sum_{i \in B^{(k)}} x_i^{(k)} \cdot \nabla_i f_\mu\big(x^{(k)}\big) \geq 0 \ .$$

Combining this with Proposition 5.2.c, we have $f_\mu(x^{(k)}) \leq 0$ for all k.

Note that the above gradient descent lemma does *not* follow from any classical theory because our objective $f_\mu(x)$ does not satisfy any good Lipschitz gradient property. Instead, we define and use a *multiplicative Lipschitz gradient property* for our objective, which may be of independent interest.

Step II: Mirror Descent. We interpret (see Section 5.3.2 for details) each update $x_i^{(k+1)} \leftarrow x_i^{(k)} \cdot \exp^{-\alpha \cdot \mathbb{T}(v_i)}$ as a mirror descent step.

A *mirror descent step* in optimization is any step from x to x' that is of the form $x' \leftarrow \arg\min_z \{V_x(z) + \langle \alpha \nabla f(x), z - x \rangle\}$. Here, $\alpha > 0$ is some step length, and $V_x(\tilde{x}) = w(\tilde{x}) - \langle \nabla w(x), \tilde{x} - x \rangle - w(x)$ is the Bregman divergence of some convex *distance generating function* $w(x)$.[9] In this paper, we pick $w(x) \stackrel{\text{def}}{=} \sum_{i \in [n]} x_i \log x_i - x_i$ to be the generalized entropy function, and accordingly, for every $x, \tilde{x} \geq 0$, let

$$V_x(\tilde{x}) = \textstyle\sum_{i \in [n]} \big(\tilde{x}_i \log \frac{\tilde{x}_i}{x_i} + x_i - \tilde{x}_i \big) \ .$$

After verifying that our update is a mirror descent step, the next lemma easily follows from the general theory of mirror descent.

Lemma 5.5 (Mirror Descent)**.** *Letting $\xi_i^{(k)} \stackrel{\text{def}}{=} \mathbb{T}(\nabla_i f_\mu(x^{(k)})) \in [-1, 1]$ be the* truncated gradient*, we have that for any $u \geq 0$,*

$$\langle \alpha \xi^{(k)}, x^{(k)} - u \rangle \leq \alpha^2 \mathsf{OPT} + V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u) \ .$$

We emphasize here that it is important to use the truncated gradient $\xi^{(k)} \in [-1, 1]^n$ in the mirror descent instead of the full gradient $\nabla f_\mu(x^{(k)})$, because the latter may have very large coordinates (whose magnitudes depend on the *width* of the matrix). This is why all previous positive-LP solvers using mirror descent are width-dependent. Our *gradient truncation* technique may be of independent interest.

Step III: Coupling. Finally, as argued in Section 5.3.3, we put together the two lemmas above and derive the following coupled bound:

Lemma 5.6 (Coupling)**.** *For any $u \geq 0$, we have*

$$\begin{aligned} &\alpha(f_\mu(x^{(k)}) - f_\mu(u)) \leq \langle \alpha \nabla f_\mu(x^{(k)}), x^{(k)} - u \rangle \\ &\leq 4(f_\mu(x^{(k)}) - f_\mu(x^{(k+1)})) + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u) \big) + \alpha \cdot 2\varepsilon \mathsf{OPT} + \alpha \cdot \varepsilon \mathbb{1}^T u \ . \end{aligned}$$

Let us point out right away that Lemma 5.6 captures benefit of combining the two

[9] This $w(x)$ is classically chosen to be any *strongly convex* function, such as $w(x) = \frac{1}{2}\|x\|_2^2$ (and in that case $V_x(y) = \frac{1}{2}\|x - y\|_2^2$).

analyses. If $f_\mu(x^{(k)}) - f_\mu(x^{(k+1)})$ is large, we are making a large gradient descent step because the objective greatly decreases. Or, if $f_\mu(x^{(k)}) - f_\mu(x^{(k+1)})$ is small (for a number of consecutive iterations), we can telescoping the above inequality and obtain a good upperbound on the average of $f_\mu(x^{(k)})$.

We are now ready to state and prove our theorem for packing LP.

Theorem 5.7 (Packing LP)**.** *For $T \geq \frac{6\log(2n)}{\alpha\varepsilon} = \Omega(\frac{\log n \cdot \log(nm/\varepsilon)}{\varepsilon^3})$, we have that $f_\mu(x^{(T)}) \leq -(1-5\varepsilon)\mathsf{OPT}$, and as a consequence, $\texttt{PosLPSolver}(A, \varepsilon)$ produces an output $x = \frac{x^{(T)}}{1+\varepsilon}$ that is a $(1-O(\varepsilon))$-approximate solution for the packing LP (5.1).*

Proof. We begin by telescoping the inequality in Lemma 5.6 for $k = 0, 1, \ldots, T-1$, and choosing $u = \tilde{u} \stackrel{\text{def}}{=} (1-\varepsilon/2)x^*$, which satisfies $\mathbb{1}^T u \leq \mathsf{OPT}$ by the definition of x^*:

$$\alpha \sum_{k=0}^{T-1} (f_\mu(x^{(k)}) - f_\mu(\tilde{u})) \leq 4(f_\mu(x^{(0)}) - f_\mu(x^{(T)})) + \big(V_{x^{(0)}}(\tilde{u}) - V_{x^{(T)}}(\tilde{u})\big) + \alpha T \cdot 3\varepsilon\mathsf{OPT} \ . \tag{5.5}$$

Notice that, the second term on the right hand side is upper bounded by

$$\begin{aligned} V_{x^{(0)}}(\tilde{u}) - V_{x^{(T)}}(\tilde{u}) \leq V_{x^{(0)}}(\tilde{u}) &\leq \sum_i \tilde{u}_i \log \frac{\tilde{u}_i}{x_i^{(0)}} + x_i^{(0)} \\ &\leq \sum_i \tilde{u}_i \log \frac{1/\|A_{\diamond i}\|_\infty}{(1-\varepsilon/2)/n\|A_{\diamond i}\|_\infty} + \frac{1-\varepsilon/2}{n\|A_{\diamond i}\|_\infty} \\ &\leq \mathbb{1}^T\tilde{u} \cdot \log(2n) + 1 \leq 2\mathsf{OPT} \cdot \log(2n) \ . \end{aligned} \tag{5.6}$$

Here, we have used the fact that $\tilde{u}_i \leq \frac{1}{\|A_{\diamond i}\|_\infty}$ since $A\tilde{u} \leq \mathbb{1}$.

From here, we want to prove that $f_\mu(x^{(T)}) \leq -(1-5\varepsilon)\mathsf{OPT}$ by way of contradiction. Suppose not, that is, $f_\mu(x^{(T)}) > -(1-5\varepsilon)\mathsf{OPT}$, we have $f_\mu(x^{(0)}) - f_\mu(x^{(T)}) \leq 0 + (1-5\varepsilon)\mathsf{OPT} \leq \mathsf{OPT}$, giving an upper bound on the first term on the right hand side in (5.5). Substituting this and (5.6) to (5.5), and dividing αT on both sides, we get

$$\begin{aligned} \frac{1}{T}\sum_{k=0}^{T-1}(f_\mu(x^{(k)}) - f_\mu(\tilde{u})) &\leq \frac{4}{\alpha T}(f_\mu(x^{(0)}) - f_\mu(x^{(T)})) + \frac{1}{\alpha T}\big(V_{x^{(0)}}(\tilde{u}) - V_{x^{(T)}}(\tilde{u})\big) + 3\varepsilon\mathsf{OPT} \\ &\leq \frac{4\mathsf{OPT}}{\alpha T} + \frac{2\mathsf{OPT}\cdot\log(2n)}{\alpha T} + 3\varepsilon\mathsf{OPT} \ . \end{aligned}$$

Finally, since we have chosen $T \geq \frac{6\log(2n)}{\alpha\varepsilon}$, the above right hand side is no greater than $4\varepsilon\mathsf{OPT}$. This, by an averaging argument, tells us the existence of some $k \in \{0, 1, \ldots, T-1\}$ with $f_\mu(x^{(k)}) \leq f_\mu(\tilde{u}) + 4\varepsilon\mathsf{OPT} \leq -(1-5\varepsilon)\mathsf{OPT}$ (where we have used $f_\mu(\tilde{u}) \leq -(1-\varepsilon)\mathsf{OPT}$ from Proposition 5.2.b). However, it contradicts to the hypothesis that $f_\mu(x^{(T)}) > -(1-5\varepsilon)\mathsf{OPT}$ because $f_\mu(x^{(k)}) \geq f_\mu(x^{(T)})$ according to Lemma 5.4. This finishes the proof that $f_\mu(x^{(T)}) \leq -(1-5\varepsilon)\mathsf{OPT}$. The fact that $\frac{x^{(T)}}{1+\varepsilon}$ provides a $(1-O(\varepsilon))$ approximate solution for the packing LP is due to Proposition 5.2.e. □

5.3.1 The Gradient Descent Lemma

In this section, we are going to view our step $x^{(k)} \to x^{(k+1)}$ as a gradient descent step, and prove Lemma 5.4.

Sketched Proof. Here, we adopt a generalized notion of *gradient descent step*, and say that any step from x to x' that decreases the objective is a gradient descent step. Classically in optimization, if a convex function $f(x)$ satisfies the so-called Lipschitz gradient property, that is, $\|\nabla f(x_1) - \nabla f(x_2)\|_* \leq L \cdot \|x_1 - x_2\|$ for some constant L (with respect to some norm $\|\cdot\|$ and its dual norm $\|\cdot\|_*$), then a gradient descent step can provably decrease the objective by a considerable amount. (We refer interested readers to our survey in [5] or Chapter 4 of this thesis.) Unfortunately, this property is not obeyed by our objective $f_\mu(x)$, so we make use of what we call the *multiplicative Lipschitz gradient* property, that may be of independent interest for convex optimization problems that have enough 'non-negativity'.

In particular, we observe that:

> In each iteration, `PosLPSolver` changes each coordinate of x *multiplicatively* by at most a factor of $1 \pm 4\alpha/3$. Owing to our choice of the smoothed objective $f_\mu(x)$, we can prove that in this iteration, for each i satisfying $|\nabla_i f_\mu(x)| > \varepsilon$, the coordinate gradient
>
> $\nabla_i f_\mu(x)$ is not changed by more than a multiplicative factor of 1 ± 0.5. (5.7)

Denoting by $x = x^{(k)}$ the vector before the update, and $x' = x^{(k+1)}$ the one after, let us now estimate the difference between $f_\mu(x) - f_\mu(x')$ using (5.7), and sketch the proof of Lemma 5.4.

Since $\nabla f_\mu(x)$ is close enough to $\nabla f_\mu(x')$ owing to (5.7), intuitively, we can show that $f_\mu(x) - f_\mu(x')$ is (up to a constant factor) close to $\langle \nabla f_\mu(x), x - x' \rangle$ due to the first-order approximation of $f_\mu(x)$ around x. Now, since $x_i - x'_i$ is positive only when $\nabla_i f_\mu(x)$ is positive, and viceversa, we conclude that the difference $f_\mu(x) - f_\mu(x') \approx \langle \nabla f_\mu(x), x - x' \rangle$ is non-negative.

Furthermore, when focusing only on the coordinates i such that $\nabla_i f_\mu(x) \geq 1$ (i.e., $i \in B^{(k)}$), we have that $x_i - x'_i = x_i(1 - e^{-\alpha}) = \Omega(\alpha) \cdot x_i$. This enables us to conclude that the amount of difference $f_\mu(x) - f_\mu(x')$ is at least $\Omega(\alpha) \cdot \sum_{i \in B_{(k)}} x_i \cdot \nabla_i \mu(x)$, arriving at the conclusion of Lemma 5.4.

Proof Details. The following proposition establishes the formal statement for (5.7).

Proposition 5.8. *If $f_\mu(x^{(k)}) \leq 0$, for any $x = \tau x^{(k)} + (1 - \tau) x^{(k+1)}$ where $\tau \in [0, 1]$:*

(a) $x_i \in x_i^{(k)} \cdot [1 - 4\alpha/3, 1 + 4\alpha/3]$

(b) $y_j(x) \in y_j(x^{(k)}) \cdot [1 - \varepsilon/2, 1 + \varepsilon/2]$

(c) When $|\nabla_i f_\mu(x^{(k)})| \geq \varepsilon$, we have that $\nabla_i f_\mu(x)$ is between $\frac{\nabla_i f_\mu(x^{(k)})}{2}$ and $\frac{3\nabla_i f_\mu(x^{(k)})}{2}$.

Proof.

(a) We can always write $x_i = x_i^{(k)} \cdot e^t$ for some $t \in [-\alpha, \alpha] \subseteq [-1/4, 1/4]$. According to the fact that $e^t \le 1 + 4t/3$ for $t \in [0, 1/4]$ and $e^t \ge 1 - t \ge 1 - 4t/3$ for $t \in [-1/4, 0]$, we must have $x_i \in x_i^{(k)} \cdot [1 - 4\alpha/3, 1 + 4\alpha/3]$.

(b) Recall from (5.4) that $y_j(x) = \exp^{\frac{1}{\mu}((Ax)_j - 1)}$. According to Proposition 5.2.d, we have $(Ax^{(k)})_j \le 1 + \varepsilon$. Now, by the non-negativity of A and the previous item, we have

$$\big|(Ax)_j - (Ax^{(k)})_j\big| \le 4\alpha/3 \cdot (Ax^{(k)})_j \le 4\alpha/3 \cdot (1 + \varepsilon) \le 5\alpha/3 \ .$$

This implies that $y_j(x) \ge y_j(x^{(k)}) \cdot \exp(-5\alpha/3\mu) = y_j(x^{(k)}) \cdot \exp(-5\varepsilon/12) > y_j(x^{(k)}) \cdot (1 - \varepsilon/2)$ for sufficiently small ε, as well as that $y_j(x) \le y_j(x^{(k)}) \cdot \exp(5\alpha/3\mu) < y_j(x^{(k)}) \cdot (1 + \varepsilon/2)$.

(c) Recall from (5.4) that $\nabla_i f_\mu(x) = \big(A^T y(x)\big)_i - 1$. By symmetry, we only prove the case when $\nabla_i f_\mu(x^{(k)}) \ge \varepsilon$, which is equivalent to $\big(A^T y(x^{(k)})\big)_i \ge 1 + \varepsilon$. By the previous item, we immediately have

$$\big(A^T y(x^{(k)})\big)_i (1 + \varepsilon/2) \ge \big(A^T y(x)\big)_i \ge \big(A^T y(x^{(k)})\big)_i (1 - \varepsilon/2) \ .$$

Denoting by $t = \big(A^T y(x^{(k)})\big)_i - 1 \ge \varepsilon$, it is not hard to verify that $(t+1)(1 - \varepsilon/2) \ge t/2 + 1$ and $(t+1)(1+\varepsilon/2) \le 3t/2 + 1$ for all $t \ge \varepsilon$, which then implies

$$\frac{3\nabla_i f_\mu(x^{(k)})}{2} = 3t/2 \ge \big(A^T y(x)\big)_i - 1 \ge t/2 = \frac{\nabla_i f_\mu(x^{(k)})}{2} \qquad \square$$

We can now use the above multiplicative Lipschitz gradient property to prove the desired gradient descent progress promised in Lemma 5.4.

Proof of Lemma 5.4. We prove by induction. Suppose that Lemma 5.4 is true for all indices less than k. This implies, in particular, that $f_\mu(x^{(k)}) \le f_\mu(x^{(k-1)}) \le \cdots \le f_\mu(x^{(0)}) \le 0$.

We compute the objective difference by the standard integral over gradients as follows.

$$\begin{aligned} f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) &= \int_0^1 \Big\langle \nabla f_\mu\big(x^{(k+1)} + \tau(x^{(k)} - x^{(k+1)})\big), x^{(k)} - x^{(k+1)} \Big\rangle d\tau \\ &= \sum_i \int_0^1 \nabla_i f_\mu\big(x^{(k+1)} + \tau(x^{(k)} - x^{(k+1)})\big) d\tau \times (x_i^{(k)} - x_i^{(k+1)}) \ge 0 \end{aligned} \tag{5.8}$$

Here the last inequality is because, whenever $x_i^{(k)} - x_i^{(k+1)}$ is strictly positive (resp. strictly negative) for some coordinate $i \in [n]$, it must be because $\nabla_i f_\mu(x^{(k)}) \ge \varepsilon$ (resp. $\le -\varepsilon$) according to our algorithm. However, owing to Proposition 5.8.c, we have that $f_\mu\big(x^{(k+1)} + \tau(x^{(k)} - x^{(k+1)})\big)$ is also positive (resp. negative) for all $\tau \in [0, 1]$, since $\nabla_i f_\mu(x^{(k)})$ is. (Here we used $f_\mu(x^{(k)}) \le 0$.) This concludes that for each i, the i-th

component in (5.8), denoted by $W_i \stackrel{\text{def}}{=} \int_0^1 \nabla_i f_\mu\big(x^{(k+1)} + \tau(x^{(k)} - x^{(k+1)})\big)d\tau \times (x_i^{(k)} - x_i^{(k+1)})$, is non-negative.

We next turn to lower bounding $f_\mu(x^{(k)}) - f_\mu(x^{(k+1)})$ by computing a lower bound on W_i for each $i \in B^{(k)}$. Indeed, recall that by the definition of our thresholding function $\mathbb{T}(\cdot)$, for each $i \in B^{(k)}$, the update on the i-th coordinate in $x^{(k)}$ is precisely $x_i^{(k+1)} \leftarrow x_i^{(k)} \cdot \exp^{-\alpha}$. In such a case,

$$\begin{aligned} W_i &= (1 - e^{-\alpha})x_i^{(k)} \times \int_0^1 \nabla_i f_\mu\big(x^{(k+1)} + \tau(x^{(k)} - x^{(k+1)})\big)d\tau \\ &\geq (1 - e^{-\alpha})x_i^{(k)} \times \frac{1}{2}\nabla_i f_\mu\big(x^{(k)}\big) \qquad \text{(using Proposition 5.8.c)} \\ &\geq \frac{\alpha}{4} \cdot x_i^{(k)} \cdot \nabla_i f_\mu\big(x^{(k)}\big) \ . \end{aligned}$$

In sum, we conclude that

$$\sum_i W_i \geq \frac{\alpha}{4} \cdot \sum_{i \in B^{(k)}} x_i^{(k)} \cdot \nabla_i f_\mu\big(x^{(k)}\big) \ . \qquad \square$$

5.3.2 The Mirror Descent Lemma

In this section, we are going to view our step $x^{(k)} \to x^{(k+1)}$ as a mirror descent step, and prove Lemma 5.5.

Recall that $\xi_i^{(k)} \stackrel{\text{def}}{=} \mathbb{T}(\nabla_i f_\mu(x^{(k)})) \in [-1, 1]$ is the truncated gradient at step k, and satisfies that $\xi_i^{(k)} = \nabla_i f_\mu(x^{(k)})$ for all coordinates i such that $\nabla_i f_\mu(x^{(k)}) \in [-1, 1] \setminus [-\varepsilon, \varepsilon]$. We can verify that our careful choice of $x^{(k)} \to x^{(k+1)}$ is in fact a mirror descent step on the truncated gradient:

Claim 5.9.

$$x^{(k+1)} = \arg\min_{z \geq 0} \left\{ V_{x^{(k)}}(z) + \langle \alpha \xi^{(k)}, z - x^{(k)} \rangle \right\} \ . \tag{5.9}$$

Proof. This can be verified coordinate by coordinate, because the arg min function is over all possible $z \geq 0$, where this constraint does not impose any inter-coordinate constraint.

In other words, by substituting the definition of $V_{x^{(k)}}(z)$, we only need to verify that

$$x_i^{(k+1)} = \arg\min_{z_i \geq 0} \left\{ \left(z_i \log \frac{z_i}{x_i^{(k)}} + x_i^{(k)} - z_i \right) + \alpha \xi_i^{(k)} \cdot (z_i - x_i^{(k)}) \right\} \stackrel{\text{def}}{=} \arg\min_{z_i \geq 0} \{ g(z_i) \} \ .$$

At this point, the univariate function $g(z_i)$ is convex and has a unique minimizer. Since the gradient $\frac{d}{dz_i} g(z_i) = \log \frac{z_i}{x_i^{(k)}} + \alpha \xi_i^{(k)}$, this unique minimizer is indeed $z_i = x_i^{(k)} \cdot \exp^{-\alpha \xi_i^{(k)}}$, finishing the proof of Claim 5.9. $\square$

After confirming that our iterative step in `PosLPSolver` is indeed a mirror descent step, it is not hard to deduce Lemma 5.5 based on the proof of the classical mirror descent analysis (see for instance [27]). However, we emphasize here that our choice of

the distance generating function $w(x)$ is not strongly convex over the entire positive orthant $\{x \in \mathbb{R}^n : x \geq 0\}$, and thus the our proof is not identical to the classical theory. We have relied on, in fact, a 'local' strong convexity which we introduce and is sufficient for our purpose (see (5.10)).

Proof of Lemma 5.5. We deduce the following sequence of inequalities:

$$\begin{aligned}
&\langle \alpha\xi^{(k)}, x^{(k)} - u\rangle = \langle \alpha\xi^{(k)}, x^{(k)} - x^{(k+1)}\rangle + \langle \alpha\xi^{(k)}, x^{(k+1)} - u\rangle \\
&\overset{①}{\leq} \langle \alpha\xi^{(k)}, x^{(k)} - x^{(k+1)}\rangle + \langle -\nabla V_{x^{(k)}}(x^{(k+1)}), x^{(k+1)} - u\rangle \\
&\overset{②}{=} \langle \alpha\xi^{(k)}, x^{(k)} - x^{(k+1)}\rangle + V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u) - V_{x^{(k)}}(x^{(k+1)}) \\
&\overset{③}{\leq} \sum_i \Big(\alpha\xi_i^{(k)} \cdot (x^{(k)} - x^{(k+1)}) - \frac{|x_i^{(k+1)} - x_i^{(k)}|^2}{2\max\{x_i^{(k+1)}, x_i^{(k)}\}}\Big) + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u)\big) \\
&\overset{④}{\leq} \sum_i \frac{(\alpha^2\xi_i^{(k)})^2 \cdot \max\{x_i^{(k+1)}, x_i^{(k)}\}}{2} + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u)\big) \qquad (5.10) \\
&\overset{⑤}{\leq} \frac{2}{3}\alpha^2 \mathbb{1}^T x^{(k)} + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u)\big) \\
&\overset{⑥}{\leq} \alpha^2 \mathsf{OPT} + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u)\big)
\end{aligned}$$

Here, ① is due to the minimality of $x^{(k+1)}$ in (5.9), which implies that $\nabla V_{x^{(k)}}(x^{(k+1)}) + \alpha\xi^{(k)} = 0$. ② is due to the triangle equality of Bregman divergence:

$$\begin{aligned}
\forall x, y \geq 0, \quad \langle -\nabla V_x(y), y - u\rangle &= \langle \nabla w(x) - \nabla w(y), y - u\rangle \\
&= (w(u) - w(x) - \langle \nabla w(x), u - x\rangle) - (w(u) - w(y) - \langle w(y), u - y\rangle) \\
&\quad - (w(y) - w(x) - \langle \nabla w(x), y - x\rangle) \\
&= V_x(u) - V_y(u) - V_x(y) \ .
\end{aligned}$$

③ is because $V_x(y) = \sum_i y_i \log \frac{y_i}{x_i} + x_i - y_i \geq \sum_i \frac{1}{2\max\{x_i, y_i\}}|x_i - y_i|^2$. ④ is by Cauchy-Schwarz. ⑤ is because we have $x_i^{(k+1)} \leq \frac{4}{3}x_i^{(k)}$ owing to Proposition 5.8.a. ⑥ is because we have $\mathbb{1}^T x^{(k)} \leq \frac{3}{2}\mathsf{OPT}$ owing to Proposition 5.2.d (and $f_\mu(x^{(k)}) \leq 0$ from Lemma 5.5). □

Remark 5.10. The main difference between this proof and its classical counterpart in optimization theory is inequality ③ in (5.10). Recall that $w(x) = \sum_{i=1}^n x_i \log x_i - x_i$. Since $w(x)$ is known to be 1-strongly convex with respect to the ℓ_1-norm over the simplex $\Delta = \{x \geq 0 : \mathbb{1}^T x = 1\}$, we automatically have $V_x(y) \geq \frac{1}{2}\|x - y\|_1^2$ for all $x, y \in \Delta$, and this was the key step used in the classical analysis. In our case, we no longer have this strong convexity because $x, y \notin \Delta$. However, the fact that $V_x(y) \geq \sum_i \frac{1}{2\max\{x_i, y_i\}}|x_i - y_i|^2$ is in fact saying that $w(x)$ is 'locally' 1-strongly convex with respect to the $\|\cdot\|_{x,2}$ norm, defined to be $\|w\|_{x,2}^2 \overset{\text{def}}{=} \sum_i w_i^2/x_i$. This local norm technique is very crucial in our analysis, and is the optimization-based intuition behind the above lemma.

5.3.3 The Coupling Lemma

The main idea in our proof to Lemma 5.6 is to divide the gradient vector $\nabla f(x) \in [-1,\infty)^n$ into three components, the component containing large coordinates (i.e., bigger than 1), the component containing small coordinates (i.e., in $[-1,1] \setminus [-\varepsilon,\varepsilon]$, and the component containing negligible coordinates (i.e., in $[-\varepsilon,\varepsilon]$). The large gradients are to be taken care by the gradient descent lemma, the small gradients are to be taken care by the mirror descent lemma. Formally,

Proof of Lemma 5.6. By convexity, the distance $f_\mu(x^{(k)}) - f_\mu(u)$ for an arbitrary $u \geq 0$ is upper bounded as follows:

$$\begin{aligned}\alpha(f_\mu(x^{(k)}) - f_\mu(u)) &\leq \langle \alpha \nabla f_\mu(x^{(k)}), x^{(k)} - u\rangle \\ &= \langle \alpha\eta^{(k)}, x^{(k)} - u\rangle + \langle \alpha\xi^{(k)}, x^{(k)} - u\rangle + \langle \alpha\zeta^{(k)}, x^{(k)} - u\rangle \ , \quad (5.11)\end{aligned}$$

where

- $\xi_i^{(k)} \stackrel{\text{def}}{=} \mathbb{T}(\nabla_i f_\mu(x^{(k)})) \in [-1,1]$ is the *truncated gradient*, capturing the small coordinates.
- $\eta_i^{(k)} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} \nabla_i f_\mu(x^{(k)}) - \xi_i^{(k)}, & \text{if } \nabla_i f_\mu(x^{(k)}) \geq 1; \\ 0, & \text{otherwise.} \end{array} \right\} \in [0,\infty)$, capturing the large coordinates.
- $\zeta_i^{(k)} \stackrel{\text{def}}{=} \nabla_i f_\mu(x^{(k)}) - \xi_i^{(k)} - \eta_i^{(k)} \in [-\varepsilon,\varepsilon]$, capturing the negligible coordinates.

We analyze the three components of (5.11) one by one.

The ζ component is small: if $f_\mu(u) \leq 0$, we have

$$\langle \alpha\zeta^{(k)}, x^{(k)} - u\rangle \leq \alpha\varepsilon \cdot (\mathbb{1}^T x^{(k)} + \mathbb{1}^T u) \leq \alpha\varepsilon \cdot (1+\varepsilon)\mathsf{OPT} + \alpha\varepsilon \cdot \mathbb{1}^T u \qquad (5.12)$$

where the last inequality is because $f_\mu(x^{(k)}) \leq 0$ from Lemma 5.4.

The η component can be upper bounded with the help from our gradient descent Lemma 5.4. Note that $\eta_i^{(k)}$ only if $i \in B^{(k)}$ (where recall from Lemma 5.4 that $B^{(k)}$ is the set of indices whose $\nabla_i f_\mu(x^{(k)})$ is no less than 1). In particular, if $i \in B^{(k)}$ we have $\eta_i^{(k)} = \nabla_i f_\mu(x^{(k)}) - 1 < \nabla_i f_\mu(x^{(k)})$, and thus Lemma 5.4 gives

$$\begin{aligned}\frac{4(f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}))}{\alpha} &\geq \sum_{i \in B^{(k)}} x_i^{(k)} \cdot \nabla_i f_\mu\big(x^{(k)}\big) \geq \langle \eta^{(k)}, x^{(k)}\rangle \\ \implies \langle \alpha\eta^{(k)}, x^{(k)} - u\rangle &\leq \langle \alpha\eta^{(k)}, x^{(k)}\rangle \leq 4(f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}))\end{aligned}$$

Finally, the ξ component is upper bounded by Lemma 5.5. Together, we obtain

$$\begin{aligned}&\alpha(f_\mu(x^{(k)}) - f_\mu(u)) \leq \langle \alpha\eta^{(k)}, x^{(k)} - u\rangle + \langle \alpha\xi^{(k)}, x^{(k)} - u\rangle + \langle \alpha\zeta^{(k)}, x^{(k)} - u\rangle \\ &\leq 4(f_\mu(x^{(k)}) - f_\mu(x^{(k+1)})) + \alpha^2\mathsf{OPT} + V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u) + \alpha\varepsilon \cdot (1+\varepsilon)\mathsf{OPT} + \alpha\varepsilon\mathbb{1}^T u \\ &\leq 4(f_\mu(x^{(k)}) - f_\mu(x^{(k+1)})) + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u)\big) + \alpha \cdot 2\varepsilon\mathsf{OPT} + \alpha \cdot \varepsilon\mathbb{1}^T u \ . \qquad \square\end{aligned}$$

5.4 Parallelizable Covering LP Solver

Since a primal solution x satisfying $f_\mu(x) \approx -\mathsf{OPT}$ does not translate into a dual solution y of the covering LP (5.2), the results in Section 5.3 do not imply any good approximate to the covering LP program. In fact, most of the previous results (except Luby and Nisan) have encountered this similar problem, and thus needed a separate algorithm to solve the covering LP. We show in this section that, in our same algorithm `PosLPSolver`, once the average $\bar{y} = \sum_{i=0}^{T-1} y(x^{(k)})$ is collected over all the iterations, this $\bar{y}$ is essentially an approximate solution to the covering LP.

The high level intuition behind this result is very clear. On one hand, the packing LP (5.1) is dual to the covering LP (5.2). On the other hand, `PosLPSolver` falls into a primal-dual framework: (a) the (primal) gradient descent ensures that the final objective $f_\mu(x^{(T)})$ is sufficiently small, while (b) the (dual) mirror descent ensures that the average of the encountered gradients (which is a function on $\bar{y}$) is sufficiently close to 0. If (a) gives rise to an approximate solution to the packing LP, then (b) should, at least intuitively, give rise to a dual solution $\bar{y}$ of the covering LP.

More formally, after telescoping Lemma 5.6 for $k = 0, 1, \ldots, T-1$, we have for any $u \geq 0$,

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}\langle \nabla f_\mu(x^{(k)}), x^{(k)} - u\rangle \\
\leq\ & \frac{4}{\alpha T}(f_\mu(x^{(0)}) - f_\mu(x^{(T)})) + \frac{1}{\alpha T}\big(V_{x^{(0)}}(u) - V_{x^{(T)}}(u)\big) + 2\varepsilon\mathsf{OPT} + \varepsilon \mathbb{1}^T u \\
\leq\ & \frac{4}{\alpha T}(f_\mu(x^{(0)}) - f_\mu(x^{(T)})) + \frac{1}{\alpha T}V_{x^{(0)}}(u) + 2\varepsilon\mathsf{OPT} + \varepsilon \mathbb{1}^T u \ . \qquad (5.13)
\end{aligned}$$

This upper bound (on the average regret) gives a lot of information about the average gradient $\frac{1}{T}\sum_k \nabla f_\mu(x^{(k)})$, thanks to the arbitrary choice of $u \geq 0$. For instance, if most of the terms in (5.13) were zero and we had $\frac{1}{T}\sum_{k=0}^{T-1}\langle \nabla f_\mu(x^{(k)}), -u\rangle \leq 0$, we would have $\frac{1}{T}\sum_k \nabla f_\mu(x^{(k)}) \geq 0$, which is equivalent to $A^T\bar{y} \geq \mathbb{1}$, the feasibility of the covering LP. However, since there are five missing terms in this wishful example, more careful studies are needed.

It is worth noting that the average $\bar{y}$ only provides a $(1 + O(\varepsilon))$ approximation to the covering LP when $T \geq \Omega(\frac{\log(n\rho)\log(nm/\varepsilon)}{\varepsilon^3})$, where ρ is the width of A. This is slightly worse than the T required in Algorithm 2, because $\log(n\rho)$ may in principle be slightly larger than $\log(n)$. We prove, however, if one is willing to perform a linear time coordinate fixing on the output $\bar{y}$, then the same number of iterations from Algorithm 2 is sufficient. This result requires a more careful choice of $u \geq 0$ in the above reasoning.

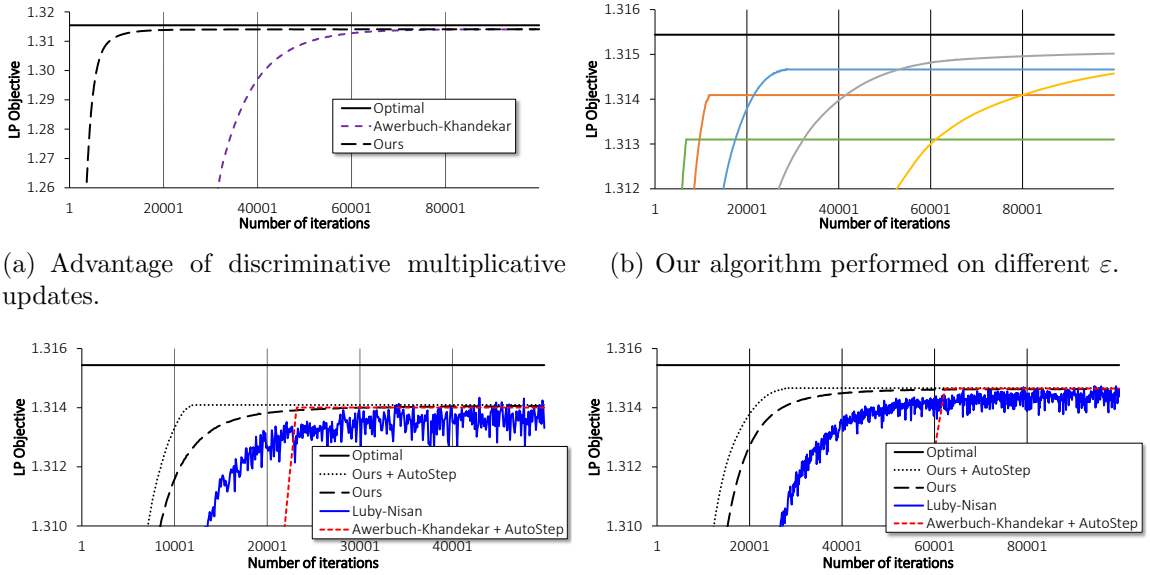
We defer all the technical details on the covering LP including the formal statement of our theorem (see Theorem 5.13 on page 142) to Appendix 5.D.

Acknowledgements

We thank Jonathan Kelner, Yin Tat Lee, Richard Peng, and Neal Young for helpful conversations. This material is based upon work partly supported by the National Science Foundation under Grant CCF-1319460 and by a Simons Graduate Student Award under grant no. 284059.

Appendix

5.A Empirical Evaluation



(a) Advantage of discriminative multiplicative updates.

(b) Our algorithm performed on different ε.

(c) Performance comparison on some small ε

(d) Performance comparison on even smaller ε

Figure 5-1: Empirical Evaluations

5.A.1 AutoStep: Automatic Step-Length Computation

We begin this section by describing an implementation trick that can be applied to both our algorithm and Awerbuch and Khandekar [17]. Recall from (5.7) that, we have chosen our α in `PosLPSolver` to be the (theoretically) most aggressive value such that $\nabla_i f_\mu(x)$ is not going to be affected multiplicatively by more than 1 ± 0.5. In practice, however, this maximal step length α can be computed numerically during each iteration, and can be made different among iterations.[10] This automatic step-length computation can also be applied to Awerbuch and Khandekar [17], and has

[10]It is even true that our theorems can be adapted to allow different α's to be used, however, we have chosen not to do so for the simplicity of our theoretical results.

already been implicitly applied to all other previous algorithms.[11]

5.A.2 Illustration

We perform some simple experiments to illustrate the performance of our new algorithm with real data. We focus on the packing LP program with a randomly generated matrix $A \in \mathbb{R}^{60\times 40}$ of 800 non-zero entries each in the range of $[0, 10]$, whose optimal value $\mathsf{OPT} = 1.31544$. We have implemented the following five algorithms.

- Luby and Nisan [101].
- Awerbuch and Khandekar [17], with and without the AutoStep trick.
- Our `PosLPSolver`, with and without the AutoStep trick.

Importance of Discriminative Updates. We compare the solver of Awerbuch and Khandekar with ours, to illustrate the importance of using *discriminative* multiplicative updates. (Recall that the algorithm of Awerbuch and Khandekar [17] is very similar to ours, except that they update all the relevant coordinates by the same factor, while we treat them differently and update a coordinate x_i more slowly if its feedback v_i is small.) Figure 5-1(a) clearly confirms that this discrimination is very important.

Role of the Smoothed Objective. Notice that, for our algorithm `PosLPSolver`, when the input parameter ε varies, the performance curves go *across* each other (see Figure 5-1(b)). To be clear, with larger ε the curve goes up faster but converges to a worse solution (see the bottommost green curve); while on the other hand, with smaller ε the curve goes up slower but has the potential to converge to a better solution (see the rightmost orange curve). This is because, for different values of ε, our smoothed objective $f_\mu(x)$ has its parameter μ dependent on ε, and therefore the minimum points of $f_\mu(x)$ will have different distances to the actual LP optimum.

(This behavior is in fact shared with all other methods as well.[12] Therefore, to conduct a fair experiment when comparing different algorithms in the next paragraph, we tune the input parameters —via binary search— on each algorithm separately, so as to make sure that they converge to the same value. Then, we plot the curves corresponding to these input parameters.)

Performance Comparison. We illustrate the performance difference between Luby-Nisan, our `PosLPSolver` (with and without AutoStep), and Awerbuch-Khandekar

[11] Other algorithms —namely, [101, 25, 165]— have implemented this automatic step-length computation for a different purpose: they need it in their convergence analysis but we do not. This is one of the reasons our algorithm `PosLPSolver` is much simpler than theirs. (In their algorithms, the convergence analysis is quite combinatorial and works essentially as follows. In each iteration, because the update rule is maximally aggressive, at least one of the inner products $\langle A_i, x\rangle$ is going to be increased by a fixed additive amount. However, this increment cannot happen too many times because otherwise at least one of the constraints will be violated.)

[12] All known methods are implicitly 'smoothing' the LP objective by some parameter, and then performing the related updates. Therefore, none of our algorithms converge to the LP optimum.

with AutoStep. We have ignored Awerbuch-Khandekar in this comparison due to its poor performance. We have chosen two quite small values of ε in order to clearly see the performance difference between algorithms that have different dependencies on ε. It is clear from Figure 5-1(c) and Figure 5-1(d) that our algorithm outperforms all others, and the practical performance of AutoStep is also considerable. It is worth noting that the solution produced by `PosLPSolver` is much more stable than Luby-Nisan (because we focus on the decreasing of some objective $f_\mu(x)$ while their algorithm is quite combinatorial), and each iteration of ours is at least 5 times faster than theirs due to the simplicity of our algorithm.

5.B Semi-Stateless Feature of our Positive-LP Solver

One typical distributed setting for implementing a parallelizable positive-LP solver is as follows.[13] Suppose that there is an agent i controlling variable x_i, and agent i is assumed to know (1) (upper bounds on) m and n, (2) the i-th column of A, and (3) the current "congestion" $(Ax)_j$ for those constraints j that agent i has non-zero influence (i.e., for those j such that $A_{i,j} > 0$). These are the only information disclosed to agent i.

It is not hard to verify that our `PosLPSolver`(A, ε), like most of the previous results in Table 5.1, can be implemented in this distributed setting in $\frac{\log^2 N}{\varepsilon^3}$ synchronized iterations.

Stateless Algorithms. Recently, distributed algorithms that are *stateless* have received a lot of attention [17, 16, 18, 19]. In the language of positive LPs (see [17]), the stateless requirement says that

> "the decisions made by agents are not dependent on the past;
> they are only dependent on the current local state observable to the agents."

Although their definition is vague, statelessness implies the following three important properties, and therefore to check if an algorithm is stateless, it suffices to verify them one by one.

(P1) *Self-stabilization.* The algorithm is robust against adversarial but finite sequence of "hard reset" events. This allows some agents to fall asleep for a finite period of time, and then to wake up; or equivalently, it means that the algorithm does not need to be initialized.

(P2) *Robustness against incremental adjustments.* Agents are allowed to join or leave dynamically. This corresponds to zeroing out or introducing new columns in A, without restarting other agents. Adding or deleting rows, or even modifications to entries of A are similarly allowed.

[13] We refer interested readers to [17] for the strong motivations and practical examples for such settings.

(P3) *No global clock.* Algorithms can proceed asynchronously without a global clock.

Before Awerbuch and Khandekar [17], all known parallelizable positive-LP solvers are stateful, and do not satisfy any of the three properties above. In particular, the width-independent ones are phaseful and have to inform each agent 'which phase it is in' (and many of them only increase x throughout the process), while the width-dependent ones (such as [131]) must keep track of the maximum violation in a constraint.

Our Semi-Stateless Positive-LP Solver. We wish to point out that our `PosLPSolver` can be easily tuned to at least satisfy (P1) and (P2). However, our current analysis still requires the agents to act synchronously and therefore needs a global clock. We call any algorithm that satisfy (P1) and (P2) *semi-stateless*.[14]

Indeed, the only line we need to change in the algorithm $\texttt{PosLPSolver}(A, \varepsilon)$ is to let

$$x_i^{(k+1)} \leftarrow \max\left\{ x_i^{(k)} \cdot \exp^{-\alpha \cdot \mathbb{T}(v_i)},\ \frac{\delta}{\|A_{\diamond i}\|_\infty} \right\} ,$$

where δ is some small enough number such as $\delta = (\varepsilon/nm)^5$. This small modification was also used in [17] to obtain stateless algorithms, and makes our algorithm robust again arbitrarily chosen input. (For instance, adversarially chosen agents may initialize some coordinate x_i to zero; without the introduction of δ, the value of x_i will freeze at zero since each step is only multiplicative.)

We ignore the formal proof of statelessness in this version of the paper because it is routinary.

5.C Missing Proof of Proposition 5.2

Proposition 5.2.

(a) $\mathsf{OPT} \in [1, n]$.

(b) Letting $x = (1 - \varepsilon/2)x^* \geq 0$, *we have* $f_\mu(x) \leq -(1 - \varepsilon)\mathsf{OPT}$.

(c) Letting $x^{(0)} \geq 0$ *be such that* $x_i^{(0)} = \frac{1-\varepsilon/2}{n\|A_{\diamond i}\|_\infty}$ *for each* $i \in [n]$, *we have* $f_\mu(x^{(0)}) \leq -\frac{1-\varepsilon}{n}$.

(d) For any $x \geq 0$ *satisfying* $f_\mu(x) \leq 0$, *we must have* $Ax \leq (1 + \varepsilon)\mathbb{1}$, *and thus* $\mathbb{1}^T x \leq (1 + \varepsilon)\mathsf{OPT}$.

(e) If $x \geq 0$ *satisfies* $f_\mu(x) \leq -(1 - O(\varepsilon))\mathsf{OPT}$, *then* $\frac{1}{1+\varepsilon}x$ *is a* $(1 - O(\varepsilon))$*-approximate solution for the packing LP.*

(f) The gradient of $f_\mu(x)$ *can be written as*

$$\nabla f_\mu(x) = A^T y(x) - \mathbb{1} \quad \textit{where} \quad y_j(x) \stackrel{\text{def}}{=} \exp^{\frac{1}{\mu}((Ax)_j - 1)} \ .$$

Proof.

[14] Technically speaking, the agents in our algorithm `PosLPSolver` do not have states as well, but do need to use a virtual global state that is the clock.

(a) Suppose that i^* is the column that achieves the smallest infinite norm $\|A_{\diamond i}\|_\infty$ over all columns. Letting x be such that $x_i = 1$ at $i = i^*$ and $x_i = 0$ elsewhere, we have obtained a feasible solution for the packing LP (5.1), owing to our choice of $\min_{i\in[n]}\{\|A_{\diamond i}\|_\infty\} = 1$ in (5.3). This feasible x gives an objective $\mathbb{1}^T x = 1$, showing that $\mathsf{OPT} \geq 1$.

On the other hand, for any solution $x \in \mathbb{R}^n_{\geq 0}$ satisfying $Ax \leq \mathbb{1}$, we must have $x_i \leq \frac{1}{\|A_{\diamond i}\|_\infty}$ for each i. Therefore, $\mathbb{1}^T x \leq \sum_i \frac{1}{\|A_{\diamond i}\|_\infty} \leq n$, showing that $\mathsf{OPT} \leq n$.

(b) We have $\mathbb{1}^T x = (1-\varepsilon/2)\mathsf{OPT}$ by the definition of OPT. Also, from the feasibility constraint $Ax^* \leq \mathbb{1}$ in the packing LP, we have $Ax - \mathbb{1} \leq -\varepsilon/2 \cdot \mathbb{1}$, and can compute $f_\mu(x)$ as follows:

$$\begin{aligned} f_\mu(x) = \mu \sum_j \exp^{\frac{1}{\mu}((Ax)_j - 1)} - \mathbb{1}^T x &\leq \mu \sum_j \exp^{\frac{-\varepsilon/2}{\mu}} - (1-\varepsilon/2)\mathsf{OPT} \\ &\leq \frac{\mu m}{(nm)^2} - (1-\varepsilon/2)\mathsf{OPT} \leq -(1-\varepsilon)\mathsf{OPT} \ . \end{aligned}$$

(c) Using the fact that $Ax^{(0)} - \mathbb{1} \leq -\varepsilon/2 \cdot \mathbb{1}$, we compute $f_\mu(x^{(0)})$ as follows:

$$\begin{aligned} f_\mu(x^{(0)}) &= \mu \sum_j \exp^{\frac{1}{\mu}((Ax^{(0)})_j - 1)} - \mathbb{1}^T x^{(0)} \\ &\leq \mu \sum_j \exp^{\frac{-\varepsilon/2}{\mu}} - \frac{1-\varepsilon/2}{n} \leq \frac{\mu m}{(nm)^2} - \frac{1-\varepsilon/2}{n} \leq -\frac{1-\varepsilon}{n} \ . \end{aligned}$$

Above, we have used that $\mathbb{1}^T x^{(0)} \geq x_i^{(0)} = \frac{1-\varepsilon/2}{n}$, where i is the column such that $\|A_{\diamond i}\|_\infty = 1$.

(d) To show $Ax \leq (1+\varepsilon)\mathbb{1}$, we can assume that $v = \max_j((Ax)_j - 1) \geq 0$ because otherwise we are done. Under this definition, we have $Ax \leq (1+v)\mathbb{1}$ and therefore $\mathbb{1}^T x \leq (1+v)\mathsf{OPT}$ by the definition of OPT. We compute $f_\mu(x)$ as follows.

$$\begin{aligned} f_\mu(x) \geq \mu \exp^{\frac{v}{\mu}} - (1+v)\mathsf{OPT} &\geq \mu((\frac{nm}{\varepsilon})^4)^{v/\varepsilon} - (1+v)n \\ &= \frac{\varepsilon}{4\log(nm/\varepsilon)}((\frac{nm}{\varepsilon})^4)^{v/\varepsilon} - (1+v)n \ . \end{aligned}$$

It is easy to see that the above quantity is positive whenever $v \geq \varepsilon$, and therefore, to satisfy $f_\mu(x) \leq 0$ we must have $v \leq \varepsilon$, which is equivalent to $Ax \leq (1+\varepsilon)\mathbb{1}$.

Finally, we notice that $Ax \leq (1+\varepsilon)\mathbb{1}$ implies $\mathbb{1}^T x \leq (1+\varepsilon)\mathsf{OPT}$ by the definition of OPT.

(e) For any x satisfying $f_\mu(x) \leq -(1-O(\varepsilon))\mathsf{OPT} \leq 0$, owing to Proposition 5.2.d, we first have that x is approximately feasible, i.e., $Ax \leq (1+\varepsilon)\mathbb{1}$. Next,

because $-\mathbb{1}^T x \le f_\mu(x) \le -(1-O(\varepsilon))\mathsf{OPT}$, we know that x yields an objective $\mathbb{1}^T x \ge (1-O(\varepsilon))\mathsf{OPT}$. Letting $x' = \frac{1}{1+\varepsilon}x$, we both have that x' is feasible (i.e., $Ax' \le \mathbb{1}$), and x' has an objective $\mathbb{1}^T x'$ at least as large as $(1-O(\varepsilon))\mathsf{OPT}$.

(f) Straightforward by some simple computation. □

5.D Parallelizable Covering LP Solver

We divide our results on the covering LP into two parts. In the first part (see Section 5.D.1), we show that the objective $\mathbb{1}^T\bar{y}$ is close to OPT; in the second part (see Section 5.D.2), we show that $A^T\bar{y} \ge (1-2\varepsilon)\mathbb{1}$ is approximately feasible. Both of our two steps rely on (5.13).

5.D.1 Objective Optimality

We now show that the covering LP objective $\mathbb{1}^T\bar{y} \le (1+O(\varepsilon))\mathsf{OPT}$ as long as $T \ge \Omega(\frac{\log(nm/\varepsilon)}{\varepsilon^3})$. Note that this is smaller than that of $T \ge \Omega(\frac{\log n \cdot \log(nm/\varepsilon)}{\varepsilon^3})$ required in Theorem 5.7; however, as we shall see, it does not imply a faster convergence rate for covering LP than packing LP, because obtaining the approximate feasibility (i.e., $A^T\bar{y} \ge (1-2\varepsilon)\mathsf{OPT}$) requires more iterations.

The following lemma can be deduced essentially by (1) substituting $u = 0$ into (5.13), and (2) noticing that $\langle \nabla f_\mu(x^{(k)}), x^{(k)}\rangle \approx \mathbb{1}^T y(x^{(k)}) - \mathbb{1}^T x^{(k)}$ is approximately the duality gap at step k.

Lemma 5.11. *For any $T \ge \frac{6}{\alpha\varepsilon} = \Omega(\frac{\log(nm/\varepsilon)}{\varepsilon^3})$, we have that $\mathbb{1}^T\bar{y} \le (1+5\varepsilon)\mathsf{OPT}$.*

Proof. Substituting $u = 0$ into inequality (5.13), and using the fact that $V_{x^{(0)}}(0) = \mathbb{1}^T x^{(0)} \le 1$, we obtain

$$\frac{1}{T}\sum_{k=0}^{T-1}\langle \nabla f_\mu(x^{(k)}), x^{(k)}\rangle \le \frac{4}{\alpha T}(f_\mu(x^{(0)}) - f_\mu(x^{(T)})) + \frac{1}{\alpha T} + 2\varepsilon\mathsf{OPT} \tag{5.14}$$

We now respectively lower and upper bound the two sides of (5.14) as follows. One one hand, using the definition of gradient, the left hand side of (5.14) is lower bounded as

$$\begin{aligned}
\langle \nabla f_\mu(x^{(k)}), x^{(k)}\rangle &= \langle A^T y(x^{(k)}), x^{(k)}\rangle - \mathbb{1}^T x^{(k)} = \langle y(x^{(k)}), Ax^{(k)}\rangle - \mathbb{1}^T x^{(k)} \\
&= \sum_j \exp^{\mu((Ax^{(k)})_j - 1)} \cdot (Ax^{(k)})_j - \mathbb{1}^T x^{(k)} \\
&\ge (1-\varepsilon)\sum_j \exp^{\mu((Ax^{(k)})_j - 1)} - \mathbb{1}^T x^{(k)} - m \cdot (\frac{\varepsilon}{nm})^4 \\
&= (1-\varepsilon)\mathbb{1}^T y(x^{(k)}) - \mathbb{1}^T x^{(k)} - m \cdot (\frac{\varepsilon}{nm})^4 \ .
\end{aligned} \tag{5.15}$$

Here, the (only) inequality is because if $(Ax^{(k)})_j < 1-\varepsilon$ for some constraint $j \in [m]$, the corresponding $\exp^{\mu((Ax^{(k)})_j - 1)} \le \exp^{-\mu/\varepsilon} = (\frac{\varepsilon}{nm})^4$ is very small.

On the other hand, since $Ax^{(T)} \le (1+\varepsilon)\mathbb{1}$ by Proposition 5.2.d, we must have $\mathbb{1}^T x^{(T)} \le (1+\varepsilon)\mathsf{OPT}$, and thus $f_\mu(x^{(T)}) \ge 0-(1+\varepsilon)\mathsf{OPT}$. This gives an upper bound on the right hand side of (5.14) that is $\frac{4(1+\varepsilon)}{\alpha T}\mathsf{OPT} + \frac{1}{\alpha T} + 2\varepsilon\mathsf{OPT} \le 3\varepsilon\mathsf{OPT}$, due to our choice of $T \ge \frac{6}{\alpha\varepsilon}$.

Together, we deduce from (5.14) that

$$(1-\varepsilon)\frac{1}{T}\sum_k \left(\mathbb{1}^T y(x^{(k)}) - \mathbb{1}^T x^{(k)}\right) - m\cdot(\frac{\varepsilon}{nm})^4 \le 3\varepsilon\mathsf{OPT}$$

$$\implies \mathbb{1}^T\left(\frac{1}{T}\sum_k y(x^{(k)})\right) \le \frac{1}{T}\sum_k \mathbb{1}^T x^{(k)} + 4\varepsilon\mathsf{OPT} \le (1+\varepsilon)\mathsf{OPT} + 4\varepsilon\mathsf{OPT}\ ,$$

where the last inequality is from $\mathbb{1}^T x^{(k)} \le (1+\varepsilon)\mathsf{OPT}$ for each k. □

5.D.2 Approximate Feasibility

The approximate feasibility is tricker to prove. Indeed, the first proof to come to one's mind only implies that for $A^T\bar{y} \ge (1-2\varepsilon)\mathbb{1}$ for $T \ge \Omega(\frac{\log(n\rho)\log(nm/\varepsilon)}{\varepsilon^3})$. Here, ρ is the largest entry of A (i.e., the width). This bound on T is slightly weaker than that in Theorem 5.7 because $\log(n\rho)$ may be larger than $\log(n)$. Fortunately, this loss can be avoided thanks to one of the two fixes below:

- Width Reduction Pre-processing. One can modify the positive LPs to ensure $\rho = n^{O(1)}$.[15] However, this modification requires some initialization which, if implemented, would make our algorithm not semi-stateless (see Appendix 5.B).
- Coordinate Fix Post-processing. We prove below that, for the same requirement on $T \ge \Omega(\frac{\log(n)\log(nm/\varepsilon)}{\varepsilon^3})$ as Theorem 5.7, although $A^T\bar{y}$ may be smaller than $1-\varepsilon$ for some coordinate, one can safely raise some coordinates of $\bar{y}$ to obtain $A^T\bar{y}' \ge (1-\varepsilon)\mathbb{1}$, without increasing $\mathbb{1}^T\bar{y}$ too much.

More specifically,

Lemma 5.12. *Let $\rho = \max_{i,j}|A_{i,j}|$, and $\bar{y} = \frac{1}{T}\sum_{k=0}^{T-1} y(x^{(k)})$.*

- *If $T \ge \max\{\frac{6}{\alpha\varepsilon}, \frac{\log(4n^2\rho)}{\alpha\varepsilon}\} = \Omega\big(\frac{\log(n\rho)\log(nm/\varepsilon)}{\varepsilon^3}\big)$, we have $A^T\bar{y} \ge (1-2\varepsilon)\mathbb{1}$.*
- *If $T \ge \frac{6\log(2n)}{\alpha\varepsilon} = \Omega(\frac{\log n\cdot\log(nm/\varepsilon)}{\varepsilon^3})$ (which is the same choice of T in* `PosLPSolver`(A,ε)*), there exists some simple fix $\bar{y}'$ from* `FixCoord`$(A,\varepsilon,\bar{y})$ *(see Algorithm 3) satisfying*

$$A^T\bar{y}' \ge (1-2\varepsilon)\mathbb{1} \qquad \text{and} \qquad \mathbb{1}^T\bar{y}' \le \mathbb{1}^T\bar{y} + \varepsilon\mathsf{OPT}\ .$$

[15] This can be done informally as follows. Within a single column of A, if the largest and smallest entries are off from either other by a factor more than $n^{\Omega(1)}$, the smallest entry can be replaced with zero without sacrificing too much accuracy. With this in mind, we can zero out "small" entries of each column. Next, we can similarly zero out "large" columns across all columns, and re-scale A to get $\rho = n^{O(1)}$.

Algorithm 3 `FixCoord`$(A, \varepsilon, \bar{y})$

Input: $A \in \mathbb{R}^{m\times n}_{\geq 0}$, $\varepsilon \in (0, 1/10]$, and $\bar{y} \in \mathbb{R}^m_{\geq 0}$.
Output: $y \in \mathbb{R}^m_{\geq 0}$ that satisfies $A^T y \geq \mathbb{1}$.

1: $\bar{y}' \leftarrow \bar{y}$.
2: **for all** i such that $\lambda_i \stackrel{\text{def}}{=} (A^T\bar{y})_i - 1 + \varepsilon \leq -\varepsilon$ **do**
3: Let $j \in [m]$ be the largest entry in the i-th column, i.e., $A_{i,j} = \|A_{\diamond i}\|_\infty$.
4: $\bar{y}'_j \leftarrow \bar{y}'_j + \frac{-\lambda_i}{A_{i,j}}$.
5: **end for**
6: **return** $\frac{\bar{y}'}{1-2\varepsilon}$.

The proof of this lemma is involved, but has a clear high level intuition behind it.

We extract from (5.13) out only those terms that have u in it, and rewrite (5.13) as follows: (here we have used the definition of $\nabla f_\mu(x^{(k)}) = A^T y(x^{(k)}) - \mathbb{1}$)

$$0 \leq \star + \frac{1}{\alpha T} V_{x^{(0)}}(u) + \langle A^T\bar{y} - \mathbb{1} + \varepsilon\mathbb{1}, u\rangle \ . \tag{5.16}$$

Now, suppose that $A^T\bar{y} \geq (1-2\varepsilon)\mathbb{1}$ is violated, there must exist some coordinate i such that $(A^T\bar{y} - \mathbb{1} + \varepsilon\mathbb{1})_i < -\varepsilon$ is very negative. In such as case, we let $u_k = 0$ for every $k \neq i$, and use the choice $T \geq \Omega(\frac{\log(n\rho)}{\alpha\varepsilon})$. Inequality (5.16) is then simplified as $0 \leq \star + O(\frac{\varepsilon}{\log(n\rho)}) \cdot (u_i \log u_i - u_i) - \varepsilon \cdot u_i$. However, we can choose $u_i = (n\rho)^{\Omega(1)}$ to be very large, making the right hand side very negative. This contradicts to inequality (5.16), and thus finishes the proof of $A^T\bar{y} \geq (1-2\varepsilon)\mathbb{1}$ for the first half of the lemma.

To obtain the second half, it is first easy to see that `FixCoord`$(A, \varepsilon, \bar{y})$ is computing some $\bar{y}'$ satisfying $A^T\bar{y}' \geq (1-2\varepsilon)\mathbb{1}$, because $\bar{y}'$ is so constructed to fix every violation of $A^T\bar{y} \geq (1-2\varepsilon)\mathbb{1}$. What is much harder to prove is that $\mathbb{1}^T\bar{y}' \approx \mathbb{1}^T\bar{y}$. In fact, this can be obtained, after some careful computation, from (5.16) again. This time, we carefully choose a different u: we identify *all* coordinates i such that $(A^T\bar{y} - \mathbb{1} + \varepsilon\mathbb{1})_i < -\varepsilon$, and let u_i be large on all of them.

Proof of Lemma 5.12. This time, we rewrite (5.13) as

$$\begin{aligned}\frac{1}{T}\sum_{k=0}^{T-1}\langle \nabla f_\mu(x^{(k)}), x^{(k)} - u\rangle &\leq \frac{4}{\alpha T}(f_\mu(x^{(0)}) - f_\mu(x^{(T)})) + \frac{1}{\alpha T}V_{x^{(0)}}(u) + 2\varepsilon\mathsf{OPT} + \varepsilon\mathbb{1}^T u \\ &\leq \frac{1}{\alpha T}V_{x^{(0)}}(u) + 3\varepsilon\mathsf{OPT} + \varepsilon\mathbb{1}^T u\end{aligned}$$

where the last inequality comes from the fact that $\frac{4}{\alpha T}(f_\mu(x^{(0)}) - f_\mu(x^{(T)})) \leq \varepsilon\mathsf{OPT}$, which we have already used once in the proof of Lemma 5.11. Let us define

$$\phi(u) \stackrel{\text{def}}{=} \frac{1}{\alpha T}V_{x^{(0)}}(u) + \frac{1}{T}\sum_{k=0}^{T-1}\langle \nabla f_\mu(x^{(k)}), u - x^{(k)}\rangle + \varepsilon\mathbb{1}^T u$$

and according to the inequality above we have $\phi(u) \geq -3\varepsilon\mathsf{OPT}$ for any $u \geq 0$.

Proof of the First Half of the Lemma. Recall from (5.15) that

$$\langle \nabla f_\mu(x^{(k)}), x^{(k)} \rangle \geq (1-\varepsilon)\mathbb{1}^T y(x^{(k)}) - \mathbb{1}^T x^{(k)} - m\cdot(\frac{\varepsilon}{nm})^4 \geq -\mathbb{1}^T x^{(k)} - m\cdot(\frac{\varepsilon}{nm})^4 \geq -(1+2\varepsilon)\mathsf{OPT}$$

and therefore

$$\frac{1}{\alpha T} V_{x^{(0)}}(u) + \langle A^T\bar{y} - \mathbb{1}, u\rangle = \phi(u) + \langle \nabla f_\mu(x^{(k)}), x^{(k)}\rangle \geq -3\varepsilon\mathsf{OPT} - (1+2\varepsilon)\mathsf{OPT} \geq -(1+5\varepsilon)n \ .$$

If there is some coordinate i^* such that $v \stackrel{\text{def}}{=} (A^T\bar{y})_{i^*} - 1 + \varepsilon \leq -\varepsilon$, we substitute $u = (0, 0, \ldots, x^{(0)}_{i^*} \cdot e^{-\alpha v T}, 0, \ldots, 0)$ where $u_{i^*} = x^{(0)}_{i^*} \cdot e^{-\alpha v T}$ into the above inequality, and we get

$$\frac{1}{\alpha T}\Big(u_{i^*} \log \frac{u_{i^*}}{x^{(0)}_{i^*}} - u_{i^*} + \sum_i x^{(0)}_i\Big) + v \cdot u_{i^*} \geq -(1+5\varepsilon)n \ .$$

Since the left hand side equals to $\frac{1}{\alpha T}\big(-u_{i^*} + \sum_i x^{(0)}_i\big)$ by our choice of u_{i^*}, we immediately obtain $-u_{i^*} \geq -(1+6\varepsilon)n \cdot \alpha T > -2n$ from it. Substituting in the definition of $u_{i^*} = x^{(0)}_{i^*} \cdot e^{-\alpha v T} \geq \frac{1/2}{n\|A_{\diamond i}\|_\infty} \cdot e^{\alpha\varepsilon T}$, we conclude that $T < \frac{\log(4n^2\|A_{\diamond i}\|_\infty)}{\alpha\varepsilon}$. However, this contradicts to our choice of $T \geq \frac{\log(4n^2\rho)}{\alpha\varepsilon}$. In other words, for $T \geq \max\{\frac{6}{\alpha\varepsilon}, \frac{\log(4n^2\rho)}{\alpha\varepsilon}\}$, we must have $(A^T\bar{y})_i - 1 + \varepsilon > -\varepsilon$ for all i, finishing the proof of $A^T\bar{y} \geq (1-2\varepsilon)\mathbb{1}$.

Proof of the Second Half of the Lemma. This time, using the definition of $\phi(u)$ and the convexity of $f_\mu(x)$, we obtain

$$-3\varepsilon\mathsf{OPT} \leq \phi(u) \leq \frac{1}{\alpha T} V_{x^{(0)}}(u) + \frac{1}{T}\sum_{k=0}^{T-1}(f_\mu(u) - f_\mu(x^{(k)})) \ .$$

From now on let us denote by $\tilde{u} \stackrel{\text{def}}{=} (1-\varepsilon/2)x^*$. Recall that our earlier analysis yields the following:

- $f_\mu(\tilde{u}) \leq -(1-\varepsilon)\mathsf{OPT}$ owing to Proposition 5.2.b;
- $f_\mu(x^{(k)}) \geq -(1+\varepsilon)\mathsf{OPT}$, owing to Proposition 5.2.d and $\mathbb{1}^T x^{(k)} \leq (1+\varepsilon)\mathsf{OPT}$; and
- $V_{x^{(0)}}(\tilde{u}) \leq 2\mathsf{OPT} \cdot \log(2n)$, owing to (5.6).

Together, we obtain that

$$-3\varepsilon\mathsf{OPT} \leq \min_{u\geq 0}\phi(u) \leq \phi(\tilde{u}) \leq \frac{1}{\alpha T} V_{x^{(0)}}(\tilde{u}) + 2\varepsilon\mathsf{OPT} \leq 3\varepsilon\mathsf{OPT} \ . \tag{5.17}$$

where the last inequality is from our choice of $T \geq \frac{6\log(2n)}{\alpha\varepsilon}$.

Next we decompose $\phi(u)$ as follows. We let $\phi(u) = \sum_i \phi^i(u_i) + \phi^0$, where

$$\phi^i(u_i) \stackrel{\text{def}}{=} \frac{1}{\alpha T}\Big(u_i \log \frac{u_i}{x^{(0)}_i} + x^{(0)}_i - u_i\Big) + ((A^T\bar{y})_i - 1 + \varepsilon)\cdot u_i \qquad \text{and} \qquad \phi^0 \stackrel{\text{def}}{=} \frac{1}{T}\sum_{k=0}^{T-1}\langle \nabla f_\mu(x^{(k)}), -x^{(k)}\rangle$$

Let us denote by $\lambda \stackrel{\text{def}}{=} A^T\bar{y} - \mathbb{1} + \varepsilon\mathbb{1}$. Then, for each i such that $\lambda_i \leq -\varepsilon$, we make the choice $u^*_i \stackrel{\text{def}}{=} x^{(0)} \cdot e^{-\alpha\lambda_i T}$; otherwise we choose $u^*_i = \tilde{u}_i$.

Focusing on each i such that $\lambda_i \le -\varepsilon$, we have $\phi^i(u_i^*) = \frac{1}{\alpha T}(x_i^{(0)} - u_i^*)$ and $\phi^i(\tilde{u}_i) \ge \lambda_i \tilde{u}_i$. This gives a lower bound on their difference

$$\phi^i(\tilde{u}_i) - \phi^i(u_i^*) \ge \frac{1}{\alpha T}(u_i^* - x_i^{(0)}) + \lambda_i \tilde{u}_i \ .$$

Before continuing to prettify the right hand side, we make a technical observation. Letting $T_0 \stackrel{\text{def}}{=} \frac{6\log(2n)}{\alpha\varepsilon}$ so that $T \ge T_0$, we have

$$\begin{aligned} u_i^* = x^{(0)} \cdot e^{-\alpha\lambda_i T} &\ge \frac{1}{2n\|A_{\diamond i}\|_\infty} \cdot \Big((e^{\alpha\varepsilon T_0})^{T/T_0}\Big)^{-\lambda_i/\varepsilon} \ge \frac{1}{\|A_{\diamond i}\|_\infty}\Big(\big(\frac{1}{2n} \cdot e^{\alpha\varepsilon T_0}\big)^{T/T_0}\Big)^{-\lambda_i/\varepsilon} \\ &\ge \frac{1}{\|A_{\diamond i}\|_\infty}\Big((100n)^{T/T_0}\Big)^{-\lambda_i/\varepsilon} \ . \end{aligned} \tag{5.18}$$

Therefore, the lower bound on $\phi^i(\tilde{u}_i) - \phi^i(u_i^*)$ can be simplified as

$$\begin{aligned} \phi^i(\tilde{u}_i) - \phi^i(u_i^*) &\stackrel{①}{\ge} \frac{1}{\alpha T} u_i^* + \lambda_i \tilde{u}_i - \frac{\varepsilon}{\|A_{\diamond i}\|_\infty} \\ &\stackrel{②}{\ge} \frac{1}{\alpha T}\frac{1}{\|A_{\diamond i}\|_\infty}\Big((100n)^{T/T_0}\Big)^{-\lambda_i/\varepsilon} + \lambda_i \tilde{u}_i - \frac{\varepsilon}{\|A_{\diamond i}\|_\infty} \\ &\stackrel{③}{\ge} \frac{1}{\alpha T}\frac{1}{\|A_{\diamond i}\|_\infty}\Big((100n)^{T/T_0}\Big)^{-\lambda_i/\varepsilon} + \frac{2\lambda_i}{\|A_{\diamond i}\|_\infty} \\ &\stackrel{④}{\ge} \frac{1}{\alpha T_0}\frac{1}{\|A_{\diamond i}\|_\infty}(100n)^{-\lambda_i/\varepsilon} + \frac{2\lambda_i}{\|A_{\diamond i}\|_\infty} \\ &\stackrel{⑤}{\ge} \frac{1}{\alpha T_0}\frac{1}{\|A_{\diamond i}\|_\infty}(100n)\frac{-\lambda_i}{\varepsilon} + \frac{2\lambda_i}{\|A_{\diamond i}\|_\infty} \\ &\stackrel{⑥}{\ge} \frac{-10\lambda_i}{\|A_{\diamond i}\|_\infty} + \frac{2\lambda_i}{\|A_{\diamond i}\|_\infty} \ge \frac{-8\lambda_i}{\|A_{\diamond i}\|_\infty} \ . \end{aligned}$$

Here ① is using the fact that $\frac{1}{\alpha T}x_i^{(0)} \le \varepsilon \cdot \frac{1}{n\|A_{\diamond i}\|_\infty}$. ② is using (5.18). ③ is using the fact that $\tilde{u}_i \le \frac{1}{\|A_{\diamond i}\|_\infty}$ (due to the feasibility $Au \le \mathbb{1}$) and $\lambda_i \le -\varepsilon$. ④ is obtained by realizing that the left hand side of ④ is minimized, over all possible $T \ge T_0$, at $T = T_0$. ⑤ is obtained by realizing that $(100n)^t \ge (100n)t$ for any $t \ge 1$. ⑥ is by the definition of $T_0 = \frac{6\log(2n)}{\alpha\varepsilon}$.

Finally, we combine this with (5.17) and get

$$\sum_{i\,:\,\lambda_i \le -\varepsilon} \frac{-8\lambda_i}{\|A_{\diamond i}\|_\infty} \le \sum_{i\,:\,\lambda_i \le -\varepsilon} \phi^i(\tilde{u}_i) - \phi^i(u_i^*) = \sum_{i\in[n]} \phi^i(\tilde{u}_i) - \phi^i(u_i^*) \le \phi(\tilde{u}) - \min_{u \ge 0} \phi(u) \le 6\varepsilon\mathsf{OPT}$$

and therefore

$$\sum_{i\,:\,\lambda_i \le -\varepsilon} \frac{-\lambda_i}{\|A_{\diamond i}\|_\infty} < \varepsilon\mathsf{OPT} \ . \tag{5.19}$$

Now we come to the last step of the lemma. For each coordinate i such that $\lambda_i = (A^T\bar{y})_i - 1 + \varepsilon \le -\varepsilon$, we find the corresponding j where $A_{i,j} = \|A_{\diamond i}\|_\infty$, and push $\bar{y}_j$ up by an additive amount of $\frac{-\lambda_i}{A_{i,j}}$. Letting $\bar{y}'$ be this new vector, we automatically have that $A^T\bar{y}' \ge (1-2\varepsilon)\mathbb{1}$, and moreover, $\mathbb{1}^T\bar{y}' - \mathbb{1}^T\bar{y} \le \varepsilon\mathsf{OPT}$ due to (5.19). □

It is now easy to see that Lemma 5.11 and Lemma 5.12 together imply that

Theorem 5.13 (Covering LP)**.** *For any* $T \geq \max\{\frac{6}{\alpha\varepsilon}, \frac{\log(4n^2\rho)}{\alpha\varepsilon}\} = \Omega\big(\frac{\log(n\rho)\log(nm/\varepsilon)}{\varepsilon^3}\big)$*, we have that* $\frac{\bar{y}}{1-2\varepsilon}$ *is a* $(1+O(\varepsilon))$*-approximate solution for the covering LP (5.2).*

Alternatively, for any $T \geq \frac{6\log(2n)}{\alpha\varepsilon} = \Omega(\frac{\log n \cdot \log(nm/\varepsilon)}{\varepsilon^3})$*, letting*

$$(x, \bar{y}) = \texttt{PosLPSolver}(A, \varepsilon) \quad \textit{and} \quad y = \texttt{FixCoord}(A, \varepsilon, \bar{y}) \ ,$$

we have that y *is a* $(1+O(\varepsilon))$*-approximate solution for the covering LP (5.2).*

Chapter 6

Nearly-Linear Time Positive LP Solver with Faster Convergence Rate

This chapter is based on the result published in [6], and its further edits can be found at:

http://arxiv.org/abs/1411.1124.

Positive linear programs (LP), also known as packing and covering linear programs, are an important class of problems that bridges computer science, operation research, and optimization. Efficient algorithms for solving such LPs have received significant attention in the past 20 years [101, 131, 24, 165, 113, 32, 25, 118, 47, 17, 115, 10, 92, 166, 7]. Unfortunately, all known nearly-linear time algorithms for producing $(1+\varepsilon)$-approximate solutions to positive LPs have a running time dependence that is at least proportional to ε^{-2}. This is also known as an $O(1/\sqrt{T})$ convergence rate and is particularly poor in many applications.

In this paper, we leverage insights from optimization theory to break this long-standing barrier. Our algorithms solve the packing LP in time $\tilde{O}(N\varepsilon^{-1})$ and the covering LP in time $\tilde{O}(N\varepsilon^{-1.5})$. At high level, they can be described as linear couplings of several first-order descent steps. This is the first application of our linear coupling technique (see [5] or Chapter 4) to problems that are not amenable to blackbox applications known iterative algorithms in convex optimization. Our work also introduces a sequence of new techniques, including the stochastic and the non-symmetric execution of gradient truncation operations, which may be of independent interest.

6.1 Introduction

A generic packing linear program (LP) takes the form $\max\{c^T x \,:\, Ax \le b\}$ where $c \in \mathbb{R}^n_{\ge 0}$, $b \in \mathbb{R}^m_{\ge 0}$, and $A \in \mathbb{R}^{m\times n}_{\ge 0}$; similarly, a generic covering LP can be written as

$\min\{b^T y : A^T y \geq c\}$, with the same requirements on A, b, and c. We denote by N the number of non-zero elements in matrix A. They are also known as positive LPs as originally studied by Luby and Nisan [101].

Similar to Chapter 5, we assume without loss of generality that the LP is in its *standard form*: $b = \mathbb{1}$ and $c = \mathbb{1}$.

$$\begin{aligned} \text{Packing LP:} \quad & \max_{x \geq 0}\{\mathbb{1}^T x : Ax \leq \mathbb{1}\} \ , \\ \text{Covering LP:} \quad & \min_{y \geq 0}\{\mathbb{1}^T y : A^T y \geq \mathbb{1}\} \ . \end{aligned}$$

Since the two programs are dual to each other, we denote by OPT their shared optimal value. We say that x is a $(1-\varepsilon)$-approximation for the packing LP if $Ax \leq \mathbb{1}$ and $\mathbb{1}^T x \geq (1-\varepsilon)\mathsf{OPT}$, and y a $(1+\varepsilon)$-approximation for the covering LP if $A^T y \geq \mathbb{1}$ and $\mathbb{1}^T y \leq (1+\varepsilon)\mathsf{OPT}$.

Of course, it is possible to adopt the general Interior Point or Ellipsoid Methods to obtain approximate solvers with a $\log(1/\varepsilon)$ dependence on the number of iterations. However, the computational cost of such algorithms is typically very high, as each iteration requires the solution of a system of linear equations in $A^T A$. As a consequence, this approach is simply not suitable to the solution of large-scale problems. To address this issue, researchers have developed *iterative approximate* solvers that achieve a better dependence on the problem size (e.g., nearly-linear time N) at the cost of having a $\mathsf{poly}(1/\varepsilon)$ dependence on the approximation parameter ε.

Fast approximate packing and covering LP solvers have been widely used in approximation algorithms (e.g., MinSetCover [101], MaxSet, MaxDiCut, Max-k-CSP [158], bipartite matching), probabilistic checkable proofs [158], zero-sum matrix games [118], scheduling [131], graph embedding [131], flow controls [24, 25], auction mechanisms [169], wireless sensor networks [38], and many other areas. In addition, techniques developed in this line of research have also inspired many other important results, most notably regarding fast algorithms for multi-commodity flow problems [131, 63, 68, 103, 20].

Previous approximate solvers can be further divided into two classes (see Table 6.1).

Width-Dependent Solvers. These algorithms[1] require a running time that is at least N multiplied with $\rho \cdot \mathsf{OPT}$, where ρ is the largest entry, i.e. the *width*, of matrix A. Since $\mathsf{OPT} \geq 1/\rho$, this value $\rho \cdot \mathsf{OPT}$ is at least 1. However, since OPT can easily be as large as 1 or even more than n, this resulting running time is not polynomial, but only pseudo-polynomial. More precisely, packing and covering LPs can be solved in $O(\frac{N\rho^2\mathsf{OPT}^2 \log m}{\varepsilon^2})$ time [131], or $O(\frac{N\rho\mathsf{OPT} \log m}{\varepsilon^2})$ time using negative-width techniques [10]. These algorithms strongly rely on multiplicative weight updates and only require "oracle-access" to the matrix A.

When A is given explicitly like in this paper, the number of iterations can be

[1]Note that most width-dependent solvers are studied under the minmax form of positive LPs, whose optimal value equals $1/\mathsf{OPT}$. Their approximation guarantees are often written in terms of *additive* error. We have translated their performances to multiplicative error for a fair comparison.

Paper	Running Time	Width Independent?
[131]	$O(N \times \frac{\rho^2 \mathsf{OPT}^2 \log m}{\varepsilon^2})$	no
[10]	$O(N \times \frac{\rho \mathsf{OPT} \log m}{\varepsilon^2})$	no
[118, 113]	$O(N \times \frac{\rho \mathsf{OPT} \log m}{\varepsilon})$	no
[32]	$O(N \times \frac{\sqrt{Kn} \log m}{\varepsilon})$	no
[115, 47]: packing LP	$\tilde{O}(N \times (n + \frac{\sqrt{n}}{\varepsilon}))$	no
[101, 24, 165, 25, 17, 7]	$O(N \times \frac{\log^2 N}{\varepsilon^3})$ at best	yes
[165]	$O((md + N) \times \frac{\log N}{\varepsilon^2})$ [a]	yes
[24, 25]	$O(nm \times \frac{\log N}{\varepsilon^2})$	yes
[166]	$O(N \times \frac{\log N}{\varepsilon^2})$	yes
[92]	$O(N + (n+m) \times \frac{\log N}{\varepsilon^2})$	yes
[this paper]: packing LP	$O(N \times \frac{\log N \log \varepsilon^{-1}}{\boldsymbol{\varepsilon}})$	yes
[this paper]: covering LP	$O(N \times \frac{\log N \log \varepsilon^{-1}}{\boldsymbol{\varepsilon}^{1.5}})$	yes

Table 6.1: Comparisons among iterative approximate solvers for packing and covering LPs.

[a] d is the maximum number of constraints each variable is in; md may be larger than N.

reduced to $O(\frac{\rho \mathsf{OPT} \log m}{\varepsilon})$ by deploying more advanced optimization tools such as Nesterov's accelerated gradient method [118], or Nemirovski's mirror prox method [113]. Bienstock and Iyengar [32] have converted this dependence on $\rho\mathsf{OPT}$ into a more benign, yet linear dependence on n. More specifically, their running time is $O(\varepsilon^{-1} N \sqrt{Kn \log m})$ where K is the maximum number of non-zeros per row of A. This is $O(\varepsilon^{-1} Nn \sqrt{\log m})$ in the worst case. The results of [115, 47] have improved this convergence rate (for packing LP only) to $\tilde{O}(\varepsilon^{-1} N \sqrt{n})$, but at a cost of enduring an $\tilde{O}(Nn)$-time preprocessing stage.

Width-Independent Solvers. In this paper, we are interested in a second, more efficient class of methods, i.e. *width-independent*,[2] truly polynomial-time approximate solvers (see Table 6.1).

[2] Some of these solvers may still have a $\mathsf{polylog}(\rho)$ dependence. Since each occurrence of $\log(\rho)$ can typically be replaced with $\log(nm)$ after slightly modifying the instance matrix A, we have done so in Table 6.1 for a fair comparisons.

This line of research was initiated by a seminal paper of Luby and Nisan [101], who gave an algorithm running in $O\big(\frac{N \log^2 N}{\varepsilon^4}\big)$ time with no dependance on ρ. This is the first *nearly-linear-time* approximate solver for solving packing and covering LPs, and also the first to run in parallel in nearly-linear-work and polylogarithmic depth.

The parallel algorithm of Luby and Nisan was extended by a sequence of works [24, 165, 17, 7]. Most notably, the algorithm of the same authors of this paper [7] (see Chapter 5) runs in $O(\frac{\log^2 N}{\varepsilon^3})$ iterations, each costing a matrix-vector multiplication operation that can be implemented in $O(N)$ total work and logarithmic depth.

The ideas of Luby and Nisan also led to sequential width-independent solvers for packing and covering LPs [165, 25, 166, 92]. Most notably, the algorithm of Koufogiannakis and Young [92] runs in time $O\big(N + \frac{\log N}{\varepsilon^2} \times (n+m)\big)$. Despite the amount of work in this area, the $O(1/\varepsilon^2)$ convergence rate has not been improved since 1997. On a separate note, Klein and Young [90] have shown that essentially any Dantzig-Wolfe type algorithm has to pay for a $O(1/\varepsilon^2)$ convergence rate. This lack of progress constitutes a significant limitation, as the ε^{-2}-dependence on the approximation parameter ε is particularly pour. This ε^{-2} dependence is also known as the $O(1/\sqrt{T})$ convergence rate in the optimization language, because the error decreases only at the rate $\varepsilon \propto 1/\sqrt{T}$.

6.1.1 Our Results

Packing LP Solver. We present an algorithm `PacLPSolver` that can be implemented to run in $O(\frac{\log(nm/\varepsilon)\log(1/\varepsilon)}{\varepsilon} N)$ total time. This gives the first nearly-linear time solver for packing LP whose running time has an ε^{-1}-dependence; this running time is also known as the $\tilde{O}(1/T)$ convergence rate in the optimization literature. No nearly-linear time algorithm has achieved any convergence rate that is faster than $O(1/\sqrt{T})$ before our work (see Table 6.1).

Interestingly, the maximum (weighted) bipartite matching is just one instance of a packing LP. Therefore, our algorithm yields an $\tilde{O}(m\varepsilon^{-1})$ approximate algorithm and an $\tilde{O}(m\sqrt{n})$ exact algorithm[3] that arise purely from optimization for bipartite matching, without the use of any dynamic trees. This matches the best known combinatorial algorithms for maximum weighted bipartite matching. Any further improvement over the dependence on ε^{-1} would result in a maximum matching algorithm that runs in time $m \cdot \tilde{o}(\sqrt{n})$, which may require very significantly different ideas.

Our algorithms optimizes a relaxation of the original packing LP, where the hard constraint $Ax \leq 1$ is replaced by an exponential penalty function for violating the constraint. In other words, we reduce the problem of approximately solving packing LP into approximately minimizing some function $f_\mu(x)$ over the positive orthant $x \geq 0$ — see (6.3). This interpretation of the solution of packing and covering linear programs

[3]It is not hard to turn an $\tilde{O}(m\varepsilon^{-1})$ approximate algorithm into an $\tilde{O}(m\sqrt{n})$ algorithm, see for instance [54].

was recently suggested by the same authors of this paper [7] (see Chapter 5). However, the techniques in our previous work [7] only lead to very slow sequential solvers (see Table 6.1). Furthermore, to the best of our knowledge, our objective $f_\mu(x)$ cannot be turned into any class of smooth functions, and therefore traditional accelerated gradient methods such as [116, 118] no longer apply. We thus need fundamentally new ideas.

Our proposed algorithm is an iterative first-order method, and has a flavor of "stochastic coordinate descent" (cf. [145, 61]). Suppose that we are given point $x \geq 0$ at some iteration, and observe the gradient $\nabla f(x) \in [-1, \infty)^n$. Then, we randomly pick a coordinate $i \in [m]$, and focus only on the coordinate gradient $\nabla_i f(x) \in [-1, \infty)$. (In fact, we do not even need to compute $\nabla_\ell f(x)$ for $\ell \neq i$, thus ensuring that each iteration can be implemented very efficiently.)

We divide $\nabla_i f(x) = \eta + \xi$, where $\eta \in [0, \infty)$ is the large component, and $\xi \in [-1, 1]$ is the small (and truncated) component. This *gradient-truncation technique* was developed in our prior work [7], but has never been applied to coordinate gradient.

We perform essentially three coordinate descent steps.

- A *gradient (descent) step* with respect to η, guaranteeing a large decrement on the objective.
- A *mirror (descent) step* and a *gradient (descent) step*, both with respect to ξ.

Both gradient and mirror descent are well-known tools from optimization (see for instance [117, 27]).[4] Motivated by the linear coupling technique developed in [5] (see Chapter 4), we combine the analysis of the above three descent steps for a faster algorithm.

To push through the idea sketched above, we also develop two independent techniques. The *redundant-constraint technique* imposes an additional box constraint; it requires each x_i to be upper bounded by a carefully chosen constant c_i. While this constraint $x_i \leq c_i$ is provably redundant from the viewpoint of minimizing $f_\mu(x)$, it is surprisingly crucial for our linear coupling to work. Our *gradient-mirror scaling* technique restricts our attention to a special type of gradient step, which is always a constant factor of the mirror step. Our two techniques together play an important role in enabling the three descent steps mentioned above to be effectively coupled.

Covering LP Solver. Unlike our most relevant prior work [7], it is not clear how one can extract an (approximate) covering LP solution from the packing LP solver mentioned above. There are at least two main issues behind this difficulty. Firstly, the dual guarantee naturally arising from `PacLPSolver` is on the history of the *full* gradients $\nabla f(x_k)$, rather than the randomly selected coordinate gradients $\nabla_i f(x_k)$,

[4] It is important to note here that we have generalized the notion of "gradient descent" to indicate any descent step that is guaranteed to decrease the objective. This is in contrast to mirror descent, which is a "dual approach" that does not necessarily decrease the objective at any iteration, but minimizes the so-called regularized regret.

over all iterations k. As we mentioned earlier, it is computationally heavy to compute full gradients. Secondly, even if the dual guarantee is on the coordinate gradients $\nabla_i f(x_k)$, it is not clear how one can compute them efficiently in only nearly-linear time.

We therefore are forced to design a new algorithm `CovLPSolver` that works directly for covering LP. On one hand, this new algorithm relies on similar idea that are present in `PacLPSolver`: the linear coupling of gradient and mirror steps and the gradient truncation. On the other hand, we need a different version of the redundant-constraint technique (over a simplex constraint), as well as a negative-width technique.

Our `CovLPSolver` can be implemented to run in $O(\frac{\log(nm/\varepsilon)\log(1/\varepsilon)}{\varepsilon^{1.5}}N)$ total time. This gives the first nearly-linear time solver for covering LP whose running time has a faster dependence than ε^{-2} (or equivalently, the first one whose convergence rate is faster than $\tilde{O}(1/\sqrt{T})$).

6.1.2 Roadmap

We transfer the packing LP problem into a convex optimization question in Section 6.2, and provide our packing LP solver in Section 6.3. We sketch the main ideas needed for our covering LP solver in Section 6.4, and defer the technical details to Section 6.5 and Section 6.6. Note that our `PacLPSolver` and `CovLPSolver` are stated in an implicit optimization language, and their (efficient) implementation details will be addressed in Appendix 6.E and Appendix 6.F.

6.2 Relaxation of the Packing Linear Program

Recall that, for input matrix $A \in \mathbb{R}^{m\times n}_{\geq 0}$, the packing LP in its standard form is $\max_{x\geq 0}\{\mathbb{1}^T x \ : \ Ax \leq \mathbb{1}\}$. Let us denote by OPT the optimal value of this linear program, and x^* any optimal solution. We say that x is a $(1-\varepsilon)$-approximation for the packing LP if $Ax \leq \mathbb{1}$ and $\mathbb{1}^T x \geq (1-\varepsilon)\mathsf{OPT}$.

Throughout this paper, we use the indices $i \in [n]$ to denote the columns of A, and the indices $j \in [m]$ to denote the rows of A. We let $A_{\diamond i}$ be the i-th column vector of A, and $A_{j\diamond}$ the j-th row vector of A. Given any vector x, we denote by $\|x\|_A = \sqrt{\sum_{i\in[n]} x_i^2 \cdot \|A_{\diamond i}\|_\infty}$ the A-norm of x.

By scaling the matrix A and the optimum value, we can assume without loss of generality that

$$\min_{i\in[n]}\{\|A_{\diamond i}\|_\infty\} = 1 \ . \tag{6.1}$$

We can now restrict the range of values x and OPT can take using the following simple fact.

Fact 6.1. *Define the bounding box* $\Delta \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \ : \ x_i \in \big[0, \frac{1}{\|A_{\diamond i}\|_\infty}\big]\}$. *Under assumption* (6.1)*, we have* $\mathsf{OPT} \in [1, n]$ *and* $\{x : x \geq 0 \wedge Ax \leq \mathbb{1}\} \subseteq \Delta$.

Proof. Suppose that i^* is the column that achieves the smallest infinite norm $\|A_{\diamond i}\|_\infty$ over all columns. Letting x be such that $x_i = 1$ at $i = i^*$ and $x_i = 0$ elsewhere, we have obtained a feasible solution for the packing LP (5.1), owing to our choice of $\min_{i\in[n]}\{\|A_{\diamond i}\|_\infty\} = 1$ in (6.1). This feasible x gives an objective $\mathbb{1}^T x = 1$, showing that $\mathsf{OPT} \geq 1$.

On the other hand, for any solution $x \in \mathbb{R}^n_{\geq 0}$ satisfying $Ax \leq \mathbb{1}$, we must have $x_i \leq \frac{1}{\|A_{\diamond i}\|_\infty}$ for each i. Therefore, $\mathbb{1}^T x \leq \sum_i \frac{1}{\|A_{\diamond i}\|_\infty} \leq n$, showing that $\mathsf{OPT} \leq n$.

The inclusion $\{x : x \geq 0 \wedge Ax \leq \mathbb{1}\} \subseteq \Delta$ is obvious, since if $x_i > \frac{1}{\|A_{\diamond i}\|_\infty}$ for some i, that must violate the constraint $Ax \leq \mathbb{1}$. □

This bounding-box constraint allows us to optimize over a bounded set for x.

Smoothed Objective. We now introduce the smoothed objective $f_\mu(x)$ that we minimize over Δ in order to approximately solve the packing LP. This objective $f_\mu(x)$ turns each row of the non-smooth LP constraint $Ax \leq \mathbb{1}$ into an exponential penalty function so that we only need to require $x \in \Delta$ throughout the algorithm. More formally, the packing LP can be written as the following minimizaton problem by introducing the Lagrangian variable $y \in \mathbb{R}^m$:

$$\min_{x\in\Delta} \; -\mathbb{1}^T x \; + \; \max_{y\geq 0}\{y^T Ax - \mathbb{1}^T y\} \; . \tag{6.2}$$

The problem can be now smoothened by introducing a strongly concave regularizer over $y \geq 0$.

This is regularizer is usually taken to be the entropy function over all possible $y \geq 0$ satisfying $\mathbb{1}^T y = 1$, which yields the width-independent solvers in for instance [118] and [113], and is closely related to that of the multiplicative weight update in [10].

In this paper, we take this regularizer to be the generalized entropy $H(y) = -\sum_{j=1}^m y_j \log y_j + y_j$ over the first orthant $y \geq 0$, and minimize the following smoothened objective $f_\mu(x)$ over $x \in \Delta$:

$$f_\mu(x) \stackrel{\text{def}}{=} -\mathbb{1}^T x \; + \; \max_{y\geq 0}\{y^T Ax - \mathbb{1}^T y + \boxed{\mu \cdot H(y)}\} \; . \tag{6.3}$$

Above, $\mu > 0$ is some smoothing parameter to be chosen later. By explicitly computing the maximization over $y \geq 0$, $f_\mu(x)$ can be rewritten as

Lemma 6.2. $f_\mu(x) = \mu \sum_{j=1}^m \exp^{\frac{1}{\mu}((Ax)_j - 1)} - \mathbb{1}^T x \; .$

We wish to study the *minimization* problem on $f_\mu(x)$ over $x \in \Delta$. Intuitively $f_\mu(x)$ captures the original packing LP (5.1) as follows. Firstly, since we want to maximize $\mathbb{1}^T x$, the negative term $-\mathbb{1}^T x$ shows up in $f_\mu(x)$. Secondly, if a packing constraint $j \in [m]$ is violated by ε, that is, $(Ax)_j \geq 1 + \varepsilon$, the exponential penalty in $f_\mu(x)$ introduces a penalty at least $\exp^{\varepsilon/\mu}$; this will be a large penalty if $\mu \leq O(\varepsilon/\log n)$. Notice that this smoothed objective also appeared in previous works [7], albeit without this smoothening interpretation and without the constraint $x \in \Delta$.

The regularization of Lemma 6.2 will give us both some smoothness properties for $f_\mu(x)$, discussed in Lemma 6.6, and a regularization error, as we are now solving an objective different from our original packing LP. This error is quantified in the following lemma for our choice of μ. This follows a similar treatment in a previous paper of the authors [7] and is proved in Appendix 6.A.

Proposition 6.3. *Let $\mu = \frac{\varepsilon}{4\log(nm/\varepsilon)}$ and x^* be an optimal solution for the packing LP (5.1). Then:*

(a) $f_\mu(u^*) \le -(1-\varepsilon)\mathsf{OPT}$ for $u^* \stackrel{\text{def}}{=} (1-\varepsilon/2)x^* \in \Delta$.

(b) $f_\mu(x) \ge -(1+\varepsilon)\mathsf{OPT}$ for every $x \in \Delta$.

(c) If $x \in \Delta$ satisfies $f_\mu(x) \le -(1-O(\varepsilon))\mathsf{OPT}$, then $\frac{1}{1+\varepsilon}x$ is a $(1-O(\varepsilon))$-approximate solution to the packing LP.

In short, they together imply that the minimum of $f_\mu(x)$ is around $-\mathsf{OPT}$, and if one can approximately find the minimum of $f_\mu(x)$, up to a multiplicative error $1 \pm O(\varepsilon)$, this corresponds to a $(1-O(\varepsilon))$-approximate solution to the packing LP (5.1).

Remark 6.4. We emphasize that our constraint $x_i \le \frac{1}{\|A_{\diamond i}\|_\infty}$ is essentially redundant from the viewpoint of minimizing $f_\mu(x)$: whenever $x \ge 0$ and $f_\mu(x) \le 0$, one should automatically have $x_i \le \frac{1+\varepsilon}{\|A_{\diamond i}\|_\infty}$. However, this redundant constraint shall become very crucial at the point we analyze the mirror-descent component our algorithm; after all, mirror descent steps do not necessarily decrease the objective, and thus may not guarantee $f_\mu(x) \le 0$.

Smoothness properties. Thanks to the smoothing of Lemma 6.2 and the choice of regularizer, our objective $f_\mu(x)$ enjoys a number of good smoothness properties. First, it is differentiable and the gradient is easy to compute:

Fact 6.5. $\nabla f_\mu(x) = A^T \mathsf{p}(x) - \mathbb{1} \quad \textit{where} \quad \mathsf{p}_j(x) \stackrel{\text{def}}{=} \exp^{\frac{1}{\mu}((Ax)_j - 1)}$.

Second, $f_\mu(x)$ enjoys two kinds of coordinate-wise smoothness properties in different regimes. These will be extremely useful in applying gradient descent arguments in Section 6.3.2, and are the main motivation for us to adopt the $\|\cdot\|_A$ norm for our proposed algorithms. Its proof is a simple manipulation of the Hessian.

Lemma 6.6. *Define the smoothness parameter $L \stackrel{\text{def}}{=} \frac{4}{\mu}$. Then, for every $x \in \Delta$, and every $i \in [n]$:*

(a) If $|\nabla_i f_\mu(x)| \le 1$, then for all $\lambda \le \frac{1}{L\|A_{\diamond i}\|_\infty}$, we have $\left|\nabla_i f_\mu(x+\lambda \mathbf{e}_i) - \nabla_i f_\mu(x)\right| \le L\|A_{\diamond i}\|_\infty \cdot |\lambda|$.

(b) If $|\nabla_i f_\mu(x)| \ge 1$, then for all $\lambda \le \frac{1}{L\|A_{\diamond i}\|_\infty}$, we have $\nabla_i f_\mu(x+\lambda \mathbf{e}_i) \ge \left(1 - \frac{\|A_{\diamond i}\|_\infty L}{2}|\lambda|\right)\nabla_i f_\mu(x)$.

Above, the first property is the same as the traditional (coordinate) Lipschitz-

smoothness property, i.e. the Lipschitz continuity of the (coordinate) gradient $\nabla_i f(x)$, but holds only conditionally and not for all $x \geq 0$. The second property is a salient characteristic of this work and requires the positivity of A. It can be seen as a formalization of the "multiplicative Lipschitz" property used in our previous work [7].

Proof of Lemma 6.6. Using the fact that $\nabla_i f_\mu(x) > -1$ for all x, we have:

$$\begin{aligned}\Big|\log\frac{\nabla_i f_\mu(x+\lambda\mathbf{e}_i)+1}{\nabla_i f_\mu(x)+1}\Big| &= \Big|\int_0^\lambda \frac{\nabla^2_{ii} f_\mu(x+\nu\mathbf{e}_i)}{\nabla_i f_\mu(x+\nu\mathbf{e}_i)+1}d\nu\Big| \\ &= \frac{1}{\mu}\Big|\int_0^\lambda \frac{(A^T\mathrm{diag}\{p(x+\nu\mathbf{e}_i)\}A)_{ii}}{(A^T p(x+\nu\mathbf{e}_i))_i}d\nu\Big| \\ &\leq \frac{\|A_{\diamond i}\|_\infty}{\mu}|\lambda| = \frac{\|A_{\diamond i}\|_\infty L}{4}|\lambda| \ .\end{aligned}$$

The last equality holds as $L = \frac{4}{\mu}$. This immediately implies the following multiplicative bound:

$$e^{-\frac{\|A_{\diamond i}\|_\infty L}{4}|\lambda|} \leq \frac{\nabla_i f_\mu(x+\lambda\mathbf{e}_i)+1}{\nabla_i f_\mu(x)+1} \leq e^{\frac{\|A_{\diamond i}\|_\infty L}{4}|\lambda|}.$$

By our assumption on λ, we know that $\frac{\|A_{\diamond i}\|_\infty L}{4}|\lambda| \leq \frac{1}{4}$, so that we can use the approximation $x \leq e^x - 1 \leq 1.2x$ over $x \in [-\frac{1}{4}, \frac{1}{4}]$. This yields the simpler bound:

$$-\frac{\|A_{\diamond i}\|_\infty L}{4}|\lambda| \leq \frac{\nabla_i f_\mu(x+\lambda\mathbf{e}_i) - \nabla_i f_\mu(x)}{\nabla_i f_\mu(x)+1} \leq 1.2\frac{\|A_{\diamond i}\|_\infty L}{4}|\lambda|.$$

Now we are ready to prove the two points of the lemma.

(a) Assuming that $\nabla_i f_\mu(x) \in (-1, 1]$, we have:

$$\Big|\nabla_i f_\mu(x+\lambda\mathbf{e}_i) - \nabla_i f_\mu(x)\Big| \leq 2.4 \cdot \frac{\|A_{\diamond i}\|_\infty L}{4}|\lambda| \leq \|A_{\diamond i}\|_\infty L|\lambda| \ .$$

(b) Assuming $\nabla_i f_\mu(x) \geq 1$, we have

$$\nabla_i f_\mu(x+\lambda\mathbf{e}_i) \geq \nabla_i f_\mu(x) - \frac{\|A_{\diamond i}\|_\infty L}{4}|\lambda|\Big(\nabla_i f_\mu(x)+1\Big) \geq \Big(1-\frac{\|A_{\diamond i}\|_\infty L}{2}|\lambda|\Big)\nabla_i f_\mu(x) \ .$$

□

Initialization. Iterative methods require the choice of a good starting point. We have

Fact 6.7. *Defining* $x_i^{\mathsf{start}} \stackrel{\text{def}}{=} \frac{1-\varepsilon/2}{n\|A_{\diamond i}\|_\infty}$ *for for each* $i \in [n]$, *we have* $x^{\mathsf{start}} \in \Delta$ *and* $f_\mu(x^{\mathsf{start}}) \leq -\frac{1-\varepsilon}{n}$.

Proof. Using the fact that $Ax^{\mathsf{start}} - \mathbb{1} \leq -\varepsilon/2 \cdot \mathbb{1}$, we compute $f_\mu(x^{\mathsf{start}})$ as follows:

$$\begin{aligned}f_\mu(x^{\mathsf{start}}) &= \mu\sum_j \exp^{\frac{1}{\mu}((Ax^{\mathsf{start}})_j - 1)} - \mathbb{1}^T x^{\mathsf{start}} \leq \mu\sum_j \exp^{\frac{-\varepsilon/2}{\mu}} - \frac{1-\varepsilon/2}{n} \\ &\leq \frac{\mu m}{(nm)^2} - \frac{1-\varepsilon/2}{n} \leq -\frac{1-\varepsilon}{n} \ .\end{aligned}$$

Algorithm 4 `PacLPSolver`$(A, x^{\mathsf{start}}, \varepsilon)$

Input: $A \in \mathbb{R}^{m\times n}_{\geq 0}, x^{\mathsf{start}} \in \Delta, \varepsilon \in (0, 1/10]$.
Output: $x \in \Delta$.

$\mu \leftarrow \frac{\varepsilon}{4\log(nm/\varepsilon)}$, $L \leftarrow \frac{4}{\mu}$, $\tau \leftarrow \frac{1}{3\cdot nL}$ and $\alpha_0 \leftarrow \frac{1}{nL}$. ▷ parameters
$T \leftarrow \lceil 3nL\log(1/\varepsilon)\rceil = O(n \cdot \frac{\log(nm/\varepsilon)\cdot\log(1/\varepsilon)}{\varepsilon})$. ▷ number of iterations
$\mathsf{x}_0 = \mathsf{y}_0 \leftarrow x^{\mathsf{start}}$, $\mathsf{z}_0 \leftarrow 0$.
for $k \leftarrow 1$ **to** T **do**
    $\alpha_k \leftarrow \frac{1}{1-\tau}\,\alpha_{k-1}$
    $\mathsf{x}_k \leftarrow \tau\mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1}$.
    Randomly select $i \in [n]$ uniformly at random.
    Define the vector $\xi_k^{(i)}$ to be all-zero except at coordinate i, where it equals $\mathbb{T}^{\mathsf{p}}(\nabla_i f_\mu(\mathsf{x}_k))$.
    $\mathsf{z}_k \leftarrow \mathsf{z}_k^{(i)} \stackrel{\text{def}}{=} \arg\min_{z\in\Delta}\left\{\frac{1}{2}\|z - \mathsf{z}_{k-1}\|_A^2 + \langle n\alpha_k\xi_k^{(i)}, z\rangle\right\}$. ▷ See Proposition 6.13
    $\mathsf{y}_k \leftarrow \mathsf{y}_k^{(i)} \stackrel{\text{def}}{=} \mathsf{x}_k + \frac{1}{n\alpha_k L}(\mathsf{z}_k^{(i)} - \mathsf{z}_{k-1})$.
end for
return y_T.

Above, we have used that $\mathbb{1}^T x^{\mathsf{start}} \geq x_i^{\mathsf{start}} = \frac{1-\varepsilon/2}{n}$, where i is the column such that $\|A_{\diamond i}\|_\infty = 1$. □

6.3 Our Packing LP Solver

To describe our algorithm, we first make the following choice of thresholding function

Definition 6.8. *The thresholding function* $\mathbb{T}^{\mathsf{p}}\colon [-1,\infty) \to [-1,1]$ *is defined as follows*

$$\mathbb{T}^{\mathsf{p}}(v) \stackrel{\text{def}}{=} \begin{cases} v, & v \in [-1,1]; \\ 1, & v > 1. \end{cases}$$

Our algorithm `PacLPSolver` starts with some initial vector $\mathsf{x}_0 = \mathsf{y}_0 = x^{\mathsf{start}}$ (introduced in Fact 6.7) and $\mathsf{z}_0 = 0$, and is divided into T iterations. In each iteration, we start by computing a weighted midpoint $\mathsf{x}_k \leftarrow \tau\mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1}$ for some parameter $\tau \in (0,1)$, and then proceed to compute y_k and z_k as follows.

- Select $i \in [n]$ uniformly at random, and let $\xi_k^{(i)} = (0,\ldots,0,\mathbb{T}^{\mathsf{p}}(v),0,\ldots,0)$ be the vector that is only non-zero at coordinate i, where $v = \nabla_i f_\mu(\mathsf{x}_k) = \sum_{j=1}^m A_{j,i}\exp^{\frac{1}{\mu}((A\mathsf{x}_k)_j - 1)} - 1 \in [-1,\infty)$.
- Perform a *mirror (descent) step* $\mathsf{z}_k \leftarrow \mathsf{z}_k^{(i)} \stackrel{\text{def}}{=} \arg\min_{z\in\Delta}\left\{\frac{1}{2}\|z - \mathsf{z}_{k-1}\|_A^2 + \langle n\alpha_k\xi_k^{(i)}, z\rangle\right\}$ for some parameter $\alpha_k \ll 1/n$ to be chosen later.
- Perform a *gradient (descent) step* $\mathsf{y}_k \leftarrow \mathsf{y}_k^{(i)} \stackrel{\text{def}}{=} \mathsf{x}_k + \frac{1}{n\alpha_k L}(\mathsf{z}_k^{(i)} - \mathsf{z}_{k-1})$.

Above, the reason that the the two steps on y_k and z_k are named after "gradient step" and "mirror step" will become clear in the follow-up sections. We use the superscript $^{(i)}$ on $\xi_i^{(i)}$, $\mathsf{y}_k^{(i)}$ and $\mathsf{z}_k^{(i)}$ to emphasize that the value depends on the choice of i. We have

used generic parameters τ, α_k, T in the above description and their precise values are presented in Algorithm 4. [5]

For readers familiar with accelerated first-order methods, the above triple sequence $\{\mathsf{x}_k, \mathsf{y}_k, \mathsf{z}_k\}_k$ is reminiscent of Nesterov's accelerated gradient method [118]. However, our algorithm is not an instance of any variant of the known accelerated gradient method. (This is so because, for instance, our objective $f_\mu(x)$ is not globally Lipschitz smooth.)

In fact, our algorithm `PacLPSolver` is strongly motivated by our linear-coupling technique introduced in [5] (see Chapter 4), a technique that allows one to linearly combine gradient and mirror steps for a better performance. This linear coupling requires one to use a triple sequence $\{\mathsf{x}_k, \mathsf{y}_k, \mathsf{z}_k\}_k$.

We emphasize here that our iterates $\mathsf{x}_k, \mathsf{y}_k, z_k$ never leave the bounding box Δ:

Lemma 6.9. *We have* $\mathsf{x}_k, \mathsf{y}_k, \mathsf{z}_k \in \Delta$ *for all* $k = 0, 1, \ldots, T$.

The proof of Lemma 6.9 is deferred to Appendix 6.B, and crucially relies on the fact that our gradient and mirror steps are multiples of each other: $\mathsf{y}_k^{(i)} - \mathsf{x}_k = \frac{1}{n\alpha_k L}(\mathsf{z}_k^{(i)} - \mathsf{z}_{k-1})$. The key idea of this lemma was also known by Fercoq and Richtárik [61].

We shall also prove in Section 6.E that

Lemma 6.10. *Each iteration of* `PacLPSolver` *can be implemented to run in expected* $O(N/n)$ *time.*

The key idea used in the implementation is to compute x_k and y_k only *implicitly*. For instance, explicitly maintaining x_k and computing $\mathsf{p}(x_k)$ require $O(N)$ time per iteration, but representing x_k implicitly as a linear combination of two less-frequently-modified vectors reduces it to $O(N/n)$.

In this section, we shall prove the following theorem in three steps.

Theorem 6.11. `PacLPSolver`$(A, x^{\mathsf{start}}, \varepsilon)$ *outputs some* y_T *satisfying*

$$\mathbb{E}[f_\mu(\mathsf{y}_T)] \le -(1 - 3\varepsilon)\mathsf{OPT} \ .$$

6.3.1 Step 1: Mirror Descent Guarantee

Since our update $\mathsf{z}_k^{(i)} = \arg\min_{z \in \Delta} \left\{ \frac{1}{2}\|z - \mathsf{z}_{k-1}\|_A^2 + \langle n\alpha_k \xi_k^{(i)}, z \rangle \right\}$ —see Line 9 of `PacLPSolver`— is written in the form of a *mirror descent step* from optimization, the following inequality is a classical upper bound on the "regret" of mirror descent. Its proof can be found in Appendix 6.B.

[5] We encourage the readers to ignore their specific values for now. Our specific choices of the parameters shall become clearer and natural at the end of this section, and be discussed whenever they are used.

Lemma 6.12. $\left\langle n\alpha_k \xi_k^{(i)}, \mathsf{z}_{k-1} - u \right\rangle \le n^2\alpha_k^2 L \cdot \left\langle \xi_k^{(i)}, \mathsf{x}_k - \mathsf{y}_k^{(i)} \right\rangle + \frac{1}{2}\|\mathsf{z}_{k-1} - u\|_A^2 - \frac{1}{2}\|\mathsf{z}_k^{(i)} - u\|_A^2$.

Although defined in a variational way, it is perhaps beneficial to explicitly describe how to implement this mirror step. Its proof is straightforward but can be found in Appendix 6.B.

Proposition 6.13. *If* $\mathsf{z}_{k-1} \in \Delta$*, the minimizer* $z = \arg\min_{z\in\Delta} \left\{ \frac{1}{2}\|z - \mathsf{z}_{k-1}\|_A^2 + \langle \delta \mathbf{e}_i, z\rangle \right\}$ *for any scalar* $\delta \in \mathbb{R}$ *and basis vector* $\mathbf{e}_i$ *can be computed as follows:*

1. $z \leftarrow \mathsf{z}_{k-1}$.
2. $z_i \leftarrow z_i - \delta/\|A_{\diamond i}\|_\infty$.
3. *If* $z_i < 0$*, then* $z_i \leftarrow 0$*; if* $z_i > 1/\|A_{\diamond i}\|_\infty$*,* $z_i \leftarrow 1/\|A_{\diamond i}\|_\infty$.
4. *Return* z.

As a simple corollary, we have the following fact

Fact 6.14. *We have* $|\mathsf{z}_{k,i}^{(i)} - \mathsf{z}_{k-1,i}| \le \frac{n\alpha_k|\xi_{k,i}^{(i)}|}{\|A_{\diamond i}\|_\infty}$ *and* $|\mathsf{y}_{k,i}^{(i)} - \mathsf{x}_{k,i}| = \frac{1}{n\alpha_k L}|\mathsf{z}_{k,i}^{(i)} - \mathsf{z}_{k-1,i}| \le \frac{|\xi_{k,i}^{(k)}|}{L\|A_{\diamond i}\|_\infty} \le \frac{1}{L\|A_{\diamond i}\|_\infty}$.

6.3.2 Step 2: Gradient Descent Guarantee

We call our update rule $\mathsf{y}_k^{(i)} \leftarrow \mathsf{x}_k + \frac{1}{n\alpha_k L}(\mathsf{z}_k^{(i)} - \mathsf{z}_{k-1})$ a gradient descent step, because the following lemma guarantees $f_\mu(\mathsf{y}_k^{(i)}) \le f_\mu(\mathsf{x}_k)$, that is, the objective only decreases; moreover, the objective decreases at least by $\frac{1}{2}\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)}\rangle$.

Lemma 6.15. *We have* $f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) \ge \frac{1}{2}\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)}\rangle$*. In particular, this implies* $f_\mu(\mathsf{x}_k) \ge f_\mu(\mathsf{y}_k^{(i)})$ *because* $\nabla_i f_\mu(\mathsf{x}_k)$ *and* $\mathsf{x}_{k,i} - \mathsf{y}_{k,i}^{(i)}$ *have the same sign, while* $\mathsf{x}_{k,\ell} = \mathsf{y}_{k,\ell}^{(i)}$ *for* $\ell \ne i$.

Proof. Note that $\mathsf{y}_k^{(i)} = \mathsf{x}_k + \lambda \mathbf{e}_i$ for some step length λ such that $|\lambda| \le \frac{1}{L\|A_{\diamond i}\|_\infty}$ according to Fact 6.14. We first prove this lemma in the case of $\nabla_i f_\mu(\mathsf{x}_k) \in [-1, 1]$ so that $\xi_{k,i}^{(i)} = \nabla_i f_\mu(\mathsf{x}_k)$.

$$
\begin{aligned}
f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) = f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{x}_k + \lambda\mathbf{e}_i) &= -\int_0^\lambda \Big(\nabla_i f_\mu(\mathsf{x}_k + \chi\mathbf{e}_i)\Big) d\chi \\
&\overset{①}{\ge} \int_0^\lambda \Big(-\nabla_i f_\mu(\mathsf{x}_k) - L\|A_{\diamond i}\|_\infty \cdot |\chi|\Big) d\chi \\
&= -\nabla_i f_\mu(\mathsf{x}_k)\cdot|\lambda| - \frac{L\|A_{\diamond i}\|_\infty}{2}\cdot\lambda^2 \\
&\overset{②}{\ge} -\nabla_i f_\mu(\mathsf{x}_k)\cdot|\lambda| - \frac{L\|A_{\diamond i}\|_\infty}{2}\cdot|\lambda|\cdot\frac{|\xi_{k,i}^{(k)}|}{L\|A_{\diamond i}\|_\infty} \\
&= -\frac{1}{2}\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{y}_k^{(i)} - \mathsf{x}_k\rangle \ .
\end{aligned}
$$

Above, ① uses Lemma 6.6.a, and ② uses Fact 6.14.

Next, we turn to the case of $\nabla_i f_\mu(\mathsf{x}_k) > 1$.

$$\begin{aligned} f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) = f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{x}_k + \lambda \mathbf{e}_i) &= -\int_0^\lambda \nabla_i f_\mu(\mathsf{x}_k + \chi \mathbf{e}_i) d\chi \\ &\overset{①}{\geq} \int_0^\lambda \Big(1 - \frac{\|A_{\diamond i}\|_\infty L}{2}|\chi|\Big) \nabla_i f_\mu(x) d\chi \\ &\overset{②}{\geq} \int_0^\lambda \frac{1}{2} \nabla_i f_\mu(x) d\chi = \frac{1}{2}\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)} \rangle \ . \end{aligned}$$

Above, ① uses Lemma 6.6.b and ② uses $|\chi| \leq |\lambda| \leq \frac{1}{L\|A_{\diamond i}\|_\infty}$. □

6.3.3 Step 3: Putting All Together

In the following, we denote by $\eta_k^{(i)} \in \mathbb{R}^n_{\geq 0}$ the vector that is only non-zero at coordinate i, and satisfies $\eta_{k,i}^{(i)} = \nabla_i f_\mu(\mathsf{x}_k) - \xi_{k,i}^{(i)} \in [0, \infty)$. In other words, the full gradient

$$\nabla f_\mu(\mathsf{x}_k) = \mathbb{E}_i[n \nabla_i f_\mu(\mathsf{x}_k)] = \mathbb{E}_i[n\eta_k^{(i)} + n\xi_k^{(i)}]$$

can be (in expectation) decomposed into the a large but non-negative component $\eta_k^{(i)} \in [0, \infty)^n$ and a small component $\xi_k^{(i)} \in [-1, 1]^n$. Recall that $\eta_k^{(i)}$ is the part of the gradient that was truncated, and did not contribute to the mirror step (see Line 9 of `PacLPSolver`). Next, for any $u \in \Delta$, we can use a basic convexity argument and the mirror descent lemma to compute that

$$\begin{aligned} &\alpha_k(f_\mu(\mathsf{x}_k) - f_\mu(u)) \leq \big\langle \alpha_k \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - u \big\rangle \\ &= \big\langle \alpha_k \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{z}_{k-1} \big\rangle + \big\langle \alpha_k \nabla f_\mu(\mathsf{x}_k), \mathsf{z}_{k-1} - u \big\rangle \\ &= \big\langle \alpha_k \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{z}_{k-1} \big\rangle + \mathbb{E}_i \Big[\big\langle n\alpha_k \eta_k^{(i)}, \mathsf{z}_{k-1} - u \big\rangle + \big\langle n\alpha_k \xi_k^{(i)}, \mathsf{z}_{k-1} - u \big\rangle \Big] \\ &\overset{①}{=} \frac{(1-\tau)\alpha_k}{\tau} \big\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{y}_{k-1} - \mathsf{x}_k \big\rangle + \mathbb{E}_i \Big[\big\langle n\alpha_k \eta_k^{(i)}, \mathsf{z}_{k-1} - u \big\rangle + \big\langle n\alpha_k \xi_k^{(i)}, \mathsf{z}_{k-1} - u \big\rangle \Big] \end{aligned} \tag{6.4}$$

$$\begin{aligned} &\overset{②}{\leq} \frac{(1-\tau)\alpha_k}{\tau}(f_\mu(\mathsf{y}_{k-1}) - f_\mu(\mathsf{x}_k)) \\ &\quad + \mathbb{E}_i \Big[\boxed{\big\langle n\alpha_k \eta_k^{(i)}, \mathsf{z}_{k-1} - u \big\rangle + n^2 \alpha_k^2 L \cdot \big\langle \xi_k^{(i)}, \mathsf{x}_k - \mathsf{y}_k^{(i)} \big\rangle} + \frac{1}{2}\|\mathsf{z}_{k-1} - u\|_A^2 - \frac{1}{2}\|\mathsf{z}_k^{(i)} - u\|_A^2 \Big] \end{aligned} \tag{6.5}$$

Above, ① is because $\mathsf{x}_k = \tau \mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1}$, which implies that $\tau(\mathsf{x}_k - \mathsf{z}_{k-1}) = (1-\tau)(\mathsf{y}_{k-1} - \mathsf{x}_k)$. ② uses convexity and Lemma 6.12. This above computation is motivated by [5] (see Chapter 4), and as we shall see below, it allows one to linearly couple gradient and mirror steps.

Intuitively, the first (non-negative) term in the box of (6.5) is the loss introduced by the large gradient $\eta_k^{(i)}$. This part was truncated so did not contribute to the mirror step. The second (non-negative) term in the box is the loss introduced by mirror descent on the small gradient $\xi_k^{(i)}$.

Now comes an important observation. As shown by Lemma 6.16 below, the performance of the gradient step —that is, the objective decrease of $f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)})$— is at least proportional to the loss incurred in the box.

Lemma 6.16. $\big\langle n\alpha_k\eta_k^{(i)}, \mathsf{z}_{k-1}-u\big\rangle + n^2\alpha_k^2 L\cdot\big\langle \xi_k^{(i)}, \mathsf{x}_k-\mathsf{y}_k^{(i)}\big\rangle \le 3n\alpha_k L\cdot(f_\mu(\mathsf{x}_k)-f_\mu(\mathsf{y}_k^{(i)}))$.

Since the proof of the above lemma is a careful case analysis and several simple applications of Lemma 6.15, we defer it to Appendix 6.B. We make two important remarks.

- First, Lemma 6.16 is why we stated in the introduction that our `PacLPSolver` incorporates *two* gradient steps: one with respect to $\eta_k^{(i)}$ and one with respect to $\xi_k^{(i)}$. We have intentionally forced the two steps to be identical, in order to present our algorithm more cleanly.[6]
- Second, to properly upper bound $\langle n\alpha_k\eta_k^{(i)}, \mathsf{z}_{k-1}-u\rangle$, one needs to have some good upper bound the coordinates of z_{k-1}. This is exactly the place we need our redundant-constraint technique, which guarantees that each $\mathsf{z}_{k-1,i} \le \frac{1}{\|A_{\diamond i}\|_\infty}$.

Plugging the above lemma into (6.5), we have

$$\begin{aligned}
&\alpha_k(f_\mu(\mathsf{x}_k) - f_\mu(u)) \le \big\langle \alpha_k\nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - u\big\rangle \\
&\overset{①}{\le} \frac{(1-\tau)\alpha_k}{\tau}(f_\mu(\mathsf{y}_{k-1}) - f_\mu(\mathsf{x}_k)) \\
&\quad + \mathbb{E}_i\Big[3n\alpha_k L\cdot(f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)})) + \frac{1}{2}\|\mathsf{z}_{k-1}-u\|_A^2 - \frac{1}{2}\|\mathsf{z}_k - u\|_A^2\Big] \\
&\overset{②}{\le} \alpha_k f_\mu(\mathsf{x}_k) + \big(3n\alpha_k L - \alpha_k\big) f_\mu(\mathsf{y}_{k-1}) \\
&\quad + \mathbb{E}_i\Big[-3n\alpha_k L\cdot f_\mu(\mathsf{y}_k^{(i)}) + \frac{1}{2}\|\mathsf{z}_{k-1}-u\|_A^2 - \frac{1}{2}\|\mathsf{z}_k - u\|_A^2\Big] \ . \qquad (6.6)
\end{aligned}$$

Above, ① is because we have chosen α_k so that $n\alpha_k \le n\alpha_T = \frac{1}{\varepsilon L} \le \frac{1}{4}$; and ② is because we have chosen τ to satisfy $\frac{1}{\tau} = 3nL$.

Next, recall that we have picked α_k so that $(3nL-1)\alpha_k = 3nL\cdot\alpha_{k-1}$ in Algorithm 4. Telescoping (6.6) for $k = 1, \dots, T$ and choosing $u^* = (1-\varepsilon/2)x^*$, we have

$$-\textstyle\sum_{k=1}^T \alpha_k f_\mu(u^*) \le 3f_\mu(\mathsf{y}_0) - 3n\alpha_T L\cdot\mathbb{E}[f_\mu(\mathsf{y}_T)] + \|\mathsf{z}_0 - u^*\|_A^2 \le -3n\alpha_T L\cdot\mathbb{E}[f_\mu(\mathsf{y}_T)] + \mathsf{OPT} \ .$$

Here, the second inequality is due to $f_\mu(\mathsf{y}_0) = f_\mu(x^{\mathsf{start}}) \le 0$ from Fact 6.7, and the fact that

$$\|\mathsf{z}_0 - u^*\|_A^2 = \|u^*\|_A^2 = \textstyle\sum_{i=1}^n (u_i^*)^2\cdot\|A_{\diamond i}\|_\infty \le \sum_{i=1}^n (x_i^*)^2\cdot\|A_{\diamond i}\|_\infty \le \sum_{i=1}^n x_i^* = \mathsf{OPT} \ .$$

Finally, using the fact that $\sum_{k=1}^T \alpha_k = \alpha_T\cdot\sum_{k=0}^{T-1}\big(1-\frac{1}{3nL}\big)^k = 3n\alpha_T L\big(1-(1-\frac{1}{3nL})^T\big)$, we rearrange and obtain that

$$\mathbb{E}[f_\mu(\mathsf{y}_T)] \le \frac{\sum_k \alpha_k}{3n\alpha_T L} f_\mu(u^*) + \frac{1}{3n\alpha_T L}\mathsf{OPT} = \big(1-(1-\frac{1}{3nL})^T\big) f_\mu(u^*) + \frac{1}{3n\alpha_T L}\mathsf{OPT} \ .$$

Choosing $T = \lceil 3nL\log(1/\varepsilon)\rceil$ so that $\frac{1}{n\alpha_T L} = (1-\frac{1}{3nL})^T \le \varepsilon$. Combining this with

[6] One can in fact separate the two gradient steps as $\mathsf{x}_k \to \mathsf{y}_k$ and $\mathsf{x}_k \to \mathsf{y}'_k$, but that will make the algorithm description only more involved.

the fact that $f_\mu(u^*) \le -(1-\varepsilon)\mathsf{OPT} < 0$ (see Proposition 6.3.a), we obtain

$$\mathbb{E}[f_\mu(\mathsf{y}_T)] \le (1-\varepsilon) f_\mu(u^*) + \varepsilon/3 \cdot \mathsf{OPT} < -(1-3\varepsilon)\mathsf{OPT} \ .$$

Therefore, we have finished proving Theorem 6.11. □

It is now straightforward (but anyways proved in Appendix 6.B) to use Markov inequality to turn the expected guarantee in Theorem 6.11 into a probabilistic one:

Corollary 6.17. *With probability at least* 9/10, $\texttt{PacLPSolver}(A, x^{\mathsf{start}}, \varepsilon)$ *outputs a* $(1-O(\varepsilon))$ *approximate solution to the packing LP program. The expected running time is* $O(\frac{\log(nm/\varepsilon)\log(1/\varepsilon)}{\varepsilon} N)$.

6.4 Sketching the Main Ideas for Our Covering LP Solver

For the reasons stated in the introduction, we are forced to build a covering LP solver from scratch, rather than implicitly from `PacLPSolver`. We begin with a similar relaxation of the covering LP (5.2). That is, we show in Appendix 6.5 that it suffices to minimize

$$f_\mu(x) \stackrel{\text{def}}{=} \mu \textstyle\sum_{j=1}^m \exp^{\frac{1}{\mu}(1-(Ax)_j)} + \mathbb{1}^T x$$

over all $x \ge 0$. For technical reasons, this objective is much harder to work with than that of (6.3), because its gradient $\nabla f_\mu(x) \in (-\infty, 1]^n$ may be very negative. (This is why our prior work [7] or Chapter 5 intentionally avoided to solve covering LP directly.)

This time, we again pick a random coordinate $i \in [n]$ at each iteration, and then decompose $\nabla_i f(\mathsf{x}_k) = \xi + \eta$. Quite different from `PacLPSolver`, we define $\eta \in (-\infty, 0]$ to be the (negative) large gradient component, and $\xi \in [-\sqrt{\varepsilon}, 1]$ to be the small gradient component. Our main idea is to perform

- a gradient (descent) step with respect to η, and
- a mirror (descent) step with respect to ξ.

Note that we have intentionally truncated the gradient $\nabla_i f(\mathsf{x}_k)$ at (negative) $\sqrt{\varepsilon}$, rather than at 1 as in `PacLPSolver`. This is so because, as it is much harder to deal with negative gradients in the covering LP case, we cannot perform both a mirror and a gradient step anymore on the small component ξ, as it was in `PacLPSolver`; instead, we can only perform a single mirror step on ξ. If ξ were between -1 and 1, and even if η were always zero, classical theory of mirror descent (or multiplicative weight update) could only imply that the mirror step converges at a rate of $\propto \varepsilon^{-2}$. Instead, we discover that if we truncate the gradient to $\xi \in [-\sqrt{\varepsilon}, 1]$, a *negative-width technique* allows us to improve this convergence from ε^{-2} to $\varepsilon^{-1.5}$. This is the first time that this gradient truncation technique is performed non-symmetrically.

Due to this weaker truncation at $-\sqrt{\varepsilon}$ instead of -1, our gradient step enjoys

a convergence rate that is only $\propto \varepsilon^{-1.5}$, matching that of the mirror step. This is precisely why we truncate the gradient at $\sqrt{\varepsilon}$, as it provides the best truncation tradeoff between gradient and mirror descent.

It is perhaps worth mentioning that our gradient step is equipped with an novel analysis quite different from its classical counterpart in optimization theory. Traditionally, given convex function $g(x)$, the convergence analysis only uses the simple upper bound $g(x) - g(x^*) \leq \langle \nabla g(x), x - x^* \rangle$ on the objective distance to optimum. If $g(x) = e^{-x}$ is a univariate function, $x = -1$, and $x^* = -100$, this upper bound becomes $e^{-1} \approx e^{-1} - e^{-100} \leq e^{-1} \cdot 99$, which is too weak to be used. This is the place we need to use a *distance-adjustment technique*, which will effectively improve the distance estimation to the optimum.

The detailed description and the analysis of our `CovLPSolver` can be found in Appendix 6.6.

6.5 Relaxation of the Covering Linear Program

Recall that, for input matrix $A \in \mathbb{R}^{m \times n}_{\geq 0}$, the covering LP in its standard form is

$$\text{Covering LP:} \qquad \min_{x \geq 0} \{ \mathbb{1}^T x \,:\, Ax \geq \mathbb{1} \} \ .$$

Let us denote by OPT the optimal value to this linear program, and by x^* any optimal solution of the covering LP (5.2). We say that x is a $(1+\varepsilon)$-approximation for the covering LP if $Ax \geq \mathbb{1}$ and $\mathbb{1}^T x \leq (1+\varepsilon)\mathsf{OPT}$. In our covering LP solver, we assume that some 2-approximate solution $x^\sharp$ is given to the algorithm, and $\mathbb{1}^T x^\sharp = \mathsf{OPT}'$ for some $\mathsf{OPT}' \in [\mathsf{OPT}, 2\mathsf{OPT}]$.[7]

Again, we use the indices $i \in [n]$ for the columns of A, and the indices $j \in [m]$ for the rows of A. We denote by $A_{\diamond i}$ the i-th column vector of A, and $A_{j\diamond}$ the j-th row vector of A. We can assume without loss of generality that[8]

$$\min_{j \in [m]} \{ \|A_{j\diamond}\|_\infty \} = 1 \ . \tag{6.7}$$

We now introduce the smoothed objective $f_\mu(x)$ that we are going to minimize in order to approximately solve the covering LP. We skip the details regarding how it arises from a relaxation using the generalized entropy regularizer, because it is essentially a repetition of Section 6.2.

This smoothed objective turns each row of the LP constraint $Ax \geq \mathbb{1}$ into an exponential penalty function so that we only need to require $x \geq 0$ throughout the algorithm.

[7]This can be obtained via for instance the covering LP solver from Young [166], whose running time is $O(N \log N)$. It can be relaxed to any constant approximation rather than 2-approximation.

[8]We can do so because first of all, we can assume $\min_{j \in [n]} \{ \|A_{j\diamond}\|_\infty \} > 0$ since otherwise the covering LP is infeasible. Next, we can scale A down by a factor of $\min_{j \in [n]} \{ \|A_{j\diamond}\|_\infty \}$; this also scales down the optimal value OPT and solution x^* by this same factor.

Definition 6.18. *Letting parameter* $\mu \stackrel{\text{def}}{=} \frac{\varepsilon}{4\log(nm/\varepsilon)}$*, we define the smoothed objective* $f_\mu(x)$ *as*

$$f_\mu(x) \stackrel{\text{def}}{=} \mu \sum_{j=1}^{m} \exp^{\frac{1}{\mu}(1-(Ax)_j)} + \mathbb{1}^T x$$

over the simplex $x \in \Delta \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n : x_i \geq 0 \ \wedge \ \mathbb{1}^T x \leq 2\mathsf{OPT}'\}$.

We wish to study the *minimization* problem on $f_\mu(x)$, subject to the constraint that each coordinate x_i is non-negative and the coordinates sum up to at most $2\mathsf{OPT}'$. The intuition that this smoothed objective $f_\mu(x)$ captures the original covering LP (5.2) is similar to that of the packing LP one. Note that our constraint $\mathbb{1}^T x \leq 2\mathsf{OPT}'$ is of course redundant; it will play some other important role in our algorithm.

We begin with several simple but important properties about OPT and $f_\mu(x)$. In short, they together imply that the minimum of $f_\mu(x)$ is around OPT, and if one can approximately find the minimum of $f_\mu(x)$ (up to an error $O(\varepsilon\mathsf{OPT})$), this corresponds to a $(1+O(\varepsilon))$-approximate solution to the covering LP (5.2). Since the proofs of these properties are completely analogous to their counterparts in packing LP, we defer them to Appendix 6.C.

Proposition 6.19.

(a) $\mathsf{OPT} \in [1, m]$.
(b) $f_\mu(u^*) \leq (1+\varepsilon)\mathsf{OPT}$ *for* $u^* \stackrel{\text{def}}{=} (1+\varepsilon/2)x^* \in \Delta$.
(c) $f_\mu(x) \geq (1-\varepsilon)\mathsf{OPT}$ *for every* $x \geq 0$.
(d) Letting $x^{\mathsf{start}} = (1+\varepsilon/2) \cdot x^\sharp + (\frac{1}{n}, \ldots, \frac{1}{n})$*, we have* $\mathbb{1}^T x^{\mathsf{start}} \leq 2\mathsf{OPT}'$ *and* $f_\mu(x^{\mathsf{start}}) \leq 4\mathsf{OPT}$.
(e) For any $x \geq 0$ *satisfying* $f_\mu(x) \leq 2\mathsf{OPT}$*, we must have* $Ax \geq (1-\varepsilon)\mathbb{1}$.
(f) If $x \geq 0$ *satisfies* $f_\mu(x) \leq (1+O(\varepsilon))\mathsf{OPT}$*, then* $\frac{1}{1-\varepsilon}x$ *is a* $(1+O(\varepsilon))$*-approximate solution to the covering LP.*
(g) The gradient of $f_\mu(x)$ *can be written as*

$$\nabla f_\mu(x) = \mathbb{1} - A^T \mathsf{p}(x) \quad \textit{where} \quad \mathsf{p}_j(x) \stackrel{\text{def}}{=} \exp^{\frac{1}{\mu}(1-(Ax)_j)} \tag{6.8}$$

6.6 Our Covering LP Solver

To describe our covering LP solver we make the following choice of the thresholding function. Recall in the packing LP case, we have truncated each coordinate gradient from $[-1, \infty)$ to $[-1, 1]$. For this covering LP case, we truncate each such gradient from $(-\infty, 1]$ to $[-\beta, 1]$, for some parameter $\beta \stackrel{\text{def}}{=} \sqrt{\varepsilon}$. The reason for this choice of $\beta = \sqrt{\varepsilon}$ shall become clear in later sections; at high level, $\sqrt{\varepsilon}$ is the best tradeoff between gradient and mirror descent.

Definition 6.20. *The thresholding function* $\mathbb{T}^{\mathsf{c}}\colon (-\infty, 1] \to [-\beta, 1]$ *is defined as fol-*

lows

$$\mathbb{T}^{\mathsf{c}}(v) \stackrel{\text{def}}{=} \begin{cases} v, & v \in [-\beta, 1]; \\ -\beta, & v < -\beta. \end{cases}$$

Our algorithm `CovLPSolver` starts with the initial vector $\mathsf{x}_0 = \mathsf{y}_0 = \mathsf{z}_0 = x^{\mathsf{start}}$ introduced in Proposition 6.19.d, and is divided into T iterations. In each iteration, we start by computing a weighted midpoint $\mathsf{x}_k \leftarrow \tau \mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1}$ for some parameter $\tau \in (0,1)$, and then proceed to compute y_k and z_k as follows.

- Select $i \in [n]$ uniformly at random, and let $\xi_k^{(i)} = (0, \ldots, 0, \mathbb{T}^{\mathsf{p}}(v), 0, \ldots, 0)$ be the vector that is only non-zero at coordinate i, where $v = \nabla_i f_\mu(\mathsf{x}_k) = 1 - \sum_{j=1}^m A_{j,i} \exp^{\frac{1}{\mu}(1-(A\mathsf{x}_k)_j)} \in (-\infty, 1]$.
- Perform a *mirror (descent) step* $\mathsf{z}_k \leftarrow \mathsf{z}_k^{(i)} \stackrel{\text{def}}{=} \arg\min_{z \in \Delta} \left\{ V_{\mathsf{z}_{k-1}}(z) + \langle (1+\gamma) n \alpha_k \xi_k^{(i)}, z \rangle \right\}$ for some parameters $\gamma \ll 1$ and $\alpha_k \ll 1/n$, where $V_x(y) = \sum_{i=1}^n y_i \log \frac{y_i}{x_i} + x_i - y_i$ is the so-called Bregman divergence of the generalized entropy function (see Proposition 6.28 below).
- Perform a *gradient (descent) step* $\mathsf{y}_k \leftarrow \mathsf{y}_k^{(i)} \stackrel{\text{def}}{=} \mathsf{x}_k + \delta \mathbf{e}_i$ for some value δ that is zero if $\nabla_i f_\mu(x_k) < -\beta$, and strictly positive otherwise. The precise definition of δ can be found in the pseudocode described in Algorithm 5.

Above, the reason that the the two steps on y_k and z_k are named after "gradient step" and "mirror step" will become clear in the follow-up sections. We use the superscript $^{(i)}$ on $\xi_i^{(i)}$, $\mathsf{y}_k^{(i)}$ and $\mathsf{z}_k^{(i)}$ to emphasize that the value depends on the choice of i. We have used generic parameters τ, α_k, T in the above description and their precise values are presented in Algorithm 5.[9]

Since the x^{start} satisfies $\mathbb{1}^T x^{\mathsf{start}} \leq 2\mathsf{OPT}'$ by Proposition 6.19.d, we have $\mathsf{z}_0 = x^{\mathsf{start}} \in \Delta$. Also, the mirror descent step ensures that $\mathsf{z}_{k,i} > 0$ for all rounds k and coordinates i, as well as $\mathsf{z}_k \in \Delta$ for all rounds k. However, we note that x_k and y_k may not necessarily lie inside Δ, but will always stay non-negative. We summarize these properties as follows:

$$\forall k \in \{0, 1, \ldots, T\}, \qquad \mathsf{x}_k, \mathsf{y}_k \geq 0, \quad \mathsf{z}_k > 0, \quad \mathsf{z}_k \in \Delta \ .$$

We shall also prove in Section 6.F that

Lemma 6.21. *Each iteration of* `CovLPSolver` *can be implemented to run in expected* $O(N/n)$ *time.*

The key idea is similar to that of the efficient implementation of `PacLPSolver`, that is to implementation the updates implicitly.

In this section, we prove the following theorem in five steps.

[9] We encourage the readers to ignore their specific values for now. Our specific choices of the parameters shall become clearer and natural at the end of this section, and be discussed whenever they are used.

Theorem 6.22. `CovLPSolver`$(A, x^{\mathsf{start}}, \varepsilon)$ *outputs some* y_T *satisfying*

$$\mathbb{E}[f_\mu(\mathsf{y}_T)] \le (1+9\varepsilon)\mathsf{OPT} \ .$$

6.6.1 Step 1: Distance Adjustment

Classically, using the convexity argument one can obtain $f_\mu(\mathsf{x}_k) - f_\mu(u) \le \langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - u\rangle$ for every $u \in \Delta$. In particular, if u is the optimal point, the right hand side is a simple upper bound on the objective distance from the current point $f_\mu(\mathsf{x}_k)$ to the optimum. This simple upper bound is essentially used by all the convergence analyses for first-order methods.

In this section, we strengthen this upper bound in the special case of $u = u^* \stackrel{\text{def}}{=} (1+\varepsilon/2)x^*$.

Define $\tilde{A}$ be the *adjusted matrix* of A described as follows.

Definition 6.23 (Adjusted matrix $\tilde{A}$)**.** *For each row $j \in [m]$, if $(Au^*)_j \le 2$ then we keep this row and define $\tilde{A}_{j\diamond} \stackrel{\text{def}}{=} A_{j\diamond}$. Otherwise, —that is, if $(Au^*)_j > 2$— we define $\tilde{A}_{j\diamond} \stackrel{\text{def}}{=} \frac{2}{(Au^*)_j} \cdot A_{j\diamond}$ to be the same j-th row $A_{j\diamond}$, but scaled down by a factor of $\frac{2}{(Au^*)_j}$. It is clear from this definition that*

$$A_{ji} \ge \tilde{A}_{ji} \text{ for all } i \in [n] \text{ and } j \in [m], \text{ while } (1+\varepsilon)\mathbb{1} \le \tilde{A}u^* \le 2\mathbb{1}.$$

We now strengthen the classical bound $f_\mu(\mathsf{x}_k) - f_\mu(u) \le \langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - u\rangle$ as follows.

Lemma 6.24 (Distance Adjustment)**.**

$$\begin{aligned} f_\mu(\mathsf{x}_k) - f_\mu(u^*) &\le \langle \mathbb{1} - A^T\mathsf{p}(\mathsf{x}_k), \mathsf{x}_k - u^*\rangle + \langle \tilde{A}^T\mathsf{p}(\mathsf{x}_k) - A^T\mathsf{p}(\mathsf{x}_k), u^*\rangle + \varepsilon\mathsf{OPT} \\ &= \langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - u^*\rangle + \langle \tilde{A}^T\mathsf{p}(\mathsf{x}_k) - A^T\mathsf{p}(\mathsf{x}_k), u^*\rangle + \varepsilon\mathsf{OPT} \end{aligned}$$

At high level, ignoring the negligible term $\varepsilon\mathsf{OPT}$ on the right hand side, the above upper bound strengthens the classical bound due to the extra term of $\langle \tilde{A}^T\mathsf{p}(\mathsf{x}_k) - A^T\mathsf{p}(\mathsf{x}_k), u^*\rangle$. This extra term is always non-positive since $\tilde{A} \le A$ coordinate-wisely, but may be very negative in certain cases.

The intuition behind the proof is to realize that the convexity inequality $e^b - e^a \le \langle e^b, b-a\rangle$ on the exponential function becomes far from tight when $a \ll 0$. For instance, when $b = 2$ and $a = -10$, we have $e^2 - e^{-10} \le 12e^2$; when $b = 2$ and $a = -100$, we only get $e^2 - e^{-100} \le 102e^2$. Although $e^{-100} \approx e^{-10}$, the two upper bounds are off from each other by a factor of 10. Therefore, when necessary, we can 'elevate' a to some higher value in order to obtain a tighter upper bound. We defer the detailed proof to Appendix 6.D.

6.6.2 Step 2: Gradient Truncation

Let us separate the indices $i \in [n]$ into large and small ones.

Definition 6.25. *We make the following definitions.*

- *Let* $\xi_k \in [-\beta, 1]^n$ *be the* **truncated gradient** *so that* $\xi_{k,i} = \mathbb{T}^{\mathsf{c}}(\nabla_i f_\mu(x_k))$ *for each* $i \in [n]$.
- *Let* $B_k \stackrel{\text{def}}{=} \{i \in [n] : \xi_{k,i} \neq \nabla_i f_\mu(\mathsf{x}_k)\}$ *be the set of* **large indices**.
- *Let* $\eta_k \in (-\infty, 0]^n$ *be the* **large gradient** *so that* $\nabla f_\mu(\mathsf{x}_k) = \xi_k + \eta_k$. *It is clear that*

$$\eta_{k,i} = 0 \text{ for every } i \notin B, \text{ and } \eta_{k,i} = (1+\beta) - (A^T \mathsf{p}(\mathsf{x}_k))_i \text{ for every } i \in B.$$

- *Let* $\tilde{\eta}_k \in (-\infty, \infty)^n$ *be the* **adjusted large gradient** *so that*

$$\tilde{\eta}_{k,i} = 0 \text{ for every } i \notin B, \text{ and } \tilde{\eta}_{k,i} = (1+\beta) - (\tilde{A}^T \mathsf{p}(\mathsf{x}_k))_i \text{ for every } i \in B.$$

For the rest of this section, we denote by $\eta_k^{(i)} = (0, \ldots, 0, \eta_{k,i}, 0, \ldots, 0)$, *the vector that is zero at all coordinates other than* i, *and equals to* $\eta_{k,i}$ *at location* i. *We similarly define* $\xi_k^{(i)}$ *as well as* $\tilde{\eta}_k^{(i)}$.

We next state the following key lemma that is very analogous to (6.4) from packing LP. Note that if one uses $\eta_k^{(i)}$ instead of $\tilde{\eta}_k^{(i)}$, the proof becomes identical to that of (6.4). The reason that we can use $\tilde{\eta}_k^{(i)}$ rather than $\eta_k^{(i)}$ —thus giving a stricter upper bound— is precisely due to the distance adjustment introduced in Lemma 6.24.

Lemma 6.26.

$$f_\mu(\mathsf{x}_k) - f_\mu(u^*) \leq \frac{(1-\tau)}{\tau}(f_\mu(\mathsf{y}_{k-1}) - f_\mu(\mathsf{x}_k)) + \mathbb{E}_i\Big[\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - u^*\rangle\Big] + \mathbb{E}_i\Big[\langle n\tilde{\eta}_k^{(i)}, -u^*\rangle\Big] + \varepsilon\mathsf{OPT} \ .$$

The proof of the above lemma is a simple repetition of that of (6.4), but replacing the classical distance upper bound with our adjusted one. See Appendix 6.D for details.

6.6.3 Step 3: Mirror Descent Guarantee

Our update $\mathsf{z}_k^{(i)} \stackrel{\text{def}}{=} \arg\min_{z \in \Delta} \left\{V_{\mathsf{z}_{k-1}}(z) + \langle (1+\gamma) n\alpha_k \xi_k^{(i)}, z\rangle\right\}$ is, by its definition, a mirror descent step. We begin by explaining an attempt that is too weak for obtaining the $\varepsilon^{-1.5}$ convergence rate.

Using the classical theory of mirror descent, it is not hard to repeat the proof of Lemma 6.12 —although changing the distance function from $\|\cdot\|_A^2$ to $V_x(y)$— and obtain that, for every $u \in \Delta$,

$$\mathbb{E}_i\big[\alpha_k\big\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - u\big\rangle\big] \leq V_{\mathsf{z}_{k-1}}(u) - \mathbb{E}_i\Big[V_{\mathsf{z}_k^{(i)}}(u)\Big] + O(\alpha_k^2 n)\mathsf{OPT} \ .$$

The above inequality can be made true whenever ξ_i is between -1 and 1 for each coordinate i, but only yields the known ε^{-2} convergence rate. Here, ± 1 is also know as the *width* from multiplicative-weight-update languages [10].

Fortunately, since we have required ξ_i to be only between $-\beta$ and 1, the $O(\alpha_k^2 n)$ factor can essentially be improved to $O(\alpha_k^2 \beta n)$. This is an improvement whenever $\beta \ll 1$, and we call it the *negative-width technique*.[10] Formally, we prove that

Lemma 6.27. *Denoting by* $\gamma \stackrel{\text{def}}{=} 2\alpha_T n$, *we have*

$$\mathbb{E}_i\big[\alpha_k\big\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - u^*\big\rangle\big] \le V_{\mathsf{z}_{k-1}}\big(\frac{u^*}{1+\gamma}\big) - \mathbb{E}_i\Big[V_{\mathsf{z}_k^{(i)}}\big(\frac{u^*}{1+\gamma}\big)\Big] + 12\mathsf{OPT}\cdot\gamma\alpha_k\beta \ .$$

The proof can be found in Appendix 6.D.

Although defined in a variational way, it is perhaps beneficial to explicitly describe how to implement this mirror step. The following proposition is straightforward but anyways proved in Appendix 6.D:

Proposition 6.28. *If* $\mathsf{z}_{k-1} \in \Delta$ *and* $\mathsf{z}_{k-1} > 0$, *the minimizer* $z = \arg\min_{z\in\Delta}\big\{V_{\mathsf{z}_{k-1}}(z) + \langle\delta\mathbf{e}_i, z\rangle\big\}$ *for any scalar* $\delta \in \mathbb{R}$ *and basis vector* $\mathbf{e}_i$ *can be computed as follows:*

1. $z \leftarrow \mathsf{z}_{k-1}$.
2. $z_i \leftarrow z_i \cdot e^{-\delta}$.
3. *If* $\mathbb{1}^T z > 2\mathsf{OPT}'$, $z \leftarrow \frac{2\mathsf{OPT}'}{\mathbb{1}^T z} z$.
4. *Return* z.

6.6.4 Step 4: Gradient Descent Guarantee

We claim that our gradient step $\mathsf{x}_k \to \mathsf{y}_k^{(i)}$ never increases the objective for all choices of i. In addition, it decreases the objective by an amount proportional to the adjusted large gradient $\tilde{\eta}_k^{(i)}$.

Lemma 6.29. *For every* $i \in [n]$, *we have*

(a) $f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) \ge 0$, *and*

(b) $f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) \ge \frac{\mu\beta}{12}\cdot\langle -\tilde{\eta}_k^{(i)}, u^*\rangle$.

The proof of Lemma 6.29 is quite technical and can be found in Appendix 6.D.

At high level, one would generally hope to prove that the gradient step decreases the objective by an amount proportional to the large gradient $\eta_k^{(i)}$, rather than the adjusted large gradient $\tilde{\eta}_k^{(i)}$. If that were true, the entire proof structure of our covering LP convergence would become much closer to that of packing LP, and there would be absolutely no need for the introduction of the distance adjustment in Section 6.6.1, as well as the definitions of $\tilde{A}$ and $\tilde{\eta}$.

Unfortunately, if one replaces $\tilde{\eta}$ with η in the above lemma, the inequality is *far* from being correct. The reason behind it is very similar to that we have summarized

[10] This negative width technique is strongly related to [10, Definition 3.2], where the authors analyze the classical multiplicative weight update method in a special case when the oracle returns loss values only between $-\ell$ and ρ, for $\ell \ll \rho$. This technique is in fact related to a more general theory of mirror descent, known as the local-norm convergence, that we have summarized in a separate paper [4] which corresponds to Chapter 8 of this thesis.

in Section 6.6.1, and related to the unpleasant behavior of the exponential penalty function.

6.6.5 Step 5: Putting All Together

Combining Lemma 6.26, Lemma 6.27, and Lemma 6.29, we obtain that

$$\begin{aligned}
&\alpha_k\big(f_\mu(\mathsf{x}_k) - f_\mu(u^*)\big) - \alpha_k\varepsilon\mathsf{OPT} \\
&\le \frac{(1-\tau)\alpha_k}{\tau}(f_\mu(\mathsf{y}_{k-1}) - f_\mu(\mathsf{x}_k)) + \mathbb{E}_i\Big[\alpha_k\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - u^*\rangle\Big] + \mathbb{E}_i\Big[\alpha_k\langle n\tilde{\eta}_k^{(i)}, -u^*\rangle\Big] \\
&\le \frac{(1-\tau)\alpha_k}{\tau}(f_\mu(\mathsf{y}_{k-1}) - f_\mu(\mathsf{x}_k)) + V_{\mathsf{z}_{k-1}}\big(\frac{u^*}{1+\gamma}\big) - \mathbb{E}_i\Big[V_{\mathsf{z}_k^{(i)}}\big(\frac{u^*}{1+\gamma}\big)\Big] \\
&\qquad\qquad + 12\mathsf{OPT}\cdot\gamma\alpha_k\beta + \mathbb{E}_i\Big[\frac{12\alpha_k n}{\mu\beta}\big(f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)})\big)\Big]
\end{aligned}$$

Remark 6.30. Above, the quantity "$12\mathsf{OPT}\cdot\gamma\alpha_k\beta$" is the loss term introduced by the mirror descent. Unlike the packing LP case —see (6.5)— this loss term is not dominated by the gradient step. (If one could do so, this would turn our `CovLPSolver` into an ε^{-1} convergence rate.)

The quantity "$\alpha_k\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - u^*\rangle$" is the loss introduced by the (adjusted) large gradient $\tilde{\eta}$, and is dominated by our gradient step progress owing to Lemma 6.29. This is similar to the packing LP case —see Lemma 6.16.

From here, let us use the special choice of $\tau = \frac{\mu\beta}{12n}$. We obtain that

$$\begin{aligned}
&-\alpha_k\big(f_\mu(u^*) + \varepsilon\mathsf{OPT}\big) \\
&\le 12\gamma\alpha_k\beta\mathsf{OPT} + \frac{(1-\tau)\alpha_k}{\tau}f_\mu(\mathsf{y}_{k-1}) + V_{\mathsf{z}_{k-1}}\big(\frac{u^*}{1+\gamma}\big) - \mathbb{E}_i\Big[\frac{\alpha_k}{\tau}f_\mu(\mathsf{y}_k^{(i)}) + V_{\mathsf{z}_k^{(i)}}\big(\frac{u^*}{1+\gamma}\big)\Big] \ .
\end{aligned}$$

Use the choice $\alpha_k = \frac{\alpha_{k-1}}{1-\tau}$ and telescoping the above inequality for $k = 1, \dots, T$, we have

$$-\big(\sum_{k=1}^T \alpha_k\big)\big(f_\mu(u^*) + \varepsilon\mathsf{OPT}\big) \le \big(\sum_{k=1}^T \alpha_k\big)\cdot 12\gamma\beta\mathsf{OPT} + \frac{\alpha_0}{\tau}f_\mu(\mathsf{y}_0) + V_{\mathsf{z}_0}\big(\frac{u^*}{1+\gamma}\big) - \frac{\alpha_T}{\tau}\mathbb{E}\big[f_\mu(\mathsf{y}_T)\big] \ .$$

We compute that $\sum_{k=1}^T \alpha_k = \alpha_T\cdot\sum_{k=0}^{T-1}(1-\tau)^k = \alpha_T\cdot\frac{1-(1-\tau)^T}{\tau} < \frac{\alpha_T}{\tau}$, and recall that $\gamma = 2\alpha_T n$. Therefore, we rearrange and get

$$\begin{aligned}
&\frac{\alpha_T}{\tau}\mathbb{E}\big[f_\mu(\mathsf{y}_T)\big] \le \frac{\alpha_T}{\tau}\big(f_\mu(u^*) + \varepsilon\mathsf{OPT}\big) + \frac{\alpha_T}{\tau}\cdot 12\gamma\beta\mathsf{OPT} + \frac{\alpha_0}{\tau}f_\mu(\mathsf{y}_0) + V_{\mathsf{z}_0}\big(\frac{u^*}{1+\gamma}\big) \ , \\
&\implies \mathbb{E}\big[f_\mu(\mathsf{y}_T)\big] \le f_\mu(u^*) + \varepsilon\mathsf{OPT} + 24\alpha_T n\beta\mathsf{OPT} + (1-\tau)^T f_\mu(\mathsf{y}_0) + \frac{\tau}{\alpha_T}V_{\mathsf{z}_0}\big(\frac{u^*}{1+\gamma}\big) \ .
\end{aligned} \tag{6.9}$$

From this point, we need to use our special choice of the initial point $\mathsf{x}_0 = \mathsf{y}_0 = \mathsf{z}_0 = x^{\mathsf{start}}$ (see Proposition 6.19.d), which implies that $f_\mu(\mathsf{y}_0) \le 4\mathsf{OPT}$ and $\mathbb{1}^T x^{\mathsf{start}} \le$

$4\mathsf{OPT}$. We also have

$$V_{\mathsf{z}_0}\big(\frac{u^*}{1+\gamma}\big) = V_{x^{\mathsf{start}}}\big(\frac{u^*}{1+\gamma}\big) = \sum_{i=1}^{n} \frac{u_i^*}{1+\gamma} \log \frac{u_i^*}{(1+\gamma)x_i^{\mathsf{start}}} + x_i^{\mathsf{start}} - \frac{u_i^*}{1+\gamma}$$
$$\overset{①}{\leq} \sum_{i=1}^{n} u_i^* \log(u_i^* \cdot n) + 4\mathsf{OPT} \overset{②}{\leq} (2\log(nm)+4)\cdot \mathsf{OPT} \ .$$

Above, inequality ① follows because $x_i^{\mathsf{start}} \geq 1/n$ for all $i \in [n]$ according to the definition in Proposition 6.19.d; inequality ② follows because $u_i^* \leq (1+\varepsilon/2)x_i^* \leq (1+\varepsilon/2)\mathsf{OPT} \leq (1+\varepsilon/2)m$ and $\mathbb{1}^T u_i^* = (1+\varepsilon/2)\mathsf{OPT}$, as well as the fact that ε is sufficiently small.

Finally, we choose $\beta = \sqrt{\varepsilon}$, $\alpha_T = \frac{\varepsilon}{12n\beta}$, and $T = \lceil \frac{1}{\tau}\log(1/\varepsilon)\rceil$. Substituting into (6.9) all of these parameters, along with the aforementioned inequalities $f_\mu(\mathsf{y}_0) \leq 4\mathsf{OPT}$ and $V_{\mathsf{z}_0}\big(\frac{u^*}{1+\gamma}\big) \leq (2\log(nm)+4)\cdot\mathsf{OPT}$, as well as $f_\mu(u^*) \leq (1+\varepsilon)\mathsf{OPT}$ from Proposition 6.19.b, we obtain that

$$\mathbb{E}\big[f_\mu(\mathsf{y}_T)\big] \leq (1+\varepsilon)\mathsf{OPT} + \varepsilon\mathsf{OPT} + 2\varepsilon\mathsf{OPT} + \varepsilon f_\mu(\mathsf{y}_0) + \frac{\mu\beta/12n}{\varepsilon/12n\beta}(2\log(nm)+4)\mathsf{OPT}$$
$$= (1+9\varepsilon)\mathsf{OPT} \ .$$

This finishes the proof of Theorem 6.22. □

It is now straightforward to use Markov inequality to turn the expected guarantee in Theorem 6.22 into a probabilistic one:

Corollary 6.31. *With probability at least* $9/10$, $\texttt{CovLPSolver}(A, x^{\mathsf{start}}, \varepsilon)$ *outputs a* $(1+O(\varepsilon))$ *approximate solution to the covering LP program. The expected running time is* $O(\frac{\log(nm/\varepsilon)\log(1/\varepsilon)}{\varepsilon^{1.5}}N)$.

Proof. Since for every $x \in \Delta$ it satisfies $f_\mu(x) \geq (1-\varepsilon)\mathsf{OPT}$ according to Proposition 6.19.c, we obtain that $f_\mu(y_T) - (1-\varepsilon)\mathsf{OPT}$ is a random variable that is non-negative, whose expectation $\mathbb{E}[f_\mu(y_T) - (1-\varepsilon)\mathsf{OPT}] \leq 10\varepsilon$. By Markov bound, with at least probability $9/10$, we obtain some y_T satisfying $f_\mu(y_T) \leq (1+O(\varepsilon))\mathsf{OPT}$, which yields some $(1+O(\varepsilon))$ approximate solution according to Proposition 6.19.f.

The running time follows from our efficient implementation in Section 6.F. □

Appendix

6.A Missing Proofs for Section 6.2

Proposition 6.3. *Let* $\mu = \frac{\varepsilon}{4\log(nm/\varepsilon)}$ *and* x^* *be an optimal solution for the packing LP* (5.1). *Then:*

(a) $f_\mu(u^*) \leq -(1-\varepsilon)\mathsf{OPT}$ *for* $u^* \overset{\text{def}}{=} (1-\varepsilon/2)x^* \in \Delta$.

(b) $f_\mu(x) \geq -(1+\varepsilon)\mathsf{OPT}$ *for every* $x \in \Delta$.

(c) *If* $x \in \Delta$ *satisfies* $f_\mu(x) \leq -(1-O(\varepsilon))\mathsf{OPT}$*, then* $\frac{1}{1+\varepsilon}x$ *is a* $(1-O(\varepsilon))$*-approximate solution to the packing LP.*

Proof.

(a) We have $\mathbb{1}^T u^* = (1-\varepsilon/2)\mathsf{OPT}$ by the definition of OPT. Also, from the feasibility constraint $Ax^* \leq \mathbb{1}$ in the packing LP, we have $Au^* - \mathbb{1} \leq -\varepsilon/2 \cdot \mathbb{1}$, and can compute $f_\mu(u^*)$ as follows:

$$f_\mu(u^*) = \mu \sum_j \exp^{\frac{1}{\mu}((Au^*)_j - 1)} - \mathbb{1}^T u^* \leq \mu \sum_j \exp^{\frac{-\varepsilon/2}{\mu}} - (1-\varepsilon/2)\mathsf{OPT}$$
$$\leq \frac{\mu m}{(nm)^2} - (1-\varepsilon/2)\mathsf{OPT} \leq -(1-\varepsilon)\mathsf{OPT} \ .$$

(b) Suppose towards contradiction that $f_\mu(x) < -(1+\varepsilon)\mathsf{OPT}$. Since $f_\mu(x) > -\mathbb{1}^T x$, it must satisfy that $\mathbb{1}^T x > (1+\varepsilon)\mathsf{OPT}$. Suppose that $\mathbb{1}^T x = (1+v)\mathsf{OPT}$ for some $v > \varepsilon$. By the definition of OPT, we must have that $Ax < (1+v)\mathbb{1}$ is broken, and therefore there exists some $j \in [m]$ satisfying that $(Ax)_j \geq 1+v$. In such a case, the objective

$$f_\mu(x) \geq \mu \exp^{v/\mu} - (1+v)\mathsf{OPT} = \frac{\varepsilon}{4\log(nm)}\Big((\frac{nm}{\varepsilon})^4\Big)^{v/\varepsilon} - (1+v)\mathsf{OPT}$$
$$\geq \Big(\Big((\frac{nm}{\varepsilon}\Big)^2)^{v/\varepsilon} - (1+v)\Big)\mathsf{OPT} > 0$$

giving a contradiction to the assumption that $f_\mu(x) < 0$.

(c) Suppose x satisfies $f_\mu(x) \leq -(1-O(\varepsilon))\mathsf{OPT} \leq 0$ and we first want to show $Ax \leq (1+\varepsilon)\mathbb{1}$. Let us assume that $v = \max_j((Ax)_j - 1) \geq 0$ because otherwise we will have $Ax \leq \mathbb{1}$. Under this definition, we have $Ax \leq (1+v)\mathbb{1}$ and therefore $\mathbb{1}^T x \leq (1+v)\mathsf{OPT}$ by the definition of OPT. We compute $f_\mu(x)$ as follows.

$$f_\mu(x) \geq \mu \exp^{\frac{v}{\mu}} - (1+v)\mathsf{OPT} \geq \mu\Big((\frac{nm}{\varepsilon})^4\Big)^{v/\varepsilon} - (1+v)n$$
$$= \frac{\varepsilon}{4\log(nm)}\Big((\frac{nm}{\varepsilon})^4\Big)^{v/\varepsilon} - (1+v)n \ .$$

It is easy to see that the above quantity is positive whenever $v \geq \varepsilon$, and therefore, to satisfy $f_\mu(x) \leq 0$ we must have $v \leq \varepsilon$, which is equivalent to $Ax \leq (1+\varepsilon)\mathbb{1}$.

Next, because $-\mathbb{1}^T x \leq f_\mu(x) \leq -(1-O(\varepsilon))\mathsf{OPT}$, we know that x yields an objective $\mathbb{1}^T x \geq (1-O(\varepsilon))\mathsf{OPT}$. Letting $x' = \frac{1}{1+\varepsilon}x$, we both have that x' is feasible (i.e., $Ax' \leq \mathbb{1}$), and x' has an objective $\mathbb{1}^T x'$ at least as large as $(1-O(\varepsilon))\mathsf{OPT}$. □

6.B Missing Proofs for Section 6.3

Lemma 6.9. *We have* $\mathsf{x}_k, \mathsf{y}_k, \mathsf{z}_k \in \Delta$ *for all* $k = 0, 1, \ldots, T$.

Proof. This is true at the beginning as $\mathsf{x}_0 = \mathsf{y}_0 = x^{\mathsf{start}} \in \Delta$ (see Fact 6.7) and $\mathsf{z}_0 = 0 \in \Delta$.

In fact, it suffices for us to show that for every $k \geq 0$, $\mathsf{y}_k = \sum_{l=0}^{k} \gamma_k^l \mathsf{z}_l$ for some scalers γ_k^l satisfying $\sum_l \gamma_k^l = 1$ and $\gamma_k^l \geq 0$ for each $l = 0, \ldots, k$. If this is true, we can prove the lemma by induction: at each iteration k,

1. $\mathsf{x}_k = \tau \mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1}$ must be in Δ because y_{k-1} and z_{k-1} are and $\tau \in [0, 1]$,
2. z_k is in Δ by the definition that $\mathsf{z}_k = \arg\min_{z \in \Delta}\{\cdots\}$, and
3. y_k is also in Δ because $\mathsf{y}_k = \sum_{l=0}^{k} \gamma_k^l \mathsf{z}_l$ is a convex combination of the z_l's and Δ is convex.

For the rest of the proof, we only need to show that $\mathsf{y}_k = \sum_{l=0}^{k} \gamma_k^l \mathsf{z}_l$ for[11]

$$\gamma_k^l = \begin{cases} (1-\tau)\gamma_{k-1}^l, & l = 0, \ldots, k-2; \\ \big(\frac{1}{n\alpha_{k-1}L} - \frac{1}{n\alpha_k L}\big) + \tau\big(1 - \frac{1}{n\alpha_{k-1}L}\big), & l = k-1; \\ \frac{1}{n\alpha_k L}, & l = k. \end{cases}$$

This is true at the base case because $\alpha_0 = \frac{1}{nL}$. It is also true at $k = 1$ because $\mathsf{y}_1 = \mathsf{x}_1 + \frac{1}{n\alpha_1 L}(\mathsf{z}_1 - \mathsf{z}_0) = \frac{1}{n\alpha_1 L}\mathsf{z}_1 + \big(1 - \frac{1}{n\alpha_1 L}\big)\mathsf{z}_0$. For the general k, we have

$$\begin{aligned} \mathsf{y}_k &= \mathsf{x}_k + \frac{1}{n\alpha_k L}(\mathsf{z}_k - \mathsf{z}_{k-1}) \\ &= \tau \mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1} + \frac{1}{n\alpha_k L}(\mathsf{z}_k - \mathsf{z}_{k-1}) \\ &= \tau \mathsf{z}_{k-1} + (1-\tau)\Big(\sum_{l=0}^{k-2} \gamma_{k-1}^l \mathsf{z}_l + \frac{1}{n\alpha_{k-1}L}\mathsf{z}_{k-1}\Big) + \frac{1}{n\alpha_k L}(\mathsf{z}_k - \mathsf{z}_{k-1}) \\ &= \Big(\sum_{l=0}^{k-2}(1-\tau)\gamma_{k-1}^l \mathsf{z}_l\Big) + \left(\Big(\frac{1}{n\alpha_{k-1}L} - \frac{1}{n\alpha_k L}\Big) + \tau\Big(1 - \frac{1}{n\alpha_{k-1}L}\Big)\right)\mathsf{z}_{k-1} + \frac{1}{n\alpha_k L}\mathsf{z}_k \ . \end{aligned}$$

Therefore, we obtain $\mathsf{y}_k = \sum_{l=0}^{k} \gamma_k^l \mathsf{z}_l$ as desired.

It is now easy to check that under our definition of α_k (which satisfies $\alpha_k \geq \alpha_{k-1}$ and $\alpha_k \geq \alpha_0 = \frac{1}{nL}$, we must have $\gamma_k^l \geq 0$ for all k and l. Also,

$$\sum_l \gamma_k^l = \sum_{l=0}^{k-2}(1-\tau)\gamma_{k-1}^l + \left(\Big(\frac{1}{n\alpha_{k-1}L} - \frac{1}{n\alpha_k L}\Big) + \tau\Big(1 - \frac{1}{n\alpha_{k-1}L}\Big)\right) + \frac{1}{n\alpha_k L}$$

[11] We wish to point out that this proof coincides with a lemma from the accelerated coordinate descent theory of Fercoq and Richtárik [61]. Their paper is about optimizing an objective function that is Lipschitz smooth, and thus irrelevant to our work.

$$= (1-\tau)\Big(1-\frac{1}{n\alpha_{k-1}L}\Big)+\bigg(\Big(\frac{1}{n\alpha_{k-1}L}-\frac{1}{n\alpha_k L}\Big)+\tau\Big(1-\frac{1}{n\alpha_{k-1}L}\Big)\bigg)+\frac{1}{n\alpha_k L}=1\ .$$

□

Lemma 6.12. *When* $\mathsf{z}_k^{(i)} = \arg\min_{z\in\Delta}\big\{\frac{1}{2}\|z-\mathsf{z}_{k-1}\|_A^2 + \langle n\alpha_k\xi_k^{(i)}, z\rangle\big\}$, *we have*

$$\big\langle n\alpha_k\xi_k^{(i)}, \mathsf{z}_{k-1}-u\big\rangle \le n^2\alpha_k^2 L\cdot\big\langle\xi_k^{(i)}, \mathsf{x}_k-\mathsf{y}_k^{(i)}\big\rangle + \frac{1}{2}\|\mathsf{z}_{k-1}-u\|_A^2 - \frac{1}{2}\|\mathsf{z}_k^{(i)}-u\|_A^2\ .$$

Proof. Denoting by $V_a(b)=\frac{1}{2}\|b-a\|_A^2$ as a function of $b\in\Delta$ parameterized at $a\in\Delta$, we have that $\nabla_i V_a(b) = \|A_{\diamond i}\|_\infty\cdot(a_i-b_i)$. In the optimization language, $V_a(b)$ is also known as the Bregman divergence of the $\|\cdot\|_A^2$ regularizer.

We deduce the following sequence of inequalities:

$$\begin{aligned}
&\big\langle n\alpha_k\xi_k^{(i)}, \mathsf{z}_{k-1}-u\big\rangle = \big\langle n\alpha_k\xi_k^{(i)}, \mathsf{z}_{k-1}-\mathsf{z}_k^{(i)}\big\rangle + \big\langle n\alpha_k\xi_k^{(i)}, \mathsf{z}_k^{(i)}-u\big\rangle\\
&\overset{①}{\le} \big\langle n\alpha_k\xi_k^{(i)}, \mathsf{z}_{k-1}-\mathsf{z}_k^{(i)}\big\rangle + \big\langle -\nabla V_{\mathsf{z}_{k-1}}(\mathsf{z}_k^{(i)}), \mathsf{z}_k^{(i)}-u\big\rangle\\
&\overset{②}{=} \big\langle n\alpha_k\xi_k^{(i)}, \mathsf{z}_{k-1}-\mathsf{z}_k^{(i)}\big\rangle - \frac{1}{2}\|\mathsf{z}_{k-1}-\mathsf{z}_k^{(i)}\|_A^2 + \frac{1}{2}\|\mathsf{z}_{k-1}-u\|_A^2 - \frac{1}{2}\|\mathsf{z}_k^{(i)}-u\|_A^2\\
&\overset{③}{=} n^2\alpha_k^2 L\Big(\big\langle\xi_k^{(i)}, \mathsf{x}_k-\mathsf{y}_k\big\rangle - \frac{L}{2}\|\mathsf{x}_k-\mathsf{y}_k\|_A^2\Big) + \frac{1}{2}\|\mathsf{z}_{k-1}-u\|_A^2 - \frac{1}{2}\|\mathsf{z}_k^{(i)}-u\|_A^2\\
&\le n^2\alpha_k^2 L\cdot\big\langle\xi_k^{(i)}, \mathsf{x}_k-\mathsf{y}_k\big\rangle + \frac{1}{2}\|\mathsf{z}_{k-1}-u\|_A^2 - \frac{1}{2}\|\mathsf{z}_k^{(i)}-u\|_A^2\ .
\end{aligned}$$

Here, ① is due to the minimality of $\mathsf{z}_k^{(i)} = \arg\min_{z\in\Delta}\big\{V_{\mathsf{z}_{k-1}}(z) + \big\langle n\alpha_k\xi_k^{(i)}, z\big\rangle\big\}$, which implies that $\big\langle\nabla V_{\mathsf{z}_{k-1}}(\mathsf{z}_k^{(i)}) + n\alpha\xi_k^{(i)}, u-\mathsf{z}_k^{(i)}\big\rangle \ge 0$ for all $u\in\Delta$. Step ② is due to the "three-point equality" of Bregman divergence (cf. [40]), which can be checked for every coordinate $\ell\in[n]$ as follows:

$$\begin{aligned}
-\nabla_\ell V_{\mathsf{z}_{k-1}}(\mathsf{z}_k^{(i)})\cdot(\mathsf{z}_{k,\ell}^{(i)}-u_\ell) &= \|A_{\diamond i}\|_\infty(\mathsf{z}_{k-1,\ell}-\mathsf{z}_{k,\ell}^{(i)})\cdot(\mathsf{z}_{k,\ell}^{(i)}-u_\ell)\\
&= \|A_{\diamond i}\|_\infty\Big(-\frac{1}{2}(\mathsf{z}_{k-1,\ell}-\mathsf{z}_{k,\ell}^{(i)})^2 + \frac{1}{2}(u_\ell-\mathsf{z}_{k-1,\ell})^2 - \frac{1}{2}(\mathsf{z}_{k,\ell}^{(i)}-u_\ell)^2\Big)\ .
\end{aligned}$$

③ is by our choice of y_k which satisfies that $\mathsf{z}_{k-1}-\mathsf{z}_k^{(i)} = n\alpha_k L(\mathsf{x}_k-\mathsf{y}_k^{(i)})$. □

Proposition 6.13. *If* $\mathsf{z}_{k-1}\in\Delta$, *the minimizer* $z = \arg\min_{z\in\Delta}\big\{\frac{1}{2}\|z-\mathsf{z}_{k-1}\|_A^2 + \langle\delta\mathbf{e}_i, z\rangle\big\}$ *for any scalar* $\delta\in\mathbb{R}$ *and basis vector* $\mathbf{e}_i$ *can be computed as follows:*

1. $z\leftarrow\mathsf{z}_{k-1}$.
2. $z_i\leftarrow z_i-\delta/\|A_{\diamond i}\|_\infty$.
3. *If* $z_i<0$, *then* $z_i\leftarrow 0$; *if* $z_i>1/\|A_{\diamond i}\|_\infty$, $z_i\leftarrow 1/\|A_{\diamond i}\|_\infty$.
4. *Return* z.

Proof of Proposition 6.13. Let us denote by z the returned value of the described procedure, and $g(u)\overset{\text{def}}{=}\frac{1}{2}\|u-\mathsf{z}_{k-1}\|_A^2 + \langle\delta\mathbf{e}_i, u\rangle$. Since Δ is a convex body and $g(\cdot)$

is convex, to show $z = \arg\min_{z\in\Delta}\{g(z)\}$, it suffices for us to prove that for every $u \in \Delta$, $\langle \nabla g(z), u - z\rangle \geq 0$. Since the gradient $\nabla g(z)$ can be written explicitly, this is equivalent to

$$\delta(u_i - z_i) + \sum_{\ell=1}^{n} \|A_{\diamond\ell}\|_\infty \cdot \big(z_\ell - \mathsf{z}_{k-1,\ell}\big) \cdot (u_\ell - z_\ell) \geq 0 \ .$$

However, since $z_\ell = \mathsf{z}_{k-1,\ell}$ for every $\ell \neq i$, this is equivalent to

$$\Big(\delta + \|A_{\diamond i}\|_\infty \cdot \big(z_i - \mathsf{z}_{k-1,i}\big)\Big) \cdot (u_i - z_i) \geq 0 \ .$$

There are three possibilities here. If $z_i = \mathsf{z}_{k-1,i} - \delta/\|A_{\diamond i}\|_\infty$ then the left-hand side is zero and we are done. Otherwise, if $z_i > \mathsf{z}_{k-1,i} - \delta/\|A_{\diamond i}\|_\infty$, then it must satisfy that $z_i = 0$; in such a case the left-hand side is the multiplication of two non-negatives, and therefore non-positive. If $z_i < \mathsf{z}_{k-1,i} - \delta/\|A_{\diamond i}\|_\infty$, then it must satisfy that $z_i = 1/\|A_{\diamond i}\|_\infty$; in such a case the left-hand side is the multiplication of two non-positives, and therefore non-positive. □

Lemma 6.16. $\big\langle n\alpha_k\eta_k^{(i)}, \mathsf{z}_{k-1} - u\big\rangle + n^2\alpha_k^2 L \cdot \big\langle \xi_k^{(i)}, \mathsf{x}_k - \mathsf{y}_k^{(i)}\big\rangle \leq 3n\alpha_k L \cdot (f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}))$.

Proof. Now there are three possibilities:

- If $\eta_k^{(i)} = 0$, then we must have $\xi_{k,i}^{(i)} = \nabla_i f_\mu(\mathsf{x}_k) \in [-1, 1]$, and Lemma 6.15 immediately implies

$$\begin{aligned}&\big\langle n\alpha_k\eta_k^{(i)}, \mathsf{z}_{k-1} - u\big\rangle + n^2\alpha_k^2 L \cdot \big\langle \xi_k^{(i)}, \mathsf{x}_k - \mathsf{y}_k^{(i)}\big\rangle \\ &\qquad = n^2\alpha_k^2 L \cdot \big\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)}\big\rangle \leq 2n^2\alpha_k^2 L \cdot (f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}))\end{aligned}$$

- If $\eta_k^{(i)} > 0$ and $\mathsf{z}_{k,i}^{(i)} > 0$, then we precisely have $\mathsf{z}_{k,i}^{(i)} = \mathsf{z}_{k-1,i} - \frac{n\alpha_k}{\|A_{\diamond i}\|_\infty}$ (see Proposition 6.13), and accordingly $\mathsf{y}_{k,i}^{(i)} = \mathsf{x}_{k,i} - \frac{1}{L\|A_{\diamond i}\|_\infty} < \mathsf{x}_{k,i}$. In this case,

$$\begin{aligned}&\big\langle n\alpha_k\eta_k^{(i)}, \mathsf{z}_{k-1} - u\big\rangle + n^2\alpha_k^2 L \cdot \big\langle \xi_k^{(i)}, \mathsf{x}_k - \mathsf{y}_k^{(i)}\big\rangle \\ &\overset{①}{\leq} n\alpha_k \cdot \nabla f_\mu(\mathsf{x}_k) \cdot \frac{1}{\|A_{\diamond i}\|_\infty} + n^2\alpha_k^2 L \cdot \big\langle \xi_k^{(i)}, \mathsf{x}_k - \mathsf{y}_k^{(i)}\big\rangle \\ &\overset{②}{<} n\alpha_k \cdot \nabla f_\mu(\mathsf{x}_k) \cdot \frac{1}{\|A_{\diamond i}\|_\infty} + n^2\alpha_k^2 L \cdot \big\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)}\big\rangle \\ &\overset{③}{=} n\alpha_k L \cdot \big\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)}\big\rangle + n^2\alpha_k^2 L \cdot \big\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)}\big\rangle \\ &\overset{④}{\leq} \big(2n\alpha_k L + 2n^2\alpha_k^2 L\big) \cdot (f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)})) \ .\end{aligned}$$

  Above, ① follows from the fact that $\mathsf{z}_{k-1} \in \Delta$ and therefore $\mathsf{z}_{k-1,i} \leq \frac{1}{\|A_{\diamond i}\|_\infty}$ by the definition of Δ, and $u \geq 0$; ② follows from the fact that x_k and $\mathsf{y}_k^{(i)}$ are only

different at coordinate i, and $\xi_{k,i}^{(i)} = 1 < \nabla_i f_\mu(\mathsf{x}_k)$ (since $\eta_{k,i}^{(i)} > 0$); ③ follows from the fact that $\mathsf{y}_k^{(i)} = \mathsf{x}_k - \frac{\mathbf{e}_i}{L\|A_{\diamond i}\|_\infty}$; and ④ uses Lemma 6.15.

- If $\eta_k^{(i)} > 0$ and $\mathsf{z}_{k,i}^{(i)} = 0$, then we have

$$\begin{aligned}
&\left\langle n\alpha_k \eta_k^{(i)}, \mathsf{z}_{k-1} - u \right\rangle + n^2\alpha_k^2 L \cdot \left\langle \xi_k^{(i)}, \mathsf{x}_k - \mathsf{y}_k^{(i)} \right\rangle \\
&\overset{①}{\leq} \left(n\alpha_k \nabla f_\mu(\mathsf{x}_k) \cdot \mathsf{z}_{k-1,i}\right) + n^2\alpha_k^2 L \cdot \left\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)} \right\rangle \\
&\overset{②}{=} \left\langle n\alpha_k \nabla f_\mu(\mathsf{x}_k), \mathsf{z}_{k-1} - \mathsf{z}_k^{(i)} \right\rangle + n^2\alpha_k^2 L \cdot \left\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)} \right\rangle \\
&\overset{③}{=} n^2\alpha_k^2 L \cdot \left\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)} \right\rangle + n^2\alpha_k^2 L \cdot \left\langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - \mathsf{y}_k^{(i)} \right\rangle \\
&\overset{④}{\leq} 4n^2\alpha_k^2 L \cdot (f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)})) \ .
\end{aligned}$$

  Above, ① is because $u \geq 0$, $\nabla_i f_\mu(\mathsf{x}_k) = \eta_{k,i}^{(i)} + 1 > \eta_{k,i}^{(i)}$ and $\nabla_i f_\mu(\mathsf{x}_k) > \xi_{k,i}^{(i)}$; ② uses the assumption that $\mathsf{z}_{k,i}^{(i)} = 0$ and the fact that $\mathsf{z}_{k-1,\ell} = \mathsf{z}_{k,\ell}^{(i)}$ for every $\ell \neq i$; ③ is from our choice of y_k which satisfies that $\mathsf{z}_{k-1} - \mathsf{z}_k^{(i)} = n\alpha_k L(\mathsf{x}_k - \mathsf{y}_k^{(i)})$; and ④ uses Lemma 6.15.

Combining the three cases above, and using the fact that $f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) \geq 0$, we conclude that

$$\begin{aligned}
\left\langle n\alpha_k \eta_k^{(i)}, \mathsf{z}_{k-1} - u \right\rangle + n^2\alpha_k^2 L \cdot \left\langle \xi_k^{(i)}, \mathsf{x}_k - \mathsf{y}_k^{(i)} \right\rangle &\leq (2n\alpha_k L + 4n^2\alpha_k^2 L) \cdot (f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)})) \\
&\leq 3n\alpha_k L \cdot (f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)})) \ .
\end{aligned}$$

Above, the last inequality uses our choice of α_k (see Algorithm 4). □

Corollary 6.17. *With probability at least* 9/10, $\texttt{PacLPSolver}(A, x^{\mathsf{start}}, \varepsilon)$ *outputs a* $(1 - O(\varepsilon))$ *approximate solution to the packing LP program. The expected running time is* $O(\frac{\log(nm/\varepsilon)\log(1/\varepsilon)}{\varepsilon} N)$.

Proof. Since for every $x \in \Delta$ it satisfies $f_\mu(x) \geq -(1+\varepsilon)\mathsf{OPT}$ according to Proposition 6.3.b, we obtain that $f_\mu(y_T) + (1+\varepsilon)\mathsf{OPT}$ is a random variable that is non-negative, whose expectation $\mathbb{E}[f_\mu(y_T) + (1+\varepsilon)\mathsf{OPT}] \leq 4\varepsilon$. By Markov bound, with at least probability 9/10, we obtain some y_T satisfying $f_\mu(y_T) \leq -(1 - O(\varepsilon))\mathsf{OPT}$, which yields some $(1 - O(\varepsilon))$ approximate solution according to Proposition 6.3.c.

The running time follows from our efficient implementation in Section 6.F. □

6.C Missing Proofs for Section 6.5

Proposition 6.19.

(a) $\mathsf{OPT} \in [1, m]$.

(b) $f_\mu(u^*) \leq (1+\varepsilon)\mathsf{OPT}$ *for* $u^* \overset{\text{def}}{=} (1 + \varepsilon/2)x^* \in \Delta$.

(c) $f_\mu(x) \geq (1-\varepsilon)\mathsf{OPT}$ *for every* $x \geq 0$.

(d) *Letting* $x^{\mathsf{start}} = (1+\varepsilon/2) \cdot x^\sharp + (\frac{1}{n}, \ldots, \frac{1}{n})$, *we have* $\mathbb{1}^T x^{\mathsf{start}} \leq 2\mathsf{OPT}'$ *and* $f_\mu(x^{\mathsf{start}}) \leq 4\mathsf{OPT}$.

(e) *For any* $x \geq 0$ *satisfying* $f_\mu(x) \leq 2\mathsf{OPT}$, *we must have* $Ax \geq (1-\varepsilon)\mathbb{1}$.

(f) *If* $x \geq 0$ *satisfies* $f_\mu(x) \leq (1+O(\varepsilon))\mathsf{OPT}$, *then* $\frac{1}{1-\varepsilon}x$ *is a* $(1+O(\varepsilon))$*-approximate solution to the covering LP.*

(g) *The gradient of* $f_\mu(x)$ *can be written as*

$$\nabla f_\mu(x) = \mathbb{1} - A^T \mathsf{p}(x) \quad \textit{where} \quad \mathsf{p}_j(x) \stackrel{\text{def}}{=} \exp^{\frac{1}{\mu}(1-(Ax)_j)}$$

Proof.

(a) Suppose that j^* is the row that achieves the smallest infinite norm $\|A_{j^\diamond}\|_\infty$ over all rows. Then, for any solution $x \in \mathbb{R}^n_{\geq 0}$ satisfying $\langle A_{\diamond j^*}, x\rangle \geq 1$, we must have $\mathbb{1}^T x \geq 1/\|A_{\diamond j^*}\|_\infty = 1$.

On the other hand, we can construct a feasible solution x as follows. Initialize $x = 0$, and then for each row j, let us find the coordinate i that maximizes the value of A_{ij} among all columns i. Then, we increase x_i by $1/A_{ij} = 1/\|A_{j^\diamond}\|_\infty$. After we have exhausted all the m rows, we arrive at some $x \geq 0$ satisfying $Ax \geq \mathbb{1}$ as well as $\mathbb{1}^T x = \sum_j 1/\|A_{j^\diamond}\|_\infty \leq m$.

(b) We have $\mathbb{1}^T u^* = (1+\varepsilon/2)\mathsf{OPT}$ by the definition of OPT. Also, from the feasibility constraint $Ax^* \geq \mathbb{1}$ in the covering LP, we have $Au^* - \mathbb{1} \geq \varepsilon/2 \cdot \mathbb{1}$, and can compute $f_\mu(u^*)$ as follows:

$$\begin{aligned} f_\mu(u^*) = \mu \sum_j \exp^{\frac{1}{\mu}(1-(Au^*)_j)} + \mathbb{1}^T u^* &\leq \mu \sum_j \exp^{\frac{-\varepsilon/2}{\mu}} + (1+\varepsilon/2)\mathsf{OPT} \\ &\leq \frac{\mu m}{(nm)^2} + (1+\varepsilon/2)\mathsf{OPT} \leq (1+\varepsilon)\mathsf{OPT} \ . \end{aligned}$$

(c) Suppose towards contradiction that $f_\mu(x) < (1-\varepsilon)\mathsf{OPT}$. Since $f_\mu(x) < \mathsf{OPT} \leq m$, we must have that for every $j \in [m]$, it satisfies that $\exp^{\frac{1}{\mu}(1-(Ax)_j)} \leq f_\mu(x)/\mu \leq m/\mu$. This further implies $(Ax)_j \geq 1-\varepsilon$ by the definition of μ. In other words, $Ax \geq (1-\varepsilon)\mathbb{1}$. By the definition of OPT, we must then have $\mathbb{1}^T x \geq (1-\varepsilon)\mathsf{OPT}$, finishing the proof that $f_\mu(x) \geq \mathbb{1}^T x \geq (1-\varepsilon)\mathsf{OPT}$, giving a contradiction.

(d) Using the fact that $Ax^{\mathsf{start}} - \mathbb{1} \geq (1+\varepsilon/2)Ax^\sharp - \mathbb{1} \geq \varepsilon/2 \cdot \mathbb{1}$, we compute $f_\mu(x^{\mathsf{start}})$ as follows:

$$f_\mu(x^{\mathsf{start}}) = \mu \sum_j \exp^{\frac{1}{\mu}(1-(Ax^{\mathsf{start}})_j)} + \mathbb{1}^T x^{\mathsf{start}} \leq \mu \sum_j \exp^{\frac{-\varepsilon/2}{\mu}} + 2\mathsf{OPT} + 1$$

$$\leq \frac{\mu m}{(nm)^2} + 3\mathsf{OPT} < 4\mathsf{OPT} \ .$$

Also, we have $\mathbb{1}^T x^{\mathsf{start}} \leq (1+\varepsilon/2)\mathsf{OPT}' + 1 \leq 2\mathsf{OPT}'$.

(e) To show $Ax \geq (1-\varepsilon)\mathbb{1}$, we can assume that $v = \max_j(1-(Ax)_j) > \varepsilon$ because otherwise we are done. Under this definition, we have

$$f_\mu(x) \geq \mu \exp^{\frac{v}{\mu}} = \mu\Big((\frac{nm}{\varepsilon})^4\Big)^{v/\varepsilon} \geq \frac{\varepsilon}{4\log(nm)}(\frac{nm}{\varepsilon})^4 \gg 2\mathsf{OPT} \ ,$$

contradicting to our assumption that $f_\mu(x) \leq 2\mathsf{OPT}$. Therefore, we must have $v \leq \varepsilon$, that is, $Ax \geq (1-\varepsilon)\mathbb{1}$.

(f) For any x satisfying $f_\mu(x) \leq (1+O(\varepsilon))\mathsf{OPT} \leq 2\mathsf{OPT}$, owing to Proposition 6.19.e, we first have that x is approximately feasible, i.e., $Ax \geq (1-\varepsilon)\mathbb{1}$. Next, because $\mathbb{1}^T x \leq f_\mu(x) \leq (1+O(\varepsilon))\mathsf{OPT}$, we know that x yields an objective $\mathbb{1}^T x \leq (1+O(\varepsilon))\mathsf{OPT}$. Letting $x' = \frac{1}{1-\varepsilon}x$, we both have that x' is feasible (i.e., $Ax' \geq \mathbb{1}$), and x' has an objective $\mathbb{1}^T x'$ at most $(1+O(\varepsilon))\mathsf{OPT}$.

(g) Straightforward by some simple computation. □

6.D Missing Proofs for Section 6.6

Lemma 6.24.

$$\begin{aligned} f_\mu(\mathsf{x}_k) - f_\mu(u^*) &\leq \langle \mathbb{1} - A^T\mathsf{p}(\mathsf{x}_k), \mathsf{x}_k - u^*\rangle + \langle \tilde{A}^T\mathsf{p}(\mathsf{x}_k) - A^T\mathsf{p}(\mathsf{x}_k), u^*\rangle + \varepsilon\mathsf{OPT} \\ &= \langle \nabla f_\mu(\mathsf{x}_k), \mathsf{x}_k - u^*\rangle + \langle \tilde{A}^T\mathsf{p}(\mathsf{x}_k) - A^T\mathsf{p}(\mathsf{x}_k), u^*\rangle + \varepsilon\mathsf{OPT} \end{aligned}$$

Proof.

$$\begin{aligned} f_\mu(\mathsf{x}_k) - f_\mu(u^*) &= \mu\sum_{j=1}^m \Big(\exp^{\frac{1}{\mu}(1-(A\mathsf{x}_k)_j)} - \exp^{\frac{1}{\mu}(1-(Au^*)_j)} \Big) + \langle \mathbb{1}, \mathsf{x}_k - u^*\rangle \\ &\overset{①}{\leq} \mu\sum_{j=1}^m \Big(\exp^{\frac{1}{\mu}(1-(A\mathsf{x}_k)_j)} - \exp^{\frac{1}{\mu}(1-(\tilde{A}u^*)_j)} \Big) + \langle \mathbb{1}, \mathsf{x}_k - u^*\rangle + \mu\cdot m\cdot \exp^{-1/\mu} \\ &\overset{②}{\leq} \sum_{j=1}^m \exp^{\frac{1}{\mu}(1-(A\mathsf{x}_k)_j)} \cdot \big((\tilde{A}u^*)_j - (A\mathsf{x}_k)_j\big) + \langle \mathbb{1}, \mathsf{x}_k - u^*\rangle + \varepsilon\mathsf{OPT} \\ &= \sum_{j=1}^m \mathsf{p}_j(\mathsf{x}_k) \cdot \big((\tilde{A}u^*)_j - (A\mathsf{x}_k)_j\big) + \langle \mathbb{1}, \mathsf{x}_k - u^*\rangle + \varepsilon\mathsf{OPT} \\ &= \sum_{j=1}^m \mathsf{p}_j(\mathsf{x}_k) \cdot \big((Au^*)_j - (A\mathsf{x}_k)_j\big) + \langle \mathbb{1}, \mathsf{x}_k - u^*\rangle \end{aligned}$$

$$+\sum_{j=1}^{m}\mathsf{p}_j(\mathsf{x}_k)\cdot\big((\tilde{A}u^*)_j-(Au^*)_j\big)+\varepsilon\mathsf{OPT}$$
$$=\langle -A\mathsf{p}(\mathsf{x}_k),\mathsf{x}_k-u^*\rangle+\langle\mathbb{1},\mathsf{x}_k-u^*\rangle+\langle\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k),u^*\rangle+\varepsilon\mathsf{OPT}\ .$$

Above, ① is because if $(Au^*)_j\neq(\tilde{A}u^*)_j$ for some j, then it must satisfy that $(\tilde{A}u^*)_j=2$, and therefore $-\exp^{\frac{1}{\mu}(1-(Au^*)_j)}\leq-\exp^{\frac{1}{\mu}(1-(\tilde{A}u^*)_j)}+\exp^{-1/\mu}$. ② uses the convexity inequality of $e^b-e^a\leq\langle e^b,b-a\rangle$, and the fact that $\mu m\exp^{-1/\mu}\ll\varepsilon\mathsf{OPT}$. □

Lemma 6.26.

$$f_\mu(\mathsf{x}_k)-f_\mu(u^*)\leq\frac{(1-\tau)}{\tau}(f_\mu(\mathsf{y}_{k-1})-f_\mu(\mathsf{x}_k))+\mathbb{E}_i\Big[\langle n\xi_k^{(i)},\mathsf{z}_{k-1}-u^*\rangle\Big]$$
$$+\mathbb{E}_i\Big[\langle n\tilde{\eta}_k^{(i)},-u^*\rangle\Big]+\varepsilon\mathsf{OPT}\ .$$

Proof.

$$\begin{aligned}
&\big(f_\mu(\mathsf{x}_k)-f_\mu(u^*)\big)-\varepsilon\mathsf{OPT}\\
&\overset{①}{\leq}\langle\nabla f_\mu(\mathsf{x}_k),\mathsf{x}_k-u^*\rangle+\langle\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k),u^*\rangle\\
&=\langle\nabla f_\mu(\mathsf{x}_k),\mathsf{x}_k-\mathsf{z}_{k-1}\rangle+\langle\nabla f_\mu(\mathsf{x}_k),\mathsf{z}_{k-1}-u^*\rangle+\langle\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k),u^*\rangle\\
&\overset{②}{=}\frac{(1-\tau)}{\tau}\langle\nabla f_\mu(\mathsf{x}_k),\mathsf{y}_{k-1}-\mathsf{x}_k\rangle+\langle\nabla f_\mu(\mathsf{x}_k),\mathsf{z}_{k-1}-u^*\rangle+\langle\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k),u^*\rangle\\
&\overset{③}{\leq}\frac{(1-\tau)}{\tau}(f_\mu(\mathsf{y}_{k-1})-f_\mu(\mathsf{x}_k))+\langle\nabla f_\mu(\mathsf{x}_k),\mathsf{z}_{k-1}-u^*\rangle+\langle\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k),u^*\rangle\\
&=\frac{(1-\tau)}{\tau}(f_\mu(\mathsf{y}_{k-1})-f_\mu(\mathsf{x}_k))+\langle\xi_k+\eta_k,\mathsf{z}_{k-1}-u^*\rangle+\langle\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k),u^*\rangle\\
&\overset{④}{\leq}\frac{(1-\tau)}{\tau}(f_\mu(\mathsf{y}_{k-1})-f_\mu(\mathsf{x}_k))+\langle\xi_k,\mathsf{z}_{k-1}-u^*\rangle+\langle\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k)-\eta_k,u^*\rangle\\
&\overset{⑤}{\leq}\frac{(1-\tau)}{\tau}(f_\mu(\mathsf{y}_{k-1})-f_\mu(\mathsf{x}_k))+\langle\xi_k,\mathsf{z}_{k-1}-u^*\rangle+\langle-\tilde{\eta}_k,u^*\rangle\\
&=\frac{(1-\tau)}{\tau}(f_\mu(\mathsf{y}_{k-1})-f_\mu(\mathsf{x}_k))+\mathbb{E}_i\Big[\langle n\xi_k^{(i)},\mathsf{z}_{k-1}-u^*\rangle+\langle-n\tilde{\eta}_k^{(i)},u^*\rangle\Big]\ .
\end{aligned}$$

Above, ① is due to Lemma 6.24. ② is because $\mathsf{x}_k=\tau\mathsf{z}_{k-1}+(1-\tau)\mathsf{y}_{k-1}$, which implies that $\tau(\mathsf{x}_k-\mathsf{z}_{k-1})=(1-\tau)(\mathsf{y}_{k-1}-\mathsf{x}_k)$. ③ is by the convexity of $f_\mu(\cdot)$. ④ is because $\langle\eta_k,\mathsf{z}_{k-1}\rangle\leq 0$, since $\eta_k\leq 0$ while $\mathsf{z}_{k-1}\geq 0$.

⑤ needs some careful justification: for every $i\notin B_k$, we have $(\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k))_i-\eta_{k,i}\leq 0-0=-\tilde{\eta}_{k,i}$; for every $i\in B_k$, we have

$$\begin{aligned}
(\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k))_i-\eta_{k,i}&=(\tilde{A}^T\mathsf{p}(\mathsf{x}_k)-A^T\mathsf{p}(\mathsf{x}_k))_i-\big((1+\beta)-(A^T\mathsf{p}(\mathsf{x}_k))_i\big)\\
&=-\big((1+\beta)-(\tilde{A}^T\mathsf{p}(\mathsf{x}_k))_i\big)=-\tilde{\eta}_{k,i}\ ,
\end{aligned}$$

where the two equalities follow from the definitions of $\eta_{k,i}$ and $\tilde{\eta}_{k,i}$ (see Definition 6.25).

□

Lemma 6.27. *Denoting by* $\gamma \stackrel{\text{def}}{=} 2\alpha_T n$*, we have*

$$\mathbb{E}_i\big[\alpha_k\big\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - u^*\big\rangle\big] \le V_{\mathsf{z}_{k-1}}\big(\frac{u^*}{1+\gamma}\big) - \mathbb{E}_i\Big[V_{\mathsf{z}_k^{(i)}}\big(\frac{u^*}{1+\gamma}\big)\Big] + 12\mathsf{OPT}\cdot\gamma\alpha_k\beta \ .$$

Proof. Define $w(x) \stackrel{\text{def}}{=} \sum_i x_i\log(x_i) - x_i$ and accordingly, $V_x(y) = w(y) - \langle w'(x), y - x\rangle - w(x) = \sum_i y_i \log\frac{y_i}{x_i} + x_i - y_i$. We first compute using the classical analysis of mirror descent step as follows:

$$\begin{aligned}
&\gamma\alpha_k\big\langle n\xi_k^{(i)}, \mathsf{z}_{k-1}\big\rangle + \alpha_k\big\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - u^*\big\rangle \\
&= (1+\gamma)\alpha_k\Big\langle n\xi_k^{(i)}, \mathsf{z}_k^{(i)} - \frac{u^*}{1+\gamma}\Big\rangle + (1+\gamma)\alpha_k\big\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - \mathsf{z}_k^{(i)}\big\rangle \\
&\stackrel{①}{\le} \Big\langle w'(\mathsf{z}_{k-1}) - w'(\mathsf{z}_k^{(i)}), \mathsf{z}_k^{(i)} - \frac{u^*}{1+\gamma}\Big\rangle + (1+\gamma)\alpha_k\big\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - \mathsf{z}_k^{(i)}\big\rangle \\
&= \Big(w\big(\frac{u^*}{1+\gamma}\big) - w(\mathsf{z}_{k-1}) - \Big\langle w'(\mathsf{z}_{k-1}), \frac{u^*}{1+\gamma} - \mathsf{z}_{k-1}\Big\rangle\Big) \\
&\quad - \Big(w\big(\frac{u^*}{1+\gamma}\big) - w(\mathsf{z}_k^{(i)}) - \Big\langle w'(\mathsf{z}_k^{(i)}), \frac{u^*}{1+\gamma} - \mathsf{z}_k^{(i)}\Big\rangle\Big) \\
&\quad + \Big(w(\mathsf{z}_{k-1}) - w(\mathsf{z}_k^{(i)}) - \big\langle w'(\mathsf{z}_{k-1}), \mathsf{z}_{k-1} - \mathsf{z}_k^{(i)}\big\rangle\Big) + (1+\gamma)\alpha_k\big\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - \mathsf{z}_k^{(i)}\big\rangle \\
&= V_{\mathsf{z}_{k-1}}\big(\frac{u^*}{1+\gamma}\big) - V_{\mathsf{z}_k^{(i)}}\big(\frac{u^*}{1+\gamma}\big) + \boxed{(1+\gamma)\alpha_k\big\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - \mathsf{z}_k^{(i)}\big\rangle - V_{\mathsf{z}_{k-1}}(\mathsf{z}_k^{(i)})} \ . \quad (6.10)
\end{aligned}$$

Above, ① is because $\mathsf{z}_k^{(i)} = \arg\min_{z\in\Delta}\big\{V_{\mathsf{z}_{k-1}}(z) + \langle(1+\gamma)\alpha_k n\xi_k^{(i)}, z\rangle\big\}$, which is equivalent to saying

$$\begin{aligned}
&\forall u\in\Delta, \quad \langle V'_{\mathsf{z}_{k-1}}(\mathsf{z}_k^{(i)}) + (1+\gamma)\alpha_k n\xi_k^{(i)}, u - \mathsf{z}_k^{(i)}\rangle \ge 0 \\
\iff \quad &\forall u\in\Delta, \quad \langle w'(\mathsf{z}_k^{(i)}) - w'(\mathsf{z}_{k-1}) + (1+\gamma)\alpha_k n\xi_k^{(i)}, u - \mathsf{z}_k^{(i)}\rangle \ge 0 \ .
\end{aligned}$$

In particular, we have $\mathbb{1}^T\frac{u^*}{1+\gamma} = \mathbb{1}^T\frac{(1+\varepsilon/2)x^*}{1+\gamma} < 2\mathsf{OPT} \le 2\mathsf{OPT}'$ and therefore substituting $u = \frac{u^*}{1+\gamma} \in \Delta$ into the above inequality we get ①.

Next, we upper bound the term in the box:

$$\begin{aligned}
&(1+\gamma)\alpha_k\langle n\xi_k^{(i)}, \mathsf{z}_{k-1} - \mathsf{z}_k^{(i)}\rangle - V_{\mathsf{z}_{k-1}}(\mathsf{z}_k^{(i)}) \\
&\stackrel{①}{\le} (1+\gamma)\alpha_k n\xi_{k,i}\cdot(\mathsf{z}_{k-1,i} - \mathsf{z}_{k,i}^{(i)}) - \Big(\mathsf{z}_{k,i}^{(i)}\log\frac{\mathsf{z}_{k,i}^{(i)}}{\mathsf{z}_{k-1,i}} + \mathsf{z}_{k-1,i} - \mathsf{z}_{k,i}^{(i)}\Big) \\
&\stackrel{②}{\le} (1+\gamma)\alpha_k n\xi_{k,i}\cdot(\mathsf{z}_{k-1,i} - \mathsf{z}_{k,i}^{(i)}) - \frac{|\mathsf{z}_{k,i}^{(i)} - \mathsf{z}_{k-1,i}|^2}{2\max\{\mathsf{z}_{k,i}^{(i)}, \mathsf{z}_{k-1,i}\}}
\end{aligned}$$

$$\overset{③}{\leq} (1+\gamma)\alpha_k n\xi_{k,i}\cdot(\mathsf{z}_{k-1,i}-\mathsf{z}^{(i)}_{k,i}) - \frac{|\mathsf{z}^{(i)}_{k,i}-\mathsf{z}_{k-1,i}|^2}{4\mathsf{z}_{k-1,i}}$$
$$\overset{④}{\leq} (1+\gamma)^2\mathsf{z}_{k-1,i}\cdot(\alpha_k n\xi_{k,i})^2 \overset{⑤}{\leq} 2\mathsf{z}_{k-1,i}\cdot(\alpha_k n\xi_{k,i})^2 \overset{⑥}{\leq} \mathsf{z}_{k-1,i}\cdot\gamma\alpha_k n|\xi_{k,i}|$$
$$\overset{⑦}{\leq} \mathsf{z}_{k-1,i}\cdot\gamma\alpha_k n\xi_{k,i} + 2\mathsf{z}_{k-1,i}\cdot\gamma\alpha_k n\beta = \gamma\alpha_k\langle n\xi^{(i)}_k, \mathsf{z}_{k-1}\rangle + 2\mathsf{z}_{k-1,i}\cdot\gamma\alpha_k n\beta \ . \quad (6.11)$$

Above, ① uses the fact that for every $i'\neq i$, $\mathsf{z}^{(i)}_{k,i'}\log\frac{\mathsf{z}^{(i)}_{k,i'}}{\mathsf{z}_{k-1,i'}}+\mathsf{z}_{k-1,i'}-\mathsf{z}^{(i)}_{k,i}\geq 0$. ② uses the inequality that for every $a,b>0$, we have $a\log\frac{a}{b}+b-a\geq\frac{(a-b)^2}{2\max\{a,b\}}$.[12] ③ uses the fact that $\mathsf{z}^{(i)}_{k,i}\leq 2\mathsf{z}_{k-1,i}$.[13] ④ uses Cauchy-Shwarz: $ab-b^2/4\leq a^2$. ⑤ uses $(1+\gamma)^2<2$. ⑥ uses $|\xi_{k,i}|\leq 1$ and $\gamma=2\alpha_T n\geq 2\alpha_k n$. ⑦ uses $\xi_{k,i}\geq-\beta$.

Next, we combine (6.10) and (6.11) to conclude that

$$\alpha_k\big\langle n\xi^{(i)}_k, \mathsf{z}_{k-1}-u^*\big\rangle \leq V_{\mathsf{z}_{k-1}}\big(\frac{u^*}{1+\gamma}\big) - V_{\mathsf{z}^{(i)}_k}\big(\frac{u^*}{1+\gamma}\big) + 2\mathsf{z}_{k-1,i}\cdot\gamma\alpha_k n\beta \ .$$

Taking expectation on both sides with respect to i, and using the property that $\mathbb{1}^T\mathsf{z}_{k-1}\leq 3\mathsf{OPT}'\leq 6\mathsf{OPT}$, we obtain that

$$\mathbb{E}_i\big[\alpha_k\big\langle n\xi^{(i)}_k, \mathsf{z}_{k-1}-u^*\big\rangle\big] \leq V_{\mathsf{z}_{k-1}}\big(\frac{u^*}{1+\gamma}\big) - \mathbb{E}_i\Big[V_{\mathsf{z}^{(i)}_k}\big(\frac{u^*}{1+\gamma}\big)\Big] + 12\mathsf{OPT}\cdot\gamma\alpha_k\beta \ . \quad \square$$

Proposition 6.28. *If $\mathsf{z}_{k-1}\in\Delta$ and $\mathsf{z}_{k-1}>0$, the minimizer $z=\arg\min_{z\in\Delta}\big\{V_{\mathsf{z}_{k-1}}(z)+\langle\delta\mathbf{e}_i,z\rangle\big\}$ for any scalar $\delta\in\mathbb{R}$ and basis vector $\mathbf{e}_i$ can be computed as follows:*

1. *$z\leftarrow\mathsf{z}_{k-1}$.*
2. *$z_i\leftarrow z_i\cdot e^{-\delta}$.*
3. *If $\mathbb{1}^T z>2\mathsf{OPT}'$, $z\leftarrow\frac{2\mathsf{OPT}'}{\mathbb{1}^T z}z$.*
4. *Return z.*

Proof. Let us denote by z the returned value of the described procedure, and $g(u)\overset{\text{def}}{=}V_{\mathsf{z}_{k-1}}(u)+\langle\delta\mathbf{e}_i,u\rangle$. Since Δ is a convex body and $g(\cdot)$ is convex, to show $z=\arg\min_{z\in\Delta}\{g(u)\}$, it suffices for us to prove that for every $u\in\Delta$, $\langle\nabla g(z),u-z\rangle\geq 0$. Since the gradient $\nabla g(z)$ can be written explicitly, this is equivalent to

$$\delta(u_i-z_i)+\sum_{\ell=1}^{n}\log\frac{z_\ell}{\mathsf{z}_{k-1,\ell}}\cdot(u_\ell-z_\ell)\geq 0 \ .$$

If the re-scaling in step 3 is not executed, then we have $z_\ell=\mathsf{z}_{k-1,\ell}$ for every $\ell\neq i$,

[12] This inequality in fact corresponds to a local strong convexity property of $w(\cdot)$. We have used this technique in our paper [7] (see Chapter 5).

[13] This is because, our parameter choices ensure that $(1+\gamma)\alpha_k n<1/2\beta$, which further means $-(1+\gamma)\alpha_k n\xi^{(i)}_k\leq 1/2$. As a result, we must have $\mathsf{z}^{(i)}_{k,i}\leq\mathsf{z}_{k-1,i}\cdot e^{0.5}<2\mathsf{z}_{k-1,i}$ (see the explicit definition of the mirror step at Proposition 6.28).

and $z_i = \mathsf{z}_{k-1,i} \cdot e^{-\delta}$; thus, the left-hand side is zero so the above inequality is true for every $u \in \Delta$.

Otherwise, we have $\mathbb{1}^T z = 2\mathsf{OPT}'$ and there exists some constant factor $Z > 1$ such that, $z_\ell = \mathsf{z}_{k-1,\ell}/Z$ for every $\ell \neq i$, and $z_i = \mathsf{z}_{k-1,i} \cdot e^{-\delta}/Z$. In such a case, the left-hand side equals to

$$(u_i - z_i) \cdot (\delta - \delta) + \sum_{\ell=1}^{n} - \log Z \cdot (u_\ell - z_\ell) \ .$$

It is clear at this moment that since $\log Z > 0$ and $\mathbb{1}^T u \leq 2\mathsf{OPT}' = \mathbb{1}^T z$, the above quantity is always non-negative, finishing the proof. □

Lemma 6.29. *For every $i \in [n]$, we have*

(a) $f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) \geq 0$, *and*

(b) $f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) \geq \frac{\mu\beta}{12} \cdot \langle -\tilde{\eta}_k^{(i)}, u^* \rangle$.

Proof of Lemma 6.29 part (a). Since if $i \notin B_k$ is not a large index we have $\mathsf{y}_k^{(i)} = \mathsf{x}_k$ and the claim is trivial, we focus on $i \in B_k$ in the remaining proof. Recall that $\mathsf{y}_k^{(i)} = \mathsf{x}_k + \delta \mathbf{e}_i$ for some $\delta > 0$ defined in Algorithm 5, so we have

$$f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) = \int_{\tau=0}^{\delta} \langle -\nabla f_\mu(\mathsf{x}_k + \tau \mathbf{e}_i), \mathbf{e}_i \rangle d\tau = \int_{\tau=0}^{\delta} \big(\langle A_{\diamond i}, \mathsf{p}(\mathsf{x}_k + \tau \mathbf{e}_i) \rangle - 1 \big) d\tau \ .$$

It is clear that $\langle A_{\diamond i}, \mathsf{p}(\mathsf{x}_k + \tau \mathbf{e}_i) \rangle$ decreases as τ increases, and therefore it suffices to prove that $\langle A_{\diamond i}, \mathsf{p}(\mathsf{x}_k + \delta \mathbf{e}_i) \rangle \geq 1$.

Suppose that the rows of $A_{\diamond i}$ are sorted (for the simplicity of notation) by the increasing order of $A_{j,i}$. Now, by the definition of the algorithm, there exists some $j^* \in [m]$ satisfying that

$$\sum_{j<j^*} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k) < 1 + \beta \quad \text{and} \quad \sum_{j \leq j^*} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k) \geq 1 + \beta \ .$$

Next, by our choice of δ which satisfies $\delta = \frac{\mu\beta}{2A_{j^*,i}} \leq \frac{\mu\beta}{2A_{j,i}}$ for every $j \leq j^*$, we have

$$\mathsf{p}_j(\mathsf{x}_k + \delta \mathbf{e}_i) = \mathsf{p}_j(\mathsf{x}_k) \cdot \exp^{-\frac{A_{j,i}\delta}{\mu}} \geq \mathsf{p}_j(\mathsf{x}_k) \cdot \exp^{-\beta/2} \geq \mathsf{p}_j(\mathsf{x}_k) \cdot (1 - \beta/2) \ ,$$

and as a result,

$$\langle A_{\diamond i}, \mathsf{p}(\mathsf{x}_k + \delta \mathbf{e}_i) \geq \sum_{j \leq j^*} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k + \delta \mathbf{e}_i) \geq (1 - \beta/2) \sum_{j \leq j^*} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k) \geq (1 - \beta/2)(1 + \beta) \geq 1 \ .$$

□

Proof of Lemma 6.29 part (b). Owing to part (a), for every coordinate i such that $\tilde{\eta}_{k,i} \geq 0$, we automatically have $f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) \geq 0$ so the lemma is obvious. Therefore, let us focus only on coordinates i such that $\tilde{\eta}_{k,i} < 0$; these are necessarily large indices $i \in B$. Recall from Definition 6.25 that $\tilde{\eta}_{k,i} = (1+\beta) - (\tilde{A}^T \mathsf{p}(\mathsf{x}_k))_i$, so we have

$$\sum_{j=1}^{m} \tilde{A}_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k) - (1+\beta) > 0 \ .$$

For the simplicity of description, suppose again that the rows of the i-th column is sorted in the non-decreasing order of $A_{j,i}$. That is, $A_{1,i} \leq \cdots A_{m,i}$. The definition of j^* can be simplified as

$$\sum_{j<j^*} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k) < 1+\beta \quad \text{and} \quad \sum_{j\leq j^*} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k) \geq 1+\beta \ .$$

Let $j^\flat \in [m]$ be the row such that

$$\sum_{j<j^\flat} \tilde{A}_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k) < 1+\beta \quad \text{and} \quad \sum_{j\leq j^\flat} \tilde{A}_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k) \geq 1+\beta \ .$$

Note that such a $j^\flat$ must exist because $\sum_{j=1}^m \tilde{A}_{j,i} \cdot \mathsf{p}_j > 1+\beta$. It is clear that $j^\flat \geq j^*$, owing to the definition that $\tilde{A}_{ji} \leq A_{ji}$ for all $i \in [n], j \in [m]$. Defining $\delta^\flat = \frac{\mu\beta}{2A_{j^\flat,i}} \leq \delta$, the objective decrease is lower bounded as

$$\begin{aligned}
f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) &= \int_{\tau=0}^{\delta} \langle -\nabla f_\mu(\mathsf{x}_k + \tau\mathbf{e}_i), \mathbf{e}_i \rangle d\tau = \int_{\tau=0}^{\delta} \big(\langle A_{\diamond i}, \mathsf{p}(\mathsf{x}_k + \tau\mathbf{e}_i) \rangle - 1 \big) d\tau \\
&\geq \int_{\tau=0}^{\delta^\flat} \big(\langle A_{\diamond i}, \mathsf{p}(\mathsf{x}_k + \tau\mathbf{e}_i) \rangle - 1 \big) d\tau \\
&= \underbrace{\int_{\tau=0}^{\delta^\flat} \Big(\sum_{j\leq j^\flat} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k + \tau\mathbf{e}_i) - 1 \Big) d\tau}_{I} + \underbrace{\sum_{j>j^\flat} \int_{\tau=0}^{\delta^\flat} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k + \tau\mathbf{e}_i) d\tau}_{I'}
\end{aligned}$$

where the inequality is because $\delta^\flat \leq \delta$ and $\langle A_{\diamond i}, \mathsf{p}(\mathsf{x}_k + \tau\mathbf{e}_i) \rangle \geq 1$ for all $\tau \leq \delta$ (see the proof of part (a)).

Part *I*. To lower bound I, we use the monotonicity of $\mathsf{p}_j(\cdot)$ and obtain that

$$I = \int_{\tau=0}^{\delta^\flat} \Big(\sum_{j\leq j^\flat} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k + \tau\mathbf{e}_i) - 1 \Big) d\tau \geq \delta^\flat \cdot \Big(\sum_{j\leq j^\flat} A_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k + \delta^\flat\mathbf{e}_i) - 1 \Big) \ .$$

However, our choice of $\delta^\flat = \frac{\mu\beta}{2A_{j^\flat,i}} \le \frac{\mu\beta}{2A_{j,i}}$ for all $j \le j^\flat$ ensures that

$$\sum_{j\le j^\flat} A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k+\delta^\flat \mathbf{e}_i) \ge \sum_{j\le j^\flat} A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k)\cdot \exp^{\frac{-A_{j,i}\cdot\delta^\flat}{\mu}} \ge \sum_{j\le j^\flat} A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k)\cdot(1-\beta/2) \ .$$

Therefore, we obtain that

$$I \ge \delta^\flat\Big((1-\beta/2)\sum_{j\le j^\flat} A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k)-1\Big) \ge \frac{\delta^\flat}{3}\Big(\sum_{j\le j^\flat} A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k)-1\Big) \ ,$$

where the inequality is because $\left(\frac{2}{3}-\frac{\beta}{2}\right)\sum_{j\le j^\flat} A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k) \ge \frac{4-3\beta}{6}\cdot(1+\beta) \ge \frac{2}{3}$ whenever $\beta \le \frac{1}{3}$ (or equivalently, whenever $\varepsilon \le 1/9$).

Now, suppose that $\sum_{j\le j^\flat} \tilde{A}_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k)-(1+\beta) = b\cdot \tilde{A}_{j^\flat,i}\cdot \mathsf{p}_{j^\flat}(\mathsf{x}_k)$ for some $b\in[0,1]$. Note that we can do so by the very definition of $j^\flat$. Then, we must have

$$\begin{aligned}\sum_{j\le j^\flat} A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k)-1 &\ge \sum_{j<j^\flat} \tilde{A}_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k) + A_{j^\flat,i}\cdot \mathsf{p}_{j^\flat}(\mathsf{x}_k)-1\\ &= (1+\beta)-(1-b)\tilde{A}_{j^\flat,i}\cdot \mathsf{p}_{j^\flat}(\mathsf{x}_k)+A_{j^\flat,i}\cdot \mathsf{p}_{j^\flat}-1\\ &\ge \beta + b\cdot A_{j^\flat,i}\cdot \mathsf{p}_{j^\flat}(\mathsf{x}_k) \ .\end{aligned}$$

Therefore, we conclude that

$$\begin{aligned}I &\ge \frac{\delta^\flat}{3}\Big(\sum_{j\le j^\flat} A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k)-1\Big) > \frac{\delta^\flat}{3}\cdot b\cdot A_{j^\flat,i}\cdot \mathsf{p}_{j^\flat}(\mathsf{x}_k) = \frac{\mu\beta}{6\tilde{A}_{j^\flat,i}}\cdot b\cdot \tilde{A}_{j^\flat,i}\cdot \mathsf{p}_{j^\flat}(\mathsf{x}_k)\\ &= \frac{\mu\beta}{6\tilde{A}_{j^\flat,i}}\cdot\Big(\sum_{j\le j^\flat}\tilde{A}_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k)-(1+\beta)\Big) \ge \frac{\mu\beta}{12}\cdot u_i^*\cdot\Big(\sum_{j\le j^\flat}\tilde{A}_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k)-(1+\beta)\Big) \ .\end{aligned}$$

Above, the last inequality is because $u_i^*\cdot \tilde{A}_{j^\flat,i} \le \langle \tilde{A}_{j^\flat\diamond}, u^*\rangle \le 2$ by our definition of the adjusted $\tilde{A}$.

Part *I′*. To lower bound I', consider every $j > j^\flat$ and the integral

$$\int_{\tau=0}^{\delta^\flat} A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k+\tau\mathbf{e}_i)d\tau \ .$$

Note that whenever $\tau \le \frac{\mu\beta}{2A_{j,i}} \le \frac{\mu\beta}{2A_{j^\flat,i}} = \delta^\flat$, we have that $\mathsf{p}_j(\mathsf{x}_k+\tau\mathbf{e}_i) \ge \mathsf{p}_j(\mathsf{x}_k)\cdot e^{-\beta/2} \ge \frac{1}{2}\mathsf{p}_j(\mathsf{x}_k)$. Therefore, the above integral is at least $\frac{\mu\beta}{2A_{j,i}}\cdot A_{j,i}\cdot\frac{1}{2}\mathsf{p}_j(\mathsf{x}_k)$. This implies a lower bound on I':

$$I' \ge \sum_{j>j^\flat}\frac{\mu\beta}{4A_{j,i}}\cdot A_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k) \ge \frac{\mu\beta}{8}\cdot\sum_{j>j^\flat} u_i^*\cdot\tilde{A}_{j,i}\cdot \mathsf{p}_j(\mathsf{x}_k) \ ,$$

where again in the last inequality we have used $u_i^* \cdot \tilde{A}_{j^\flat,i} \leq \langle \tilde{A}_{j^\flat \diamond}, u^* \rangle \leq 2$ by our definition of $\tilde{A}$.

Together. Combining the lower bounds on I and I', we obtain

$$f_\mu(\mathsf{x}_k) - f_\mu(\mathsf{y}_k^{(i)}) \geq I + I' \geq \frac{\mu\beta}{12} \cdot u_i^* \cdot \Big(\sum_{j=1}^{m} \tilde{A}_{j,i} \cdot \mathsf{p}_j(\mathsf{x}_k) - (1+\beta) \Big) = \frac{\mu\beta}{12} \cdot \langle -\tilde{\eta}_k^{(i)}, u^* \rangle \ . \ \square$$

6.E Efficient Implementation of `PacLPSolver`

In this section, we illustrate how to implement each iteration of `PacLPSolver` to run in an expected $O(N/n)$ time. We maintain the following quantities

$$\mathsf{z}_k \in \mathbb{R}^n_{\geq 0}, \quad \mathsf{az}_k \in \mathbb{R}^m_{\geq 0}, \quad \mathsf{y}'_k \in \mathbb{R}^n, \quad \mathsf{ay}_k \in \mathbb{R}^m, \quad B_{k,1}, B_{k,2} \in \mathbb{R}_+$$

throughout the algorithm, so as to ensure the following invariants are always satisfied

$$A\mathsf{z}_k = \mathsf{az}_k \ , \tag{6.12}$$

$$\mathsf{y}_k = B_{k,1} \cdot \mathsf{z}_k + B_{k,2} \cdot \mathsf{y}'_k \ , \qquad A\mathsf{y}_k = B_{k,1} \cdot A\mathsf{z}_k + B_{k,2} \cdot \mathsf{ay}_k \ . \tag{6.13}$$

It is clear that when $k = 0$, letting $\mathsf{az}_k = A\mathsf{z}_0$, $\mathsf{y}'_k = \mathsf{y}_0$, $\mathsf{ay}_k = A\mathsf{y}_0$, $B_{k,1} = 0$, and $B_{k,2} = 1$, we can ensure that all the invariants are satisfied initially. We denote $\|A_{\diamond i}\|_0$ the number of nonzeros elements in vector $A_{\diamond i}$. In each iteration $k = 1, 2, \ldots, T$:

- The step $\mathsf{x}_k = \tau\mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1}$ does not need to be implemented.

- The value $\nabla_i f(\mathsf{x}_k)$ requires the knowledge of $\mathsf{p}_j(\mathsf{x}_k) = \exp^{\frac{1}{\mu}((A\mathsf{x}_k)_j - 1)}$ for each j such that $A_{ij} \neq 0$. Accordingly, we need to know the value

  $$(A\mathsf{x}_k)_j = \tau(A\mathsf{z}_{k-1})_j + (1-\tau)(A\mathsf{y}_{k-1})_j = \big(\tau + (1-\tau)B_{k-1,1}\big)(A\mathsf{z}_{k-1})_j + (1-\tau)B_{k-1,2}\mathsf{ay}_{k-1,j}$$

  for each such j. This can be computed in $O(1)$ time for each j, and $O(\|A_{\diamond i}\|_0)$ time in total.

- Recall that the step $\mathsf{z}_k \leftarrow \arg\min_{z \in \Delta} \left\{ \frac{1}{2}\|z - \mathsf{z}_{k-1}\|_A^2 + \langle n\alpha_k \xi_k^{(i)}, z \rangle \right\}$ can be written as $\mathsf{z}_k = \mathsf{z}_{k-1} + \delta \mathbf{e}_i$ for some $\delta \in \mathbb{R}$ that can be computed in $O(1)$ time (see Proposition 6.13). Observe also that $\mathsf{z}_k = \mathsf{z}_{k-1} + \delta\mathbf{e}_i$ yields $\mathsf{y}_k = \tau\mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1} + \frac{\delta \mathbf{e}_i}{n\alpha_k L}$ due to Line 6 and Line 10 of Algorithm 4. Therefore, we can perform two explicit updates on z_k and az_k as

  $$\mathsf{z}_k \leftarrow \mathsf{z}_{k-1} + \delta\mathbf{e}_i \ , \quad \mathsf{az}_k \leftarrow A\mathsf{z}_{k-1} + \delta A_{\diamond i}$$

and two implicit updates on y_k as

$$B_{k,1} = \tau + (1-\tau)B_{k-1,1} \quad , \quad B_{k,2} = (1-\tau)B_{k-1,2} \ ,$$
$$\mathsf{y}'_k \leftarrow \mathsf{y}'_{k-1} + \delta \mathbf{e}_i \cdot \Big(- \tfrac{B_{k,1}}{B_{k,2}} + \tfrac{1}{n\alpha_k L}\tfrac{1}{B_{k,2}} \Big) \quad , \quad \mathsf{ay}_k \leftarrow \mathsf{ay}_{k-1} + \delta A_{\diamond i} \cdot \Big(- \tfrac{B_{k,1}}{B_{k,2}} + \tfrac{1}{n\alpha_k L}\tfrac{1}{B_{k,2}} \Big)$$

It is not hard to verify that after these updates, we have

$$\begin{aligned}
\mathsf{y}_k &= B_{k,1} \cdot \mathsf{z}_k + B_{k,2} \cdot \mathsf{y}'_k \\
&= B_{k,1} \cdot \big(\mathsf{z}_{k-1} + \delta \mathbf{e}_i\big) + B_{k,2} \cdot \Big(\mathsf{y}'_{k-1} + \delta \mathbf{e}_i \cdot \Big(- \frac{B_{k,1}}{B_{k,2}} + \frac{1}{n\alpha_k L}\frac{1}{B_{k,2}} \Big)\Big) \\
&= B_{k,1} \cdot \mathsf{z}_{k-1} + B_{k,2} \cdot \Big(\mathsf{y}'_{k-1} + \delta \mathbf{e}_i \cdot \Big(\frac{1}{n\alpha_k L}\frac{1}{B_{k,2}} \Big)\Big) \\
&= B_{k,1} \cdot \mathsf{z}_{k-1} + B_{k,2} \cdot \mathsf{y}'_{k-1} + \frac{\delta \mathbf{e}_i}{n\alpha_k L} \\
&= \big(\tau + (1-\tau)B_{k-1,1}\big) \cdot \mathsf{z}_{k-1} + \big((1-\tau)B_{k-1,2}\big) \cdot \mathsf{y}'_{k-1} + \frac{\delta \mathbf{e}_i}{n\alpha_k L} \\
&= \tau \mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1} + \frac{\delta \mathbf{e}_i}{n\alpha_k L} \ .
\end{aligned}$$

One can similarly verify that $A\mathsf{y}_k = B_{k,1} \cdot A\mathsf{z}_k + B_{k,2} \cdot \mathsf{ay}_k$ equals $A\mathsf{y}_k = \tau A\mathsf{z}_{k-1} + (1-\tau)A\mathsf{y}_{k-1} + \frac{\delta A e_i}{n\alpha_k L}$. In sum, these updates are dominated by the updates on $A\mathsf{z}_k$ and ay_k, each costing an $O(\|A_{\diamond i}\|_0)$ running time, and ensure that the invariants in (6.12) and (6.13) are satisfied at iteration k.

In sum, we only need $O(\|A_{\diamond i}\|_0)$ time to perform the updates in `PacLPSolver` for an iteration k if the coordinate i is selected. Therefore, each iteration of `PacLPSolver` can be implemented to run in an expected $O(\mathbb{E}_i[\|A_{\diamond i}\|_0]) = O(N/n)$ time.

6.F Efficient Implementation of `CovLPSolver`

In this section we illustrate how to implement each iteration of `CovLPSolver` to run in an expected $O(N/n)$ time. We maintain the following quantities

$$\mathsf{z}'_k \in \mathbb{R}^n_+, \quad \mathsf{sz}_k \in \mathbb{R}_+, \quad \mathsf{sumz}_k \in \mathbb{R}_+, \quad \mathsf{az}_k \in \mathbb{R}^m_{\geq 0}, \quad \mathsf{y}'_k \in \mathbb{R}^n, \quad \mathsf{ay}_k \in \mathbb{R}^m, \quad B_{k,1}, B_{k,2} \in \mathbb{R}_+$$

throughout the algorithm, so as to maintain the following invariants are always satisfies

$$\mathsf{z}_k = \mathsf{z}'_k/\mathsf{sz}_k, \qquad \mathsf{sumz}_k = \mathbb{1}^T \mathsf{z}'_k, \qquad A\mathsf{z}_k = \mathsf{az}_k/\mathsf{sz}_k, \quad (6.14)$$
$$\mathsf{y}_k = B_{k,1} \cdot \mathsf{z}'_k + B_{k,2} \cdot \mathsf{y}'_k, \qquad A\mathsf{y}_k = B_{k,1} \cdot \mathsf{az}_k + B_{k,2} \cdot \mathsf{ay}_k \ . \quad (6.15)$$

It is clear that when $k = 0$, letting $\mathsf{z}'_k = \mathsf{z}_0$, $\mathsf{sz}_k = 1$, $\mathsf{sumz}_k = \mathbb{1}^T \mathsf{z}_0$, $\mathsf{az}_k = A\mathsf{z}_0$, $\mathsf{y}'_k = \mathsf{y}_0$, $\mathsf{ay}_k = A\mathsf{y}_0$, $B_{k,1} = 0$, and $B_{k,2} = 1$, we can ensure that all the invariants are satisfied

initially.

We denote by $\|A_{\diamond i}\|_0$ the number of nonzero elements in vector $A_{\diamond i}$. In each iteration $k = 1, 2, \ldots, T$:

- The step $\mathsf{x}_k = \tau \mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1}$ does not need to be implemented.
- The value $\mathsf{p}_j(\mathsf{x}_k) = \exp^{\frac{1}{\mu}(1-(A\mathsf{x}_k)_j)}$ for each j only requires the knowledge of

  $$(A\mathsf{x}_k)_j = \tau(A\mathsf{z}_{k-1})_j + (1-\tau)(A\mathsf{y}_{k-1})_j = \big(\tau + (1-\tau)B_{k-1,1}\big)\frac{\mathsf{az}_{k-1,j}}{\mathsf{sz}_{k-1}} + (1-\tau)B_{k-1,2}\mathsf{ay}_{k-1,j} \ .$$

  This can be computed in $O(1)$ time.
- The value $\nabla_i f(\mathsf{x}_k)$ requires the knowledge of $\mathsf{p}_j(\mathsf{x}_k)$ for each $j \in [m]$ such that $A_{ij} \neq 0$. Since we have $\|A_{\diamond i}\|_0$ such j's, we can compute $\nabla_i f(\mathsf{x}_k)$ in $O(\|A_{\diamond i}\|_0)$ time.
- Letting $\delta = (1+\gamma)n\alpha_k\xi_{k,i}^{(i)}$, recall that the mirror step $\mathsf{z}_k \leftarrow \arg\min_{z\in\Delta}\big\{V_{\mathsf{z}_{k-1}}(z) + \langle \delta \mathbf{e}_i, z\rangle\big\}$ has a very simple form (see Proposition 6.28): first multiply the i-th coordinate of z_{k-1} by $e^{-\delta}$ and then, if the sum of all coordinates have exceeded $2\mathsf{OPT}'$, scale everything down so as to sum up to $2\mathsf{OPT}'$. This can be implemented as follows: setting $\delta_1 = \mathsf{z}'_{k-1,i}(e^{-\delta}-1)$,

  $$\begin{aligned} &\mathsf{z}'_k \leftarrow \mathsf{z}'_{k-1} + \delta_1 \mathbf{e}_i \quad , \quad \mathsf{az}_k \leftarrow \mathsf{az}_{k-1} + \delta_1 A_{\diamond i} \ , \\ &\mathsf{sumz}_k \leftarrow \mathsf{sumz}_{k-1} + \delta_1 \quad , \quad \mathsf{sz}_k \leftarrow \mathsf{sz}_k \cdot \max\Big\{1, \tfrac{\mathsf{sumz}_k}{\mathsf{sz}_{k-1}\cdot 2\mathsf{OPT}'}\Big\} \ . \end{aligned}$$

  These updates can be implemented to run in $O(\|A_{\diamond i}\|_0)$ time, and they together ensure that the invariants in (6.14) are satisfied at iteration k.
- Recall that the gradient step is of the form $\mathsf{y}_k \leftarrow \mathsf{x}_k + \delta_2 \cdot \mathbf{e}_i$ for some value $\delta_2 \geq 0$. This value δ_2 can be computed in $O(\|A_{\diamond i}\|_0)$ time, since each $\mathsf{p}_j(\mathsf{x}_k)$ can be computed in $O(1)$ time, and we can sort the rows of each column of A by preprocessing.

  Since $\mathsf{y}_k = \mathsf{x}_k + \delta_2 \cdot \mathbf{e}_i = \tau\mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1} + \delta_2\mathbf{e}_i$, we can implement this update by letting

  $$\begin{aligned} &B_{k,1} = \tfrac{\tau}{\mathsf{sz}_{k-1}} + (1-\tau)B_{k-1,1} \quad , \quad B_{k,2} = (1-\tau)B_{k-1,2} \\ &\mathsf{y}'_k \leftarrow \mathsf{y}'_{k-1} + \mathbf{e}_i \cdot \Big(-\tfrac{B_{k,1}\delta_1}{B_{k,2}} + \tfrac{\delta_2}{B_{k,2}}\Big) \quad , \quad \mathsf{ay}_k \leftarrow \mathsf{ay}_{k-1} + A_{\diamond i} \cdot \Big(-\tfrac{B_{k,1}\delta_1}{B_{k,2}} + \tfrac{\delta_2}{B_{k,2}}\Big) \end{aligned}$$

  It is not hard to verify that after these updates, we have

  $$\begin{aligned} \mathsf{y}_k &= B_{k,1}\cdot\mathsf{z}'_k + B_{k,2}\cdot\mathsf{y}'_k = B_{k,1}\cdot\big(\mathsf{z}'_{k-1} + \delta_1\mathbf{e}_i\big) + B_{k,2}\cdot\Big(\mathsf{y}'_{k-1} + \mathbf{e}_i\cdot\Big(-\frac{B_{k,1}\delta_1}{B_{k,2}} + \frac{\delta_2}{B_{k,2}}\Big)\Big) \\ &= B_{k,1}\cdot\mathsf{z}'_{k-1} + B_{k,2}\cdot\big(\mathsf{y}'_{k-1} + \delta_2\mathbf{e}_i/B_{k,2}\big) \end{aligned}$$

$$
\begin{aligned}
&= B_{k,1} \cdot \mathsf{z}'_{k-1} + B_{k,2} \cdot \mathsf{y}'_{k-1} + \delta_2 \mathbf{e}_i \\
&= \big(\frac{\tau}{\mathsf{sz}_{k-1}} + (1-\tau) B_{k-1,1}\big) \cdot \mathsf{z}'_{k-1} + \big((1-\tau) B_{k-1,2}\big) \cdot \mathsf{y}'_{k-1} + \delta_2 \mathbf{e}_i \\
&= \tau \mathsf{z}_{k-1} + (1-\tau) \mathsf{y}_{k-1} + \delta_2 \mathbf{e}_i \ .
\end{aligned}
$$

One can similarly verify that $A\mathsf{y}_k = B_{k,1} \cdot \mathsf{az}_k + B_{k,2} \cdot \mathsf{ay}_k$ equals $A\mathsf{y}_k = \tau A\mathsf{z}_{k-1} + (1-\tau) A\mathsf{y}_{k-1} + \delta_2 A_{\diamond i}$. These updates can be implemented to run in $O(\|A_{\diamond i}\|_0)$ time, and they together ensure that the invariants in (6.15) are satisfied at iteration k.

In sum, we only need $O(\|A_{\diamond i}\|_0)$ time to perform the updates in `CovLPSolver` for an iteration k if the coordinate i is selected. Therefore, each iteration of `CovLPSolver` can be implemented to run in an expected $O(\mathbb{E}_i[\|A_{\diamond i}\|_0]) = O(N/n)$ time.

Algorithm 5 `CovLPSolver`$(A, x^{\mathsf{start}}, \varepsilon)$

Input: $A \in \mathbb{R}^{m\times n}_{\geq 0}, x^{\mathsf{start}} \in \Delta, \varepsilon \in (0, 1/10]$.

Output: $x \in \Delta$.

$\mu \leftarrow \frac{\varepsilon}{4\log(nm/\varepsilon)}$, $\beta \leftarrow \sqrt{\varepsilon}$, $\tau \leftarrow \frac{\mu\beta}{12n}$. ▷ parameters

$T \leftarrow \lceil \frac{1}{\tau}\log(1/\varepsilon)\rceil = O(\frac{\log(nm/\varepsilon)\log(1/\varepsilon)}{\varepsilon^{1.5}}n)$. ▷ number of iterations

$\alpha_0 \leftarrow (1-\tau)^T \frac{\varepsilon}{12n\beta}$ and $\gamma \leftarrow \frac{\varepsilon}{6\beta}$. ▷ so that $\alpha_T = \frac{\varepsilon}{12n\beta}$ and $\gamma = 2\alpha_T n$

$\mathsf{x}_0 = \mathsf{y}_0 = \mathsf{z}_0 \leftarrow x^{\mathsf{start}}$.

for $k \leftarrow 1$ **to** T **do**

  $\alpha_k \leftarrow \frac{1}{1-\tau}\alpha_{k-1}$.

  $\mathsf{x}_k \leftarrow \tau\mathsf{z}_{k-1} + (1-\tau)\mathsf{y}_{k-1}$.

  Randomly select i uniformly at random from $[n]$.

  Define $\xi_k^{(i)}$ to be a vector that is only non-zero at coordinate i, and equals to $\mathbb{T}^{\mathsf{c}}(\nabla_i f_\mu(\mathsf{x}_k))$.

  ▷ recall from (6.8) that $\nabla_i f_\mu(\mathsf{x}_k) = 1 - \sum_{j=1}^m A_{j,i}\exp^{\frac{1}{\mu}(1-(A\mathsf{x}_k)_j)}$

  ▷ recall from Definition 6.20 that $\mathbb{T}^{\mathsf{c}}(v) \stackrel{\text{def}}{=} \begin{cases} v, & v \in [-\beta, 1]; \\ -\beta, & v < -\beta. \end{cases}$

  $\mathsf{z}_k \leftarrow \mathsf{z}_k^{(i)} \stackrel{\text{def}}{=} \arg\min_{z\in\Delta}\left\{V_{\mathsf{z}_{k-1}}(z) + \langle(1+\gamma)n\alpha_k\xi_k^{(i)}, z\rangle\right\}$. ▷ See Proposition 6.28

  if $\nabla_i f_\mu(\mathsf{x}_k) < -\beta$ **then**

    Denote by π the permutation that sorts the entries of $A_{\diamond i}$ into $A_{\pi(1),i} \leq \cdots \leq A_{\pi(m),i}$.

    Pick $j^* \in [m]$ such that $\sum_{j<j^*} A_{\pi(j),i}\cdot \mathsf{p}_{\pi(j)}(\mathsf{x}_k) < 1+\beta$ but $\sum_{j\leq j^*} A_{\pi(j),i}\cdot \mathsf{p}_{\pi(j)}(\mathsf{x}_k) \geq 1+\beta$.

    ▷ Such a $j^* \in [m]$ must exist because $\sum_{j=1}^m A_{ji}\cdot\mathsf{p}_j(\mathsf{x}_k) \geq 1+\beta$.

    $\mathsf{y}_k \leftarrow \mathsf{y}_k^{(i)} \stackrel{\text{def}}{=} \mathsf{x}_k + \delta\cdot\mathbf{e}_i$ where $\delta = \frac{\mu\beta}{2A_{\pi(j^*),i}}$.

  else

    $\mathsf{y}_k \leftarrow \mathsf{y}_k^{(i)} \stackrel{\text{def}}{=} \mathsf{x}_k$.

  end if

end for

return y_T.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Using Optimization to Obtain a Width-Independent, Parallel, Simpler, and Faster Positive SDP Solver

This chapter is based on an unpublished result of the author, and its further edits can be found at:

http://arxiv.org/abs/1507.02259.

We study the design of polylogarithmic depth algorithms for approximately solving packing and covering semidefinite programs (or positive SDPs for short). This is a natural SDP generalization of the well-studied positive LP problem.

Although positive LPs can be solved in polylogarithmic depth while using only $\log^2 n/\varepsilon^3$ parallelizable iterations [7], the best known positive SDP solvers due to Jain and Yao [84] require $\log^{14} n/\varepsilon^{13}$ parallelizable iterations. Several alternative solvers have been proposed to reduce the exponents in the number of iterations [85, 129]. However, the correctness of the convergence analyses in these works has been called into question [129], as they both rely on algebraic monotonicity properties that do not generalize to matrix algebra.

In this paper, we propose a very simple algorithm based on the optimization framework proposed in [7] (see Chapter 5) for LP solvers. Our algorithm only needs $\log^2 n/\varepsilon^3$ iterations, matching that of the best LP solver. To surmount the obstacles encountered by previous approaches, our analysis requires a new matrix inequality that extends Lieb-Thirring's inequality, and a sign-consistent, randomized variant of the gradient truncation technique proposed in [7, 6].

7.1 Introduction

Solvers for linear programs (LPs) and semidefinite programs (SDPs) are important algorithmic tools for many computational tasks, spanning the fields of computer science, operations research, statistics, and applied mathematics. Although polynomial-time generic solvers for LPs and SDPs have been known for a long time, their performance is often unsatisfactory in the big-data scenario.

In the past two decades, a significant amount of attention has been paid towards a special class of LPs and SDPs, known as positive LPs [101] and positive SDPs [89] respectively. At a high level, positive LPs are characterized by non-negative variables and a non-negative constraint matrix; similarly, positive SDPs are described by positive semidefinite (PSD) matrix variables and a family of PSD matrices as constraints. In this paper, we are interested in solving positive SDPs, formally defined as follows.

Positive SDP. Given $m \times m$ PSD matrices $A_1, A_2, \ldots, A_n$, positive SDP (after putting in its standard form) refers to the following pair of SDPs:[1]

$$\text{Packing SDP:} \qquad \max_{x \geq 0} \left\{ \mathbb{1}^T x \,:\, \sum_{i=1}^n x_i A_i \preceq I \right\} \,, \tag{7.1}$$

$$\text{Covering SDP:} \qquad \min_{Y \succeq 0} \left\{ \mathrm{Tr}(Y) \,:\, A_i \bullet Y \geq 1 \;\forall i \in [n] \right\} \,. \tag{7.2}$$

Since the two programs are dual to each other, let us denote by OPT the optimal value to both of them. Also, let x^* be any optimal solution for the packing SDP (7.1). We say that $x \geq 0$ is a $(1-\varepsilon)$-approximation to the packing SDP if $\sum_{i=1}^n x_i A_i \preceq I$ and $\mathbb{1}^T x \geq (1-\varepsilon)\mathsf{OPT}$, and $Y \succeq 0$ a $(1+\varepsilon)$-approximation to the covering SDP if $A_i \bullet Y \geq 1$ for all $i \in [n]$ and $\mathrm{Tr}(Y) \leq (1+\varepsilon)\mathsf{OPT}$.

In this paper, we assume without loss of generality that

$$\min_{i \in [n]} \{ \|A_i\|_{\mathsf{spe}} \} = 1 \qquad \text{where } \|A_i\|_{\mathsf{spe}} \text{ is the spectral norm of } A_i \,,$$

since otherwise one can scale all A_i by a constant factor, and the solution OPT as well as x^* are only affected by this same constant factor. We denote by $\mathbf{A} = (A_1, \ldots, A_n)$.

History. Positive SDP instances have been used to model a large numer of computational problems, such as Max-Cut [89, 78], sparse PCA [78], coloring [78], the ARV relaxation of SparsestCut [77] and BalancedSeparator [11, 126], and many others. Positive SDPs also found application in computational complexity, where they were crucial in establish the $\mathsf{QIP} = \mathsf{PSPACE}$ equivalence [81], as well as in quantum interactive proofs [82] and quantum zero-sum games [83]. In addition, techniques developed in this line of research have also inspired many other important results, most notably regarding spectral graph theory [125, 126, 4].

[1] The most general form of covering SDP can be written as follows. Given $m \times m$ PSD matrices $C, A_1, \ldots, A_n$, and non-negative scalars $b_1, \ldots, b_n$, a general covering SDP is to

minimize $C \bullet Y$ subject to the constraint that $A_i \bullet Y \geq b_i$ for each $i \in [m]$ and $Y \succeq 0$.

It is a simple exercise, but anyways proved in [129, Appendix A], to see that the above general form can be easily translated into our standard form. This is also true for packing SDP.

While there has been a lot of research on the fast approximate solution of positive LPs [101, 131, 24, 165, 113, 32, 25, 118, 47, 17, 115, 10, 92, 166, 7, 6], the more general positive SDP case has lagged somewhat behind. Most known positive SDP solvers [9, 11, 83, 82, 81, 77, 78] demand a parallel running time that is $\mathsf{polylog}(nm/\varepsilon)\cdot \mathsf{poly}(\rho)$ in order to produce a $(1 \pm \varepsilon)$ approximation of the optimal value. In this expression, ρ is a "width" parameter that depends on the *numeric value* of the SDP and that can sometimes be as large as $\mathsf{poly}(n, m)$.

In a seminal work in 1993, Luby and Nisan [101] introduced the first width-independent and polylogarithmic-parallel-time positive LP solver. Based on this breakthrough, in 2011, Jain and Yao [84] proposed the first approximate positive-SDP solver that is *width-independent* and whose parallel running time is only $\mathsf{poly}(\log n, \frac{1}{\varepsilon})$. In fact, their algorithm is a faithful generalization of the positive LP solver of Luby and Nisan [101] to positive SDPs. Although the convergence rate (i.e., number of parallelizable iterations) required by Luby and Nisan's algorithm is only $O(\log^2(nm)/\varepsilon^4)$, the convergence rate of Jain and Yao's is as large as $O(\log^{14}(nm)/\varepsilon^{13})$ (see Table 7.1). This significant loss in the running time stems from the harder task of computing with matrices and in particular by the loss of commutativity in matrix algebra with respect to the vector setting.

The poor theoretical performance of [84] has attracted some researchers to study alternative positive-SDP solvers. Motivated by Young's algorithm [165] for positive LPs, two alternative solvers have been proposed [85, 129]. However, the theoretical convergence of these two new solvers remains unclear, as the correctness of both convergence analyses has been called into question. The issue with the algorithm of [129] is explicitly stated in the latest ArXiv version of that paper [130]. A similar issue has been identified [127, 164] with the proof of [85]. In a nutshell, the proof difficulties in both works arise because Young's algorithm, in its current form, relies on a monotonicity argument. While such monotonicity holds naturally in the vector (i.e., LP) case, it does not generalize to the matrix (i.e. SDP) world. See Section 7.2 for a detailed discussion of this.

As a result, the best parallel running time of width-independent positive SDP solvers remains to be $O(\log^{14}(nm)/\varepsilon^{13})$ due to Jain and Yao [84].

This Paper. In this paper, we present an algorithm $\texttt{PosSDPSolver}(\mathbf{A}, \varepsilon)$ that runs only in $O(\frac{\log n \cdot \log(nm/\varepsilon)}{\varepsilon^3})$ iterations. This matches the best convergence rate of the width-independent parallel positive LP solver [7], and is a significant improvement over the best known width-independent positive SDP solver by Jain and Yao [84]. It is also an improvement over the solvers of [129] and [85], even if their analyses can be fixed. (See Table 7.1.)

Our algorithm is also much simpler than all the previous width-independent positive SDP solvers, as it avoids the use of "phases" and restarts that are required by previous solvers [84, 85, 129]. Our algorithm is simply divided into $O(\frac{\log n \cdot \log(nm/\varepsilon)}{\varepsilon^3})$

Problem	Paper	Parallel Depth Per Iteration	Number of Iterations
p/c LP	[101]	$\log(nm)$	$\log^2(nm)/\varepsilon^4$
p/c LP	[7]	$\log(nm)$	$\log^2(nm)/\varepsilon^3$
p/c SDP	[84]	$\mathsf{polylog}(nm)\cdot\mathsf{poly}(1/\varepsilon)$	$\log^{14}(nm)/\varepsilon^{13}$
p/c SDP	[129, 85]	$\log^2(nm)/\varepsilon$	$\log^2(nm)/\epsilon^4$, in doubt [a]
p/c SDP	**[this paper]**	$\log^2(nm)/\varepsilon$	$\log^2(nm)/\varepsilon^3$

Table 7.1: Comparisons of asymptotic running times among width-independent approximate solvers for positive LPs and SDPs. Notice that each iteration of a SDP solver requires a $1/\varepsilon$-dependance to approximate the matrix exponential using the Johnson-Lindestrauss Lemma [129].

[a] See Section 7.2 for details.

iterations. Starting from some initial vector $x \geq 0$, in each iteration, we compute n matrix exponential computations $A_1 \bullet e^{\Psi}, \ldots A_n \bullet e^{\Psi}$ in parallel for some symmetric matrix Ψ satisfying $\|\Psi\|_{\mathsf{spe}} \leq O(\log(nm)/\varepsilon)$, and then change x_i according to the value of $A_i \bullet e^{\Psi}$. This same algorithm simultaneously produces $1 \pm O(\varepsilon)$ approximate solutions to the packing SDP (7.1) and the covering SDP (7.2),.

We remark here that, as originally put forward by Arora and Kale [11], and then formally established by Peng and Tangwongsan [129], each of our iterations can be implemented to run in $O(\log^2(nm)/\varepsilon)$ parallel time after some simple preprocessing. In fact, such computations are required by all the previous width-independent positive SDP solvers.

Our Techniques. Our algorithm is directly based on the optimization framework of the positive LP solver recently put forward by Allen-Zhu and Orecchia [7] (see Chapter 5). The non-commutativity introduced by matrices creates significant obstascles and technical challenges that have forced us to make both our algorithm and analysis different from [7].

To begin with, just like the result in [7], we interpret the positive SDP problem as a purely optimization question, i.e., to minimize $f(x)$ for some convex function $f = f^{sdp}$ that is an SDP extension over its LP choice f^{lp} proposed in [7]. In each iteration of our algorithm, we compute the coordinate gradient $\nabla_i f(x) \stackrel{\text{def}}{=} A_i \bullet e^{\frac{1}{\mu}(\sum_{i\in[n]} x_i A_i - I)} - 1$ for each $i \in [n]$.

An Old Story. In [7], the authors update each x_i as follows. They first define the truncated gradient by letting ξ_i be essentially $\min\{1, \nabla_i f(x)\}$.[2] Next, update each $x_i \leftarrow x_i \cdot e^{-\alpha\xi_i}$ for some global parameter $\alpha = \Theta(\varepsilon^2/\log(nm)) > 0$.

[2] There is an optimization insight behind why such a truncation is needed and we refer the interested readers to the introduction of [7].

The key idea behind the convergence result of [7] is that, if one changes x according to the rule above, then for each "important" $i \in [n]$ (i.e., coordinates i satisfying $\nabla_i f(x) \notin [-\varepsilon, \varepsilon]$), we have that $\nabla_i f(x)$ is guaranteed to change multiplicatively within a factor of $1 \pm \frac{1}{2}$ as x changes, and therefore the sign of $\nabla_i f(x)$ for each important i remains the same before and after each update. This leads to the conclusion that the objective value $f(x)$ effectively decreases during each iteration.

Unfortunately, this "multiplicative-change" guarantee, which is a crucial component of most width-independent solvers, is false in the SDP setting.

Our New Ideas. In this paper, we make two important observations. First, suppose for a moment that x is updated in a sign-consistent manner: either it non-decreases or it non-increases for all the coordinates. Even under this sign-consistent assumption, $\nabla_i f(x)$ does not necessarily remain of the same sign for each important coordinate i, so the previous analysis of [7] still fails in the SDP setting. However, under this sign-consistencty assumption, we can show that a carefully chosen weighted summation of $\nabla_i f(x)$ does maitain the same sign. This consideration is sufficient to prove that the objective signficantly decreases at every iteration. To show that the weighted summation remains of the same sign, we require a generalization of the Lieb-Thirring inequality. To the best of our knowledge, this is a new matrix inequality, which may be of independent interest. We shall discuss the relation between our generalizaiton of Lieb-Thirring and positive SDPs in Section 7.2.

Finally, to ensure that x is updated in a sign-consistent manner, we introduce randomness as follows. We flip an unbiased coin at each of our iterations, and choose to either update x_i's in a non-decreasing manner (therefore ignoring all coordinates i with $\nabla_i f(x) > 0$), or in a non-increasing manner (therefore ignoring all coordinates i with $\nabla_i f(x) < 0$). Such a random choice can be shown to decrease the objective $f(x)$ well *in expectation*, but adds a lot difficulty to the analysis of the covering SDP. In short, after such randomness is introduced, the old analysis of [7] only gives a solution Y whose expectation $\mathbb{E}[Y]$ is feasible to the covering SDP (7.2): that is, $A_i \bullet \mathbb{E}[Y] \leq 1$ for each $i \in [n]$. Such a result is totally useless because we need $A_i \bullet Y \leq 1$ for each $i \in [n]$, and therefore we need to propose a totally different analysis that bypasses this difficulty (see Section 7.6).

Conclusion. In this paper we show that the positive LP solver by Allen-Zhu and Orecchia [7] (see Chapter 5) can be extended to the SDP setting without any asymptotic loss in the convergence rate.

At a high level, to convert *any* positive LP solver to SDP, one needs to tradeoff between (a) "what is allowed to be changed in the algorithm without hurting its performance" and (b) "what must be changed in order to work with matrix algebra". In this paper, we make use of the optimization framework of [7], which gives us the greatest degree of freedom in (a), and prove a new matrix inequality that gives us a better understanding of (b). Together, these techincal advances lead to a width-

independent, parallel, simpler, and faster solver for positive SDPs.

7.1.1 Roadmap

We introduce our new matrix inequality and discuss about its connection to positive SDP in Section 7.2. Next in Section 7.3 we describe our algorithm `PosSDPSolver`. In Section 7.4, we define an objective $f_\mu(x)$ and relates it to positive SDP. In Section 7.5 and Section 7.6 respectively, we describe the convergece analyses for the packing and the covering SDPs.

7.2 Some False and Some True Inequalities in Matrix Algebra

We denote by $A \bullet B = \mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ the matrix inner product, and by $\|A\|_{\mathsf{spe}}$ the spectral norm of a matrix A. If X is symmetric, we use e^X to denote its matrix exponential. We write $A \succeq 0$ if A is positive semidefinite (PSD), and $A \succeq B$ if $A - B \succeq 0$.

Some False Matrix Inequalities. The following is the SDP version of a fundamental inequality that the positive LP solver of [7] relies on: for every symmetric matrix Ψ and every $i \in [n]$,

$$A_i \bullet e^{\Psi+B} = (1 \pm O(\varepsilon)) \cdot A_i \bullet e^{\Psi} \text{ if } -\varepsilon I \preceq B \preceq \varepsilon I \ . \tag{7.3}$$

Unfortunately, this inequality is *false* in the general SDP case. It is straightforward to check that it holds when all matrices involved are diagonal.

Similarly, here is another SDP inequality, whose LP version is crucial to to many positive LP solvers [165, 24, 25, 17, 166]. It is the following monotonocity statement: for every symmetric matrix Ψ and every $i \in [n]$,

$$A_i \bullet e^{\Psi+B} \geq A_i \bullet e^{\Psi} \text{ if } B \succeq 0 \ .$$

However, this inequality is again *false.*

Unfortunately, these false matrix facts have found their ways in the positive SDP solvers proposed in [129, 85]. It is not clear at this point if these analyses can be fixed [127, 164].[3] Both the inequalities above become true if Ψ and B commute. This is precisely why the aforementioned positive LP solvers are correct.

Our New Approach. In this section, we shall prove that

$$B \bullet e^{\Psi+B} = (1 \pm O(\varepsilon)) \cdot B \bullet e^{\Psi} \text{ as long as } \varepsilon I \succeq B \succeq 0 \text{ or } -\varepsilon I \preceq B \preceq 0. \tag{7.4}$$

This non-trivial matrix inequality holds *even if* B and Ψ are not commutable, and shall become important for our later proofs in Section 7.5.1. We shall prove this by

[3] The ArXiv version [130] of the paper of Peng and Tangwongsan [129] acknowledges the error. The error in the analysis of [85] lies in the proof of Lemma 8, where they use the fact that "$\mathrm{local}_j(x)$ only increases". This is an instantiation of the second false inequality above.

first establishing an interesting extended form of the Lieb-Thirring inequality.

In 1976, Lieb and Thirring [97] proved that for every $A, B \succeq 0$ and every $r \geq 1$, it holds that $\mathrm{Tr}(B^{1/2}A^{1/2}B^{1/2})^r \leq \mathrm{Tr}(B^{r/2}A^{r/2}B^{r/2})$. This inequality is known as the Lieb-Thirring inequality and is famous for its applications in quantum mechanics and differential equations. Very recently, Allen-Zhu, Liao, and Orecchia have connected it to the online matrix optimization problems [4] (see also Chapter 8).

In the special case of $r = 2$, the Lieb-Thirring inequality says that $\mathrm{Tr}(B^{1/2}A^{1/2}B^{1/2})^2 \leq \mathrm{Tr}(BAB)$. In this paper, we establish the following generalization of the Lieb-Thirring inequality, which turns out to be crucial for the convergence analysis of our positive SDP solver. To the best of our knowledge, this inequality has not appeared in the literature.

Lemma 7.1 (Extended Lieb-Thirring Inequality)**.** *Given $A \succ 0$, $B \succeq 0$ and $\alpha \in [0, 1]$, we have*

$$B^{1/2}A^{\alpha}B^{1/2} \bullet B^{1/2}A^{1-\alpha}B^{1/2} \leq \mathrm{Tr}(BAB) \ .$$

Unlike the original proof of Lieb-Thirring inequality which relies on Epstein's concavity theorem, our proof of Lemma 7.1 relies on Lieb's concavity theorem:

Proposition 7.2 (Lieb's concavity theorem)**.** *For all $m \times n$ matrices K, and all q, r such that $0 \leq q \leq 1$ and $0 \leq r \leq 1$, with $q + r \leq 1$, the function $F(A, B) \stackrel{\text{def}}{=} \mathrm{Tr}(K^T A^q K B^r)$ is jointly concave over (A, B), where A (resp. B) is over the set of all $m \times m$ (resp. $n \times n$) positive definite matrices.*

Proof of Lemma 7.1. The inequality is obvious when $\alpha = 0$ or $\alpha = 1$, and therefore we shall assume without loss of generality that $\alpha \in (0, 1)$. In addition, we can assume without loss of generality that B is diagonal: otherwise, one can apply an orthogonal transformation to make B diagonal.

Let us write $A = A^D + A^0$, where A^D is the diagonal part of A, and A^0 is the off-diagonal part of A. Define $A_\lambda \stackrel{\text{def}}{=} A^D + \lambda A^0 = \lambda A + (1 - \lambda)A^D$. It is clear from this definition that $A_\lambda \succeq 0$ for all $\lambda \in [0, 1]$. In fact, we notice that $A \succ 0$ implies A^D is positive in all of its diagonal entries. As a consequence, there exists some constant $\varepsilon > 0$ such that $A_\lambda \succ 0$ even for all $\lambda \in [-\varepsilon, 1]$.

Now, consider two matrix-to-real functions $g(A) \stackrel{\text{def}}{=} B^{1/2}A^{\alpha}B^{1/2} \bullet B^{1/2}A^{1-\alpha}B^{1/2}$ and $h(A) \stackrel{\text{def}}{=} \mathrm{Tr}(BAB)$. Since $g(A) = \mathrm{Tr}(BA^{\alpha}BA^{1-\alpha})$, Lieb's concavity theorem (cf. Proposition 7.2) implies that $g(A)$ is concave in A (over the positive definite cone). In contrast, $h(A)$ is simply a function that is linear in A. Therefore, $R(\lambda) \stackrel{\text{def}}{=} g(A_\lambda) - h(A_\lambda)$ is defined and concave over $\lambda \in [-\varepsilon, 1]$, and Lemma 7.1 is equivalent to saying that $R(1) \leq 0$.

We begin analyzing $R(\lambda)$ by noticing that $R(0) = g(A_0) - h(A_0) = 0$: this is a simple consequence of the fact that B, being a diagonal matrix, commutes with $A_0 = A^D$. Therefore, combined with the concavity of $R(\lambda)$, to prove $R(1) \leq 0$ it

suffices to prove that $R(\lambda)$ is differentiable at $\lambda = 0$ and $R'(0) = 0$.

First of all, $M_1(\lambda) \stackrel{\text{def}}{=} (A_\lambda)^\alpha$ is differentiable at $\lambda = 0$ and its derivative at $\lambda = 0$ has zero diagonal entries. Indeed, using the representation $M_1(\lambda) = \frac{1}{\pi \csc(\alpha\pi)} \cdot \int_0^\infty x^{\alpha-1} \cdot A_\lambda(A_\lambda + xI)^{-1} dx$, one can verify that,

$$\begin{aligned}
&\frac{dM_1(\lambda)}{d\lambda}\Big|_{\lambda=0} \\
&= \frac{1}{\pi \csc(\alpha\pi)} \cdot \int_0^\infty x^{\alpha-1} \cdot \Big(\frac{dA_\lambda}{d\lambda}(A_\lambda + xI)^{-1} - A_\lambda(A_\lambda + xI)^{-1}\frac{dA_\lambda}{d\lambda}(A_\lambda + xI)^{-1}\Big)\Big|_{\lambda=0} dx \\
&= \frac{1}{\pi \csc(\alpha\pi)} \cdot \int_0^\infty x^{\alpha-1} \cdot \Big(A^0(A^D + xI)^{-1} - A^D(A^D + xI)^{-1}A^0(A^D + xI)^{-1}\Big) dx \ .
\end{aligned}$$

Noticing in the above equality A^0 is a matrix with zero diagonal entries, while $(A^D + xI)^{-1}$ and $A^D(A^D + xI)^{-1}$ are both diagonal matrices. Therefore, $M_1'(0)$ is a matrix with zero diagonal entries.

Similarly, defining $M_2(\lambda) \stackrel{\text{def}}{=} (A_\lambda)^{1-\alpha}$ we have that $M_2(\lambda)$ is differentiable at $\lambda = 0$ and $M_2'(0)$ is a matrix with zero diagonal entries.

Finally, we can compute that

$$\begin{aligned}
R'(0) &= \frac{d\big(B^{1/2}(A_\lambda)^\alpha B^{1/2} \bullet B^{1/2}(A_\lambda)^{1-\alpha}B^{1/2}\big)}{d\lambda}\Big|_{\lambda=0} - \frac{d(B^2 \bullet A_\lambda)}{d\lambda}\Big|_{\lambda=0} \\
&= B^{1/2}M_1'(0)B^{1/2} \bullet B^{1/2}(A^D)^{1-\alpha}B^{1/2} + B^{1/2}(A^D)^\alpha B^{1/2} \bullet B^{1/2}M_2'(0)B^{1/2} - B^2 \bullet A^0 \ .
\end{aligned}$$

Clearly, this means $R'(0) = 0$ because $M_1'(0)$, $M_2'(0)$ and A^0 are all matrices with zero diagonal entries, and B and A^D are diagonal matrices. □

Our extended Lieb-Thirring inequality immediately yields the following monotonicity property on matrix exponential, which is a formal statement of (7.4). Its proof is deferred to Appendix 7.A.

Lemma 7.3. *Given PSD matrix A satisfying $\varepsilon I \succeq A \succeq 0$ and symmetric matrix Ψ, define function $f(t) \stackrel{\text{def}}{=} A \bullet e^{\Psi + tA}$ over real values t. Then, $0 \le f'(t) \le \varepsilon A \bullet e^{\Psi+tA} = \varepsilon f(t)$ for all t. As a result:*

(a) $f(t) \le f(0) \cdot e^{\varepsilon t}$ for all $t \ge 0$, and

(b) $f(t) \ge f(0) \cdot e^{\varepsilon t}$ for all $t \le 0$.

7.3 Our Algorithm

Our algorithm $\texttt{PosSDPSolver}(\mathbf{A}, \varepsilon)$ runs only in $T = O(\frac{\log n \cdot \log(nm/\varepsilon)}{\varepsilon^3})$ parallelizable iterations. We iteratively update x so as to maximize $\mathbb{1}^T x$, while keeping the approximate feasibility $\sum_i x_i A_i \preceq (1+\varepsilon)I$. At each iteration k, we compute a feedback vector v so that $v_i = e^{\frac{1}{\mu}(\sum_{i\in[n]} x_i A_i - I)} \bullet A_i - 1 \in [-1, \infty)$, and perform a multiplicative update $x_i \leftarrow x_i \cdot e^{-\alpha \cdot \mathbb{T}(v_i)}$. Here, $\mathbb{T}(\cdot)$ is randomly chosen (for each iteration k) as either $\mathbb{T}_-$ or $\mathbb{T}_+$, defined as follows:

Algorithm 6 `PosSDPSolver`$(\mathbf{A}, \varepsilon)$

Input: $\mathbf{A} = (A_1, \ldots, A_n)$ where each $A_i \in \mathbb{R}^{m \times m}$ is PSD, and $\varepsilon \in (0, 1/10]$.
Output: nonnegative vector $x \in \mathbb{R}^n_{\geq 0}$ and PSD matrix $Y \in \mathbb{R}_{m \times m}$.

$\mu \leftarrow \frac{\varepsilon}{4\log(nm/\varepsilon)}$ and $\alpha \leftarrow \frac{\varepsilon\mu}{4}$. ▷ parameters
$x_i^{(0)} \leftarrow \frac{1-\varepsilon/2}{n\|A_i\|_{\mathsf{spe}}}$ for all $i \in [n]$. ▷ initial vector $x^{(0)}$
$T \leftarrow \frac{8\log(2n)}{\alpha\varepsilon}$. ▷ number of iterations
for $k \leftarrow 0$ **to** $T-1$ **do**
    Randomly choose $\mathbb{T}^{(k)}$ to be either $\mathbb{T}_-$ or $\mathbb{T}_+$, each with probability half.
    for $i \leftarrow 1$ **to** n **do**
        Compute the feedback $v_i \leftarrow e^{\frac{1}{\mu}(\sum_{i\in[n]} x_i A_i - I)} \bullet A_i - 1$
        Perform an update: $x_i^{(k+1)} \leftarrow x_i^{(k)} \cdot e^{-\alpha \cdot \mathbb{T}^{(k)}(v_i)}$.
    end for
end for
return $\frac{x^{(T)}}{1+\varepsilon}$ and $\frac{\bar{Y}}{1-2\varepsilon}$, where $\bar{Y} \stackrel{\text{def}}{=} \sum_{i=0}^{T-1} Y(x^{(k)})$.
▷ recall that $Y(x) \stackrel{\text{def}}{=} e^{\frac{1}{\mu}(\sum_{i\in[n]} x_i A_i - I)}$

Definition 7.4. *The thresholding functions* $\mathbb{T}_-, \mathbb{T}_+ \colon [-1, \infty) \to [-1, 1]$ *are defined as follows*

$$\mathbb{T}_-(v) \stackrel{\text{def}}{=} \begin{cases} 0, & v \in [-\varepsilon, \infty); \\ v, & v \in [-1, -\varepsilon). \end{cases} \qquad \mathbb{T}_+(v) \stackrel{\text{def}}{=} \begin{cases} 0, & v \in [-1, \varepsilon]; \\ v, & v \in (\varepsilon, 1]; \\ 1, & v > 1. \end{cases}$$

Note that if $\mathbb{T} = \mathbb{T}_-$ then the variables of x monotonically non-decreases, and vice versa.

Remark 7.5 (Matrix Exponentials)**.** Matrix exponential computations are required by all width-independent positive SDP solvers, and dominate the complexity of each algorithmic iteration. Like in previous solvers, it is a simple exercise to verify that our entire analysis in this paper continues to hold, though with a worsen constant, if we are only computing the values $v_i = e^{\frac{1}{\mu}(\sum_{i\in[n]} x_i A_i - I)} \bullet A_i$ up to a $1 \pm \varepsilon/2$ multiplicative factor. Therefore, for simplicity's sake, in this paper we assume that the matrix exponentials can be computed exactly. Note that the $1 \pm \varepsilon/2$ approximate computations of $e^{\frac{1}{\mu}(\sum_{i\in[n]} x_i A_i - I)} \bullet A_i$ for *all* $i \in [n]$ can be performed in polylog parallel iterations.[4]

We summarize our theorem as follows.

[4]More precisely, when each $A_i = Q_i Q_i^T$ is presented in its Cholesky decomposition, we have

Theorem 7.6 ([129])**.** *Given an $m \times m$ PSD matrix Φ with p non-zero entries and $\|\Phi\|_{\mathsf{spe}} \leq \kappa$, and given $m \times m$ matrices $\{A_1, \ldots, A_n\}$ in the form of $A_i = Q_i Q_i^T$ where the total non-zero entries across all Q_i is q. Then, there exists an algorithm that computes $e^{\Phi} \bullet A_i$ for all $i \in [n]$ up to a $(1 \pm \varepsilon)$ factor in*

$$O\Big(\max\{\kappa, \log\frac{1}{\varepsilon}\}\log m + \log\log m\Big) \text{ depth} \quad \text{and} \quad O\Big(\frac{1}{\varepsilon^2}\Big(\max\{\kappa, \log\frac{1}{\varepsilon}\}\cdot p + q\Big)\log m\Big) \text{ work}$$

Theorem 7.7 (Positive SDP). *Letting* $(x, Y) = \texttt{PosSDPSolver}(\mathbf{A}, \varepsilon)$*, we have that with at least a constant probability*

- x *is a* $(1 - O(\varepsilon))$*-approximate solution for the packing SDP (7.1),*
- Y *is a* $(1 + O(\varepsilon))$*-approximate solution for the covering SDP (7.2), and*
- *the number of iterations for* PosSDPSolver *is* $T = O(\log n \cdot \log(nm/\varepsilon) \cdot \varepsilon^{-3})$.

If each $A_i = Q_i Q_i^T$ *is preprocessed into its Cholesky decomposition, each iteration can be implemented in* $O(\log^2(nm)/\varepsilon)$ *parallel depth.*

7.4 The Convex Objective

We define the following convex objective for the positive SDP problem. It is completely analogous to its LP variant introduced in [7], and therefore we state its properties without proof.

Definition 7.8. *Letting parameter* $\mu \stackrel{\text{def}}{=} \frac{\varepsilon}{4\log(nm/\varepsilon)}$*, we define the smoothed objective* $f_\mu(x)$ *as*

$$f_\mu(x) \stackrel{\text{def}}{=} \mu \cdot \mathrm{Tr}\big(e^{\frac{1}{\mu}(\sum_{i\in[n]} x_i A_i - I)}\big) - \mathbb{1}^T x \ .$$

We want to study the minimization problem on $f_\mu(x)$ over all $x \geq 0$. This objective $f_\mu(x)$ captures the packing SDP because, on one hand we want to minimize $-\mathbb{1}^T x$ so as to maximize $\mathbb{1}^T x$, and on the other hand the exponential penalty function says if $\sum_{i\in[n]} x_i A_i \preceq (1+\varepsilon) I$ is violated, a large positive penalty is introduced.

Proposition 7.9.

(a) $\mathsf{OPT} \in [1, n]$.

(b) Letting $x = (1 - \varepsilon/2)x^* \geq 0$*, we have* $f_\mu(x) \leq -(1-\varepsilon)\mathsf{OPT}$.

(c) Letting $x^{(0)} \geq 0$ *be such that* $x_i^{(0)} = \frac{1-\varepsilon/2}{n\|A_i\|_{\mathsf{spe}}}$ *for each* $i \in [n]$*, we have* $f_\mu(x^{(0)}) \leq -\frac{1-\varepsilon}{n}$.

(d) For any $x \geq 0$ *satisfying* $f_\mu(x) \leq 0$*, we have* $\sum_{i\in[n]} x_i A_i \preceq (1+\varepsilon) I$ *and thus* $\mathbb{1}^T x \leq (1+\varepsilon)\mathsf{OPT}$.

(e) If $x \geq 0$ *satisfies* $f_\mu(x) \leq -(1 - O(\varepsilon))\mathsf{OPT}$*, then* $\frac{1}{1+\varepsilon} x$ *is a* $(1 - O(\varepsilon))$*-approximate solution for the packing SDP.*

(f) The gradient of $f_\mu(x)$ *can be written as*

$$\nabla f_\mu(x) = (A_1 \bullet Y(x), \ldots, A_n \bullet Y(x)) - \mathbb{1} \quad \text{where} \quad Y(x) \stackrel{\text{def}}{=} e^{\frac{1}{\mu}(\sum_{i\in[n]} x_i A_i - I)} \tag{7.5}$$

Since one can verify that $\|\Phi\|_{\mathsf{spe}} \leq \kappa \stackrel{\text{def}}{=} 1/\mu = O(\log(nm/\varepsilon)/\varepsilon)$ in our case, each iteration of PosSDPSolver can be implemented to run in $O(\log^2(nm)/\varepsilon)$ parallel time. (Here, we can safely assume that $\varepsilon > 1/(nm)^{O(1)}$; if ε is smaller than $1/(nm)^{O(1)}$, one should use for instance Interior Point Method to solve the given SDP instead.)

7.5 Convergence Analysis for Packing SDP

Throughout this paper, we use superscript $x^{(k)}$ to represent vector x at iteration k, and subscript x_i to represent the i-th coordinate of vector x. Our convergence analysis is divided into three steps, and the first step is the main technical difference between this paper and its LP variant [7].

Step I: Gradient Descent. We interpret (see Section 7.5.1 for details) each update $x_i^{(k+1)} \leftarrow x_i^{(k)} \cdot e^{-\alpha \cdot \mathbb{T}^{(k)}(v_i)}$ as a gradient descent step,[5] and show that the objective $f_\mu(x)$ monotonically decreases between consecutive iterations:

Lemma 7.10 (Gradient Descent)**.** *For every iteration $k = 0, \ldots, T-1$ in* `PosSDPSolver`*, the objective $f_\mu(x)$ does not increases: $f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) \geq 0$. Combining this with Proposition 7.9.c, we have $f_\mu(x^{(k)}) \leq 0$ for all k.*

In addition, letting $B^{(k)} \subseteq [n]$ be the set of indices i such that $\nabla_i f_\mu(x^{(k)}) \geq 1$, then

$$f_\mu(x^{(k)}) - \mathbb{E}[f_\mu(x^{(k+1)})] \geq \frac{\alpha}{4} \cdot \textstyle\sum_{i \in B^{(k)}} x_i^{(k)} \cdot \nabla_i f_\mu\big(x^{(k)}\big) \geq 0 \ .$$

Above, the expectation is over the random choice of $\mathbb{T}^{(k)}$ at iteration k.

We remark here that Lemma 7.10 does *not* follow from any classical theory of gradient descent because our objective $f_\mu(x)$ is simply not smooth in the positive orthant. Neither does Lemma 7.10 follow from the so-called "multiplicative Lipschitz gradient property" introduced in [7], because the fundamental property that the work [7] replies on, "$\nabla_i f_\mu(x)$ increases as x decreases, and vice versa", no longer holds in the SDP case. This is also one of the major reasons that the results of [129, 85] fail to produce any theoretical guarantee.

Our proof of Lemma 7.10 crucially relies on two key properties. First, the sign-consistent and random choice of $\mathbb{T}^{(k)}$ ensures that x either only increases or only decreases at a single iteration k. Second, our new matrix inequality introduced in Section 7.2 ensures that "$\nabla_i f_\mu(x)$ increases *in an average sense* as x decreases". We defer the technical proof of Lemma 7.10 to Section 7.5.1.

Step II: Mirror Descent. It is not hard to show, and in fact proven in [7] for a slightly different variant, that each update $x_i^{(k+1)} \leftarrow x_i^{(k)} \cdot e^{-\alpha \cdot \mathbb{T}^{(k)}(v_i)}$ can also be viewed as a mirror-descent step.

A *mirror descent step* in optimization is any step from x to x' that is of the form $x' \leftarrow \arg\min_z \{V_x(z) + \langle \alpha \nabla f(x), z - x \rangle\}$. Here, $\alpha > 0$ is some step length, and $V_x(\tilde{x}) = w(\tilde{x}) - \langle \nabla w(x), \tilde{x} - x \rangle - w(x)$ is the Bregman divergence of some convex *distance generating function* $w(x)$. In this paper, we pick $w(x) \stackrel{\text{def}}{=} \sum_{i \in [n]} x_i \log x_i - x_i$

[5] To be clear, in some literature, the gradient descent is referred only to $x \leftarrow x - c \cdot \nabla f(x)$ for some constant c. In this paper, we adopt the more general notion, and refer it to any step that directly decreases $f(x)$.

to be the generalized entropy function, and accordingly,

$$\text{for every } x, \tilde{x} \geq 0, \quad V_x(\tilde{x}) \stackrel{\text{def}}{=} \sum_{i\in[n]} \left(\tilde{x}_i \log \tfrac{\tilde{x}_i}{x_i} + x_i - \tilde{x}_i\right) \ .$$

The next lemma easily follows from the general theory of mirror descent. Since its proof has essentially appeared in [7, Lemma 3.3], we prove it in Section 7.B.3 only for the sake of completeness.

Lemma 7.11 (Mirror Descent)**.** *Letting $\gamma \in [-1,1]^n$ be defined as $\gamma_i = \mathbb{T}(\nabla_i f_\mu(x^{(k)}))$, we have that for any $u \geq 0$,*

$$\langle \alpha\gamma, x^{(k)} - u\rangle \leq \alpha^2 \mathsf{OPT} + V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u) \ .$$

Step III: Coupling. Finally, as formally argued in Section 7.B.2, the two lemmas above can be naturally combined, yielding the following bound:

Lemma 7.12 (Coupling)**.** *For any $u \geq 0$ and $k = 0, \ldots, T-1$, we have*

$$\begin{aligned}&\alpha(f_\mu(x^{(k)}) - f_\mu(u)) \leq \langle \alpha \nabla f_\mu(x^{(k)}), x^{(k)} - u\rangle \\ &\leq 4(f_\mu(x^{(k)}) - \mathbb{E}[f_\mu(x^{(k+1)})]) + 2\left(V_{x^{(k)}}(u) - \mathbb{E}[V_{x^{(k+1)}}(u)]\right) + \alpha \cdot 2\varepsilon \mathsf{OPT} + \alpha \cdot \varepsilon \mathbb{1}^T u \ .\end{aligned}$$

Above, the expectation is over the random choice of $\mathbb{T}^{(k)}$ at iteration k.

The proof of Lemma 7.12 relies on a decomposition of the gradient $\nabla_i f_\mu(x^{(k)})$ into four components $\nabla_i f_\mu(x^{(k)}) = \xi_i^+ + \xi_i^- + \eta_i + \zeta_i$, where $\xi_i^+ \in [0,1]$, $\xi_i^- \in [-1,0]$, $\eta_i \in [0,\infty)$, and $\zeta_i \in [-\varepsilon, \varepsilon]$. This is a main difference that distinguishes our proof from [7]: we need to decompose the ξ_i part into a positive and a negative terms, and then apply Lemma 7.11 twice.

Putting All Together. By telescoping the inequality in Lemma 7.12, one can obtain the following final theorem for packing SDP. Its proof is only slightly different from that of [7, Theorem 3.5] due to the special treatment of the randomness, and deferred to Section 7.B.4.

Theorem 7.13 (Packing SDP)**.** *For $T \geq \frac{8\log(2n)}{\alpha\varepsilon} = \Omega(\frac{\log n \cdot \log(nm/\varepsilon)}{\varepsilon^3})$, we have that $\mathbb{E}[f_\mu(x^{(T)})] \leq -(1-5\varepsilon)\mathsf{OPT}$. As a consequence, $\texttt{PosSDPSolver}(A,\varepsilon)$ produces an output $x = \frac{x^{(T)}}{1+\varepsilon}$ that is a $(1-O(\varepsilon))$-approximate solution for the packing SDP (7.1) with at least a constant probability.*

7.5.1 The Gradient Descent Lemma

In this subsection we view our update $x^{(k)} \to x^{(k+1)}$ as a gradient-descent step and prove Lemma 7.10. We begin by observing that each x_i is changed by a factor of at most $1 \pm 4\alpha/3$ per iteration:

Fact 7.14. *We always have $x_i^{(k+1)} \in x_i^{(k)} \cdot [1 - 4\alpha/3, 1 + 4\alpha/3]$.*

Proof. We can always write $x_i^{(k+1)} = x_i^{(k)} \cdot e^t$ for some $t \in [-\alpha, \alpha] \subseteq [-1/4, 1/4]$. According to the fact that $e^t \leq 1 + 4t/3$ for $t \in [0, 1/4]$ and $e^t \geq 1 - t \geq 1 - 4t/3$ for

$t \in [-1/4, 0]$, we must have $x_i^{(k+1)} \in x_i^{(k)} \cdot [1 - 4\alpha/3, 1 + 4\alpha/3]$. $\square$

Proof of Lemma 7.10. We prove by induction. Suppose that Lemma 7.10 is true for all indices less than k. This implies, in particular, that $f_\mu(x^{(k)}) \le f_\mu(x^{(k-1)}) \le \cdots \le f_\mu(x^{(0)}) \le 0$.

There are two cases to consider at iteration k: (1) if we choose $\mathbb{T}_-(\cdot)$ and (2) if we choose $\mathbb{T}_+(\cdot)$. Each of them happens with probability $1/2$.

In the first case, that is, if we choose $\mathbb{T}_-(\cdot)$, we have the property that our vector does not decrease: that is, $x_i^{(k+1)} \ge x_i^{(k)}$ for every $i \in [n]$. We compute the objective difference by the standard integral over gradients:

$$\begin{aligned} f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) &= \int_0^1 \Big\langle \nabla f_\mu\big(x^{(k)} + \tau(x^{(k+1)} - x^{(k)})\big), x^{(k)} - x^{(k+1)} \Big\rangle d\tau \\ &= \mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} + \int_0^1 \Big(e^{\frac{1}{\mu}(\sum_{i\in[n]} x_i^{(k)} A_i - I + \tau \sum_{i\in[n]} (x_i^{(k+1)} - x_i^{(k)}) A_i)} \bullet \sum_i (x_i^{(k)} - x_i^{(k+1)}) A_i \Big) d\tau \\ &= \mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - \mu \int_0^1 B \bullet e^{\Psi + \tau B} d\tau \ , \end{aligned} \tag{7.6}$$

where in the last equality we have defined $\Psi \stackrel{\text{def}}{=} \frac{1}{\mu}(\sum_{i\in[n]} x_i^{(k)} A_i - I)$ and $B \stackrel{\text{def}}{=} \frac{1}{\mu} \sum_{i\in[n]} (x_i^{(k+1)} - x_i^{(k)}) A_i \succeq 0$.

Notice that $f_\mu(x^{(k)}) \le 0$ together with Proposition 7.9.d tells us that $\sum_{i\in[n]} x_i^{(k)} A_i \preceq (1+\varepsilon) I$. Combining it with Fact 7.14 we have $\sum_{i\in[n]} \big(x_i^{(k+1)} - x_i^{(k)}\big) A_i \preceq \frac{4\alpha}{3}(1+\varepsilon) I \preceq \frac{5\alpha}{3} I$ and therefore $B \preceq \frac{5\alpha}{3\mu} I = \frac{5\varepsilon}{12} I$. Applying Lemma 7.3.a with $B \preceq \frac{5\varepsilon}{12} I$ to (7.6), we have

$$\begin{aligned} f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) &\ge \mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - \mu \int_0^1 B \bullet e^{\Psi} \cdot e^{5\varepsilon\tau/12} d\tau \\ &\ge \mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - (1 + \varepsilon/4) \mu B \bullet e^{\Psi} \ . \end{aligned}$$

Recall that, for each $i \in [n]$ satisfying $x_i^{(k+1)} \ne x_i^{(k)}$, we must have $e^\Psi \bullet A_i - 1 < -\varepsilon$ by the definition of $\mathbb{T}_-(\cdot)$. Therefore, multiplying both sides by $x_i^{(k+1)} - x_i^{(k)} \ge 0$ and summing up over $i \in [n]$, we obtain

$$\mu B \bullet e^\Psi = e^\Psi \bullet (\sum_{i\in[n]} (x_i^{(k+1)} - x_i^{(k)}) A_i) \le (1 - \varepsilon)(\mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)}) \ .$$

This further implies that (after some careful term rearranging)

$$\begin{aligned} \mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - (1 + \varepsilon/4) \mu B \bullet e^\Psi &\ge \tfrac{3}{4}(\mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - \mu B \bullet e^\Psi) \\ &= \tfrac{3}{4} \langle \nabla f_\mu(x^{(k)}), x^{(k)} - x^{(k+1)} \rangle \ge 0 \ . \end{aligned}$$

Above, the last inequality is again by our definition of $\mathbb{T}_-$: for each $i \in [n]$ satisfying $x_i^{(k)} \ne x_i^{(k+1)}$, it must satisfy that $\nabla_i f_\mu(x^{(k)}) < -\varepsilon$ and $x_i^{(k)} \le x_i^{(k+1)}$. In conclusion, we arrive at the inequality

$$f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) \ge \frac{3}{4} \langle \nabla f_\mu(x^{(k)}), x^{(k)} - x^{(k+1)} \rangle \ge 0 \ .$$

In the case when $\mathbb{T}_+$ is chosen, a symmetric argument (although replacing the use of Lemma 7.3.a with Lemma 7.3.b and using slightly different constants, see Appendix 7.B.1) yields that

$$\begin{aligned} f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) &\geq \frac{2}{3}\langle \nabla f_\mu(x^{(k)}), x^{(k)} - x^{(k+1)} \rangle \\ &\geq \frac{2}{3} \textstyle\sum_{i \in B^{(k)}} \nabla_i f_\mu(x^{(k)}) \cdot (x_i^{(k)} - x_i^{(k+1)}) \ . \end{aligned}$$

Above, the second inequality is because for each $i \in [n]$ satisfying $x_i^{(k)} \neq x_i^{(k+1)}$, it must satisfy that $\nabla_i f_\mu(x^{(k)}) > \varepsilon$ and $x_i^{(k)} \geq x_i^{(k+1)}$. Next, observe that for each coordinate $i \in B^{(k)}$ we have $x_i^{(k+1)} = x_i^{(k)} \cdot e^{-\alpha} \leq (1 - 0.9\alpha) x_i^{(k)}$ for our choice of α. Plugging this into the inequality above, we arrive at the inequality

$$f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) \geq \frac{2}{3} \cdot 0.9\alpha \sum_{i \in B^{(k)}} \nabla_i f_\mu(x^{(k)}) \cdot x_i^{(k)} \geq \frac{\alpha}{2} \sum_{i \in B^{(k)}} \nabla_i f_\mu(x^{(k)}) \cdot x_i^{(k)} \geq 0 \ .$$

Finally, combining the two cases above, we conclude that

$$f_\mu(x^{(k)}) - \mathbb{E}[f_\mu(x^{(k+1)})] \geq \frac{\alpha}{4} \textstyle\sum_{i \in B^{(k)}} \nabla_i f_\mu(x^{(k)}) \cdot x_i^{(k)} \ . \qquad \square$$

7.6 Convergence Analysis for Covering SDP

We have seen in Section 7.5 that a vector $x \geq 0$ satisfying $f_\mu(x) \approx -\mathsf{OPT}$ yields an approximate solution to the packing SDP (7.1). However, this vector x itself gives no information about the solution to the covering SDP (7.2).

In this section, we show that, defining $\bar{Y} \stackrel{\text{def}}{=} \sum_{i=0}^{T-1} Y(x^{(k)})$ where $Y(x) \stackrel{\text{def}}{=} e^{\frac{1}{\mu}(\sum_{i \in [n]} x_i A_i - I)}$, then $\frac{\bar{Y}}{1-2\varepsilon}$ is a $(1+O(\varepsilon))$-approximate solution to the covering SDP (7.2) with at least a constant probability. Therefore, $\texttt{PosSDPSolver}(\mathbf{A}, \varepsilon)$ is an algorithm that simultaneously solves both the primal and the dual side of the positive SDP problem.

Our proof can be divided into two parts. First, using similar proof techniques as in [7], one can show that $\bar{Y}$ satisfies the *approximate optimality*, at least in an expected sense. We prove this lemma below in Appendix 7.C only for the sake of completeness.

Lemma 7.15. *For any $T \geq \frac{8}{\alpha\varepsilon} = \Omega(\frac{\log(nm/\varepsilon)}{\varepsilon^3})$, we have that $\mathbb{E}[\mathrm{Tr}(\bar{Y})] \leq (1+7\varepsilon)\mathsf{OPT}$.*

In the second part, we wish to show that $\bar{Y}$ satisfies the *approximate feasibility* as well, that is, $A_i \bullet \bar{Y} \leq 1 + O(\varepsilon)$ for all $i \in [n]$. However, we encounter two difficulties:

- First, a similar analysis as in [7] would only imply that the expected matrix $\mathbb{E}[\bar{Y}]$ satisfies such approximate feasibility, rather than $\bar{Y}$. By Markov's inequality, this only suggests that for each (rather than for all) $i \in [n]$, $A_i \bullet \bar{Y} \leq 1 + O(\varepsilon)$ holds with constant probability.[6]

[6]Previously, the first and third authors of this paper have tried to bypass this difficulty using a dual smoothed objective in the LP case [6] (see Chapter 6). However, their analysis is more involved and loses a factor of $\varepsilon^{0.5}$ in the running time.

- Second, the analysis in [7] does not directly imply that $\bar{Y}$ is approximately feasible. Instead, one has to modify $\bar{Y}$ in a non-trivial manner which is very unpleasant in practice.

Due to the above difficulties, we propose in this paper a fundamentally different, yet much simpler analysis for proving the approximate feasibility. This is deferred to Appendix 7.C.

Lemma 7.16. *For any $T \geq \frac{8}{\alpha\varepsilon}$, with probability at least $1-\frac{\varepsilon}{100}$ we have $A_i \bullet \bar{Y} \geq 1-2\varepsilon$ for all $i \in [n]$.*

It is now easy to see that Lemma 7.15 and Lemma 7.16 together imply that

Corollary 7.17 (Covering SDP)**.** *With at least a constant probability, we have*

$$\forall i \in [n], A_i \bullet \bar{Y} \geq 1-2\varepsilon \quad \textit{and} \quad \mathrm{Tr}(\bar{Y}) \leq 1 + O(\varepsilon)\mathsf{OPT} \ .$$

Therefore, $\frac{\bar{Y}}{1-2\varepsilon}$ gives a $(1+O(\varepsilon))$-approximate solution to the covering SDP (7.2).

Acknowledgements

We thank Richard Peng for helpful conversations. This material is based upon work partly supported by the National Science Foundation under Grant CCF-1319460.

Appendix

7.A Missing Proofs for Section 7.2

We need the following chain rule for the derivative of matrix exponential:

Proposition 7.18 ([162])**.** *If $X(t)$ is a differentiable function from reals to symmetric matrices,*

$$\frac{d}{dt}e^{X(t)} = \int_{\alpha=0}^{1} e^{\alpha X(t)} \frac{dX(t)}{dt} e^{(1-\alpha)X(t)} d\alpha \ .$$

Proof of Lemma 7.3. According to Proposition 7.18, we have

$$f'(t) = A \bullet \int_{\alpha=0}^{1} e^{\alpha(\Psi+tA)} A e^{(1-\alpha)(\Psi+tA)} d\alpha$$

Suppose further that $A = PP^T$. Then, we can write

$$f'(t) = \int_{\alpha=0}^{1} \mathrm{Tr}\Big(P^T e^{\alpha(\Psi+tA)} P P^T e^{(1-\alpha)(\Psi+tA)} P\Big) d\alpha$$

However, since $P^T e^{\alpha(\Psi+tA)} P \succeq 0$ and $P^T e^{(1-\alpha)(\Psi+tA)} P \succeq 0$, we conclude that $P^T e^{\alpha(\Psi+tA)} P \bullet P^T e^{(1-\alpha)(\Psi+tA)} P \geq 0$ and therefore $f'(t) \geq 0$ for all reals t.

Next, applying Lemma 7.1 we have that

$$f'(t) = \int_{\alpha=0}^{1} \mathrm{Tr}\Big(Ae^{\alpha(\Psi+tA)}Ae^{(1-\alpha)(\Psi+tA)}\Big)d\alpha \le \int_{\alpha=0}^{1} \mathrm{Tr}\Big(A^2e^{\Psi+tA}\Big)d\alpha$$
$$= A^2 \bullet e^{\Psi+tA} \le \varepsilon A \bullet e^{\Psi+tA} \ .$$

□

7.B Missing Proofs for Section 7.5

7.B.1 The Gradient Descent Lemma

In this section, we provide the detailed analysis of the symmetric case (i.e., when $\mathbb{T}_+$ is chosen) in the proof for Lemma 7.10.

Notice that $f_\mu(x^{(k)}) \le 0$ together with Proposition 7.9.d tells us that $\sum_{i\in[n]} x_i^{(k)} A_i \preceq (1+\varepsilon)I$. Combining it with Fact 7.14 we have $\sum_{i\in[n]} \big(x_i^{(k+1)} - x_i^{(k)}\big)A_i \succeq -\frac{4\alpha}{3}(1+\varepsilon)I \succeq -\frac{5\alpha}{3}I$ and therefore $0 \succeq B \succeq -\frac{5\alpha}{3\mu}I = -\frac{5\varepsilon}{12}I$. Applying Lemma 7.3.b with $0 \succeq B \succeq -\frac{5\varepsilon}{12}I$ to (7.6), we have

$$f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) \ge \mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - \mu \int_0^1 B \bullet e^{\Psi} \cdot e^{-5\varepsilon\tau/12} d\tau$$
$$\ge \mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - (1-\varepsilon/4)\mu B \bullet e^{\Psi} \ .$$

Recall that, for each $i \in [n]$ satisfying $x_i^{(k+1)} \neq x_i^{(k)}$, we must have $e^{\Psi} \bullet A_i - 1 > \varepsilon$ by the definition of $\mathbb{T}_+(\cdot)$. Therefore, multiplying both sides by $x_i^{(k+1)} - x_i^{(k)} \le 0$ and summing up over $i \in [n]$, we obtain

$$\mu B \bullet e^{\Psi} = e^{\Psi} \bullet \Big(\sum_{i\in[n]} (x_i^{(k+1)} - x_i^{(k)})A_i\Big) \le (1+\varepsilon)(\mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)}) \ .$$

This further implies that (after some careful term rearranging)[7]

$$\mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - (1-\varepsilon/4)\mu B \bullet e^{\Psi} \ge \frac{2}{3}(\mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - \mu B \bullet e^{\Psi})$$
$$= \frac{2}{3}\langle \nabla f_\mu(x^{(k)}), x^{(k)} - x^{(k+1)} \rangle \ge 0 \ .$$

Above, the last inequality is again by our definition of $\mathbb{T}_-$: for each $i \in [n]$ satisfying $x_i^{(k)} \neq x_i^{(k+1)}$, it must satisfy that $\nabla_i f_\mu(x^{(k)}) < -\varepsilon$ and $x_i^{(k)} \le x_i^{(k+1)}$. In conclusion, we arrive at the inequality

$$f_\mu(x^{(k)}) - f_\mu(x^{(k+1)}) \ge \frac{2}{3}\langle \nabla f_\mu(x^{(k)}), x^{(k)} - x^{(k+1)} \rangle \ge 0 \ .$$

[7] Indeed, $\mu B \bullet e^{\Psi} \le (1+\varepsilon)(\mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)})$ implies that $(1-3\varepsilon/4)\cdot \mu B \bullet e^{\Psi} \le \mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)}$ because both sides are nonpositive and $1 - 3\varepsilon/4 \ge \frac{1}{1+\varepsilon}$ for our choice of ε. Multiplying both sides by $1/3$, we have that $(1/3 - \varepsilon/4)\cdot \mu B \bullet e^{\Psi} \le (1/3)\cdot(\mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)})$. This is now equivalent to $\mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - (1-\varepsilon/4)\mu B \bullet e^{\Psi} \ge \frac{2}{3}(\mathbb{1}^T x^{(k+1)} - \mathbb{1}^T x^{(k)} - \mu B \bullet e^{\Psi})$.

7.B.2 The Coupling Lemma

The main idea in our proof to Lemma 7.12 is to divide the gradient vector $\nabla f(x) \in [-1,\infty)^n$ into four components, the component containing large coordinates (i.e., bigger than 1), the component containing positive small coordinates (i.e., in $(\varepsilon, 1]$), the component containing negative small coordinates (i.e., in $[-1,-\varepsilon)$), and the component containing negligible coordinates (i.e., in $[-\varepsilon,\varepsilon]$). The large gradients are to be taken care by the gradient descent lemma, the small (positive and negative) gradients are to be taken care by the mirror descent lemma. Formally,

Proof of Lemma 7.12. By convexity, the distance $f_\mu(x^{(k)}) - f_\mu(u)$ for an arbitrary $u \geq 0$ is upper bounded as follows:

$$\begin{aligned} &\alpha(f_\mu(x^{(k)}) - f_\mu(u)) \leq \langle \alpha \nabla f_\mu(x^{(k)}), x^{(k)} - u\rangle \\ &= \langle \alpha\eta^{(k)}, x^{(k)} - u\rangle + \langle \alpha\xi^{(k-)}, x^{(k)} - u\rangle + \langle \alpha\xi^{(k+)}, x^{(k)} - u\rangle + \langle \alpha\zeta^{(k)}, x^{(k)} - u\rangle \ , \end{aligned} \tag{7.7}$$

where

- $\xi_i^{(k-)} \stackrel{\text{def}}{=} \mathbb{T}_-(\nabla_i f_\mu(x^{(k)})) \in [-1,-\varepsilon)$ is the *truncated gradient*, capturing small negative coordinates.
- $\xi_i^{(k+)} \stackrel{\text{def}}{=} \mathbb{T}_+(\nabla_i f_\mu(x^{(k)})) \in (\varepsilon, 1]$ is the *truncated gradient*, capturing small positive coordinates.
- $\eta_i^{(k)} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} \nabla_i f_\mu(x^{(k)}) - 1, & \text{if } \nabla_i f_\mu(x^{(k)}) \geq 1; \\ 0, & \text{otherwise.} \end{array} \right\} \in [0,\infty)$, capturing the large coordinates.
- $\zeta_i^{(k)} \stackrel{\text{def}}{=} \left\{ \begin{array}{ll} \nabla_i f_\mu(x^{(k)}), & \text{if } \nabla_i f_\mu(x^{(k)}) \in [-\varepsilon,\varepsilon]; \\ 0, & \text{otherwise.} \end{array} \right\} \in [-\varepsilon,\varepsilon]$, capturing the negligible coordinates.

We analyze the four components of (7.7) one by one.

The ζ component is small: if $f_\mu(u) \leq 0$, we have

$$\langle \alpha\zeta^{(k)}, x^{(k)} - u\rangle \leq \alpha\varepsilon \cdot (\mathbb{1}^T x^{(k)} + \mathbb{1}^T u) \leq \alpha\varepsilon \cdot (1+\varepsilon)\mathsf{OPT} + \alpha\varepsilon \cdot \mathbb{1}^T u \tag{7.8}$$

where the last inequality is because $f_\mu(x^{(k)}) \leq 0$ from Lemma 7.10.

The η component can be upper bounded with the help from Lemma 7.10 as follows. Note that $\eta_i^{(k)} \neq 0$ only if $i \in B^{(k)}$ (where recall from Lemma 7.10 that $B^{(k)}$ is the set of indices whose $\nabla_i f_\mu(x^{(k)})$ is no less than 1). In particular, if $i \in B^{(k)}$ we have $\eta_i^{(k)} = \nabla_i f_\mu(x^{(k)}) - 1 < \nabla_i f_\mu(x^{(k)})$, and thus Lemma 7.10 gives

$$\begin{aligned} &\frac{4(f_\mu(x^{(k)}) - \mathbb{E}[f_\mu(x^{(k+1)})])}{\alpha} \geq \sum_{i\in B^{(k)}} x_i^{(k)} \cdot \nabla_i f_\mu\big(x^{(k)}\big) \geq \langle \eta^{(k)}, x^{(k)}\rangle \\ &\qquad \Longrightarrow \ \langle \alpha\eta^{(k)}, x^{(k)} - u\rangle \leq \langle \alpha\eta^{(k)}, x^{(k)}\rangle \leq 4(f_\mu(x^{(k)}) - \mathbb{E}[f_\mu(x^{(k+1)})]) \end{aligned}$$

Finally, the ξ components are upper bounded by Lemma 7.11 as follows. Letting $\gamma = \xi^{(k-)}$ if $\mathbb{T}^{(k)} = \mathbb{T}_-$, and $\gamma = \xi^{(k+)}$ if $\mathbb{T}^{(k)} = \mathbb{T}_+$, we have that

$$\begin{aligned}\langle \alpha\xi^{(k-)}, x^{(k)} - u\rangle + \langle \alpha\xi^{(k+)}, x^{(k)} - u\rangle &= 2\mathbb{E}[\langle \alpha\gamma, x^{(k)} - u\rangle] \\ &\le 2\alpha^2\mathsf{OPT} + 2V_{x^{(k)}}(u) - 2\mathbb{E}[V_{x^{(k+1)}}(u)] \ ,\end{aligned}$$

where the expectation is over the random choice of $\mathbb{T}$ at iteration k.

Together, we obtain

$$\begin{aligned}&\alpha(f_\mu(x^{(k)}) - f_\mu(u)) \le \langle \alpha\eta^{(k)}, x^{(k)} - u\rangle + \langle \alpha\xi^{(k-)} + \alpha\xi^{(k+)}, x^{(k)} - u\rangle + \langle \alpha\zeta^{(k)}, x^{(k)} - u\rangle \\ &\le 4(f_\mu(x^{(k)}) - \mathbb{E}[f_\mu(x^{(k+1)})]) + 2\alpha^2\mathsf{OPT} + 2V_{x^{(k)}}(u) - 2\mathbb{E}[V_{x^{(k+1)}}(u)] \\ &\qquad + \alpha\varepsilon\cdot(1+\varepsilon)\mathsf{OPT} + \alpha\varepsilon\mathbb{1}^T u \\ &\le 4(f_\mu(x^{(k)}) - \mathbb{E}[f_\mu(x^{(k+1)})]) + 2\big(V_{x^{(k)}}(u) - 2\mathbb{E}[V_{x^{(k+1)}}(u)]\big) + \alpha\cdot 2\varepsilon\mathsf{OPT} + \alpha\cdot\varepsilon\mathbb{1}^T u \ .\end{aligned}$$

□

7.B.3 The Mirror Descent Lemma

In this subsection, we are going to view our step $x^{(k)} \to x^{(k+1)}$ as a mirror descent step, and prove Lemma 7.11. We emphasize that this subsection is included in this paper only for the sake of completeness: it is almost a simple replication of the proof of [7, Lemma 3.3].

Recall that $\xi_i^{(k)} \stackrel{\text{def}}{=} \mathbb{T}^{(k)}(\nabla_i f_\mu(x^{(k)})) \in [-1,1]$ is the truncated gradient at step k, and satisfies that $\xi_i^{(k)} = \nabla_i f_\mu(x^{(k)})$ for all coordinates i such that $\nabla_i f_\mu(x^{(k)}) \in [-1,1] \setminus [-\varepsilon,\varepsilon]$. We can verify that our careful choice of $x^{(k)} \to x^{(k+1)}$ is in fact a mirror descent step on the truncated gradient:

Claim 7.19.

$$x^{(k+1)} = \arg\min_{z\ge 0}\left\{V_{x^{(k)}}(z) + \langle \alpha\xi^{(k)}, z - x^{(k)}\rangle\right\} \ . \tag{7.9}$$

Proof. This can be verified coordinate by coordinate, because the arg min function is over all possible $z \ge 0$, where this constraint does not impose any inter-coordinate constraint.

In other words, by substituting the definition of $V_{x^{(k)}}(z)$, we only need to verify that

$$x_i^{(k+1)} = \arg\min_{z_i\ge 0}\left\{\left(z_i\log\frac{z_i}{x_i^{(k)}} + x_i^{(k)} - z_i\right) + \alpha\xi_i^{(k)}\cdot(z_i - x_i^{(k)})\right\} \stackrel{\text{def}}{=} \arg\min_{z_i\ge 0}\{g(z_i)\} \ .$$

At this point, the univariate function $g(z_i)$ is convex and has a unique minimizer. Since the gradient $\frac{d}{dz_i}g(z_i) = \log\frac{z_i}{x_i^{(k)}} + \alpha\xi_i^{(k)}$, this unique minimizer is indeed $z_i =$

$x_i^{(k)} \cdot e^{-\alpha\xi_i^{(k)}}$, finishing the proof of Claim 7.19. □

After confirming that our iterative step in `PosSDPSolver` is indeed a mirror descent step, it is not hard to deduce Lemma 7.11 based on the proof of the classical mirror descent analysis.

Proof of Lemma 7.11. We deduce the following sequence of inequalities:

$$\begin{aligned}
&\langle \alpha\xi^{(k)}, x^{(k)} - u\rangle = \langle \alpha\xi^{(k)}, x^{(k)} - x^{(k+1)}\rangle + \langle \alpha\xi^{(k)}, x^{(k+1)} - u\rangle \\
&\overset{①}{=} \langle \alpha\xi^{(k)}, x^{(k)} - x^{(k+1)}\rangle + \langle -\nabla V_{x^{(k)}}(x^{(k+1)}), x^{(k+1)} - u\rangle \\
&\overset{②}{=} \langle \alpha\xi^{(k)}, x^{(k)} - x^{(k+1)}\rangle + V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u) - V_{x^{(k)}}(x^{(k+1)}) \\
&\overset{③}{\le} \sum_i \Big(\alpha\xi_i^{(k)} \cdot (x^{(k)} - x^{(k+1)}) - \frac{|x_i^{(k+1)} - x_i^{(k)}|^2}{2\max\{x_i^{(k+1)}, x_i^{(k)}\}}\Big) + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u)\big) \\
&\overset{④}{\le} \sum_i \frac{(\alpha^2\xi_i^{(k)})^2 \cdot \max\{x_i^{(k+1)}, x_i^{(k)}\}}{2} + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u)\big) \qquad (7.10) \\
&\overset{⑤}{\le} \frac{2}{3}\alpha^2 \mathbb{1}^T x^{(k)} + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u)\big) \\
&\overset{⑥}{\le} \alpha^2 \mathsf{OPT} + \big(V_{x^{(k)}}(u) - V_{x^{(k+1)}}(u)\big)
\end{aligned}$$

Here, ① is due to the minimality of $x^{(k+1)}$ in (7.9), which implies that $\nabla V_{x^{(k)}}(x^{(k+1)}) + \alpha\xi^{(k)} = 0$. ② is due to the triangle equality of Bregman divergence:

$$\begin{aligned}
\forall x, y \ge 0, \quad \langle -\nabla V_x(y), y - u\rangle &= \langle \nabla w(x) - \nabla w(y), y - u\rangle \\
&= (w(u) - w(x) - \langle \nabla w(x), u - x\rangle) - (w(u) - w(y) - \langle \nabla w(y), u - y\rangle) \\
&\quad - (w(y) - w(x) - \langle \nabla w(x), y - x\rangle) \\
&= V_x(u) - V_y(u) - V_x(y) \ .
\end{aligned}$$

③ is because $V_x(y) = \sum_i y_i \log\frac{y_i}{x_i} + x_i - y_i \ge \sum_i \frac{1}{2\max\{x_i, y_i\}}|x_i - y_i|^2$. ④ is by Cauchy-Schwarz. ⑤ is because we have $x_i^{(k+1)} \le \frac{4}{3}x_i^{(k)}$ owing to Fact 7.14. ⑥ is because we have $\mathbb{1}^T x^{(k)} \le \frac{3}{2}\mathsf{OPT}$ owing to Proposition 7.9.d (and $f_\mu(x^{(k)}) \le 0$ from Lemma 7.11). □

7.B.4 Proof of Theorem 7.13

Proof of Theorem 7.13. We begin by telescoping the inequality in Lemma 7.12 for $k = 0, 1, \ldots, T-1$, and choosing $u = \tilde{u} \overset{\text{def}}{=} (1 - \varepsilon/2)x^*$, which satisfies $\mathbb{1}^T u \le \mathsf{OPT}$ by the definition of x^*:

$$\mathbb{E}\Big[\alpha\sum_{k=0}^{T-1}(f_\mu(x^{(k)}) - f_\mu(\tilde{u}))\Big] \le 4(f_\mu(x^{(0)}) - \mathbb{E}[f_\mu(x^{(T)})]) + 2\big(V_{x^{(0)}}(\tilde{u}) - \mathbb{E}[V_{x^{(T)}}(\tilde{u})]\big) + \alpha T \cdot 3\varepsilon\mathsf{OPT} \ . \qquad (7.11)$$

Above, the expectation is over the randomness of the entire algorithm. Notice that, the second term on the right hand side of (7.11) is upper bounded by

$$\begin{aligned}
&V_{x^{(0)}}(\tilde{u}) - \mathbb{E}[V_{x^{(T)}}(\tilde{u})] \le V_{x^{(0)}}(\tilde{u}) \\
&\le \sum_i \tilde{u}_i \log \frac{\tilde{u}_i}{x_i^{(0)}} + x_i^{(0)} \le \sum_i \tilde{u}_i \log \frac{1/\|A_i\|_{\mathsf{spe}}}{(1-\varepsilon/2)/n\|A_i\|_{\mathsf{spe}}} + \frac{1-\varepsilon/2}{n\|A_i\|_{\mathsf{spe}}} \\
&\le \mathbb{1}^T \tilde{u} \cdot \log(2n) + 1 \le 2\mathsf{OPT} \cdot \log(2n) \ . \qquad (7.12)
\end{aligned}$$

Here, we have used the fact that $\tilde{u}_i \le \frac{1}{\|A_i\|_{\mathsf{spe}}}$ since $\tilde{u}_i A_i \preceq I$.

From here, we want to prove that $\mathbb{E}[f_\mu(x^{(T)})] \le -(1-5\varepsilon)\mathsf{OPT}$ by way of contradiction. Suppose not, that is, $\mathbb{E}[f_\mu(x^{(T)})] > -(1-5\varepsilon)\mathsf{OPT}$, we have $f_\mu(x^{(0)}) - \mathbb{E}[f_\mu(x^{(T)})] \le 0 + (1-5\varepsilon)\mathsf{OPT} \le \mathsf{OPT}$, giving an upper bound on the first term on the right hand side in (7.11). Substituting this and (7.12) to (7.11), and dividing αT on both sides, we get

$$\begin{aligned}
&\frac{1}{T}\sum_{k=0}^{T-1}(\mathbb{E}[f_\mu(x^{(k)})] - f_\mu(\tilde{u})) \\
&\le \frac{4}{\alpha T}(f_\mu(x^{(0)}) - \mathbb{E}[f_\mu(x^{(T)})]) + \frac{2}{\alpha T}\big(V_{x^{(0)}}(\tilde{u}) - \mathbb{E}[V_{x^{(T)}}(\tilde{u})]\big) + 3\varepsilon\mathsf{OPT} \\
&\le \frac{4\mathsf{OPT}}{\alpha T} + \frac{4\mathsf{OPT}\cdot\log(2n)}{\alpha T} + 3\varepsilon\mathsf{OPT} \ .
\end{aligned}$$

Finally, since we have chosen $T \ge \frac{8\log(2n)}{\alpha\varepsilon}$, the above right hand side is no greater than $4\varepsilon\mathsf{OPT}$. This, by an averaging argument, tells us the existence of some $k \in \{0,1,\ldots,T-1\}$ with $\mathbb{E}[f_\mu(x^{(k)})] \le f_\mu(\tilde{u}) + 4\varepsilon\mathsf{OPT} \le -(1-5\varepsilon)\mathsf{OPT}$ (where we have used $f_\mu(\tilde{u}) \le -(1-\varepsilon)\mathsf{OPT}$ from Proposition 7.9.b). However, it contradicts to the hypothesis that $\mathbb{E}[f_\mu(x^{(T)})] > -(1-5\varepsilon)\mathsf{OPT}$ because $f_\mu(x^{(k)}) \ge f_\mu(x^{(T)})$ according to Lemma 7.10. This finishes the proof that $\mathbb{E}[f_\mu(x^{(T)})] \le -(1-5\varepsilon)\mathsf{OPT}$.

The fact that $\frac{x^{(T)}}{1+\varepsilon}$ provides a $(1-O(\varepsilon))$ approximate solution for the packing SDP is due to Proposition 7.9.e and Markov's inequality which states that $f_\mu(x^{(T)}) \le -(1-O(\varepsilon))\mathsf{OPT}$ with at least constant probability. □

7.C Missing Proofs for Section 7.6

The proof of Lemma 7.15 is completely analogous to its LP variant in [7]. We include it only for the sake of completeness.

Lemma 7.15. *For any $T \ge \frac{8}{\alpha\varepsilon} = \Omega(\frac{\log(nm/\varepsilon)}{\varepsilon^3})$, we have that $\mathbb{E}[\mathrm{Tr}(\bar{Y})] \le (1+7\varepsilon)\mathsf{OPT}$.*

Proof. Telescoping Lemma 7.12 for $k = 0, 1, \ldots, T-1$ and $u = 0$, we have that

$$\frac{1}{T}\mathbb{E}\Big[\sum_{k=0}^{T-1}\langle \nabla f_\mu(x^{(k)}), x^{(k)}\rangle\Big]$$

$$\leq \frac{4}{\alpha T}(f_\mu(x^{(0)}) - \mathbb{E}[f_\mu(x^{(T)})]) + \frac{2}{\alpha T}\big(V_{x^{(0)}}(0) - \mathbb{E}[V_{x^{(T)}}(0)]\big) + 2\varepsilon\mathsf{OPT}$$
$$\leq \frac{4}{\alpha T}(f_\mu(x^{(0)}) - \mathbb{E}[f_\mu(x^{(T)})]) + \frac{2}{\alpha T}V_{x^{(0)}}(0) + 2\varepsilon\mathsf{OPT}$$
$$\leq \frac{4}{\alpha T}(f_\mu(x^{(0)}) - \mathbb{E}[f_\mu(x^{(T)})]) + \frac{2}{\alpha T} + 2\varepsilon\mathsf{OPT}\ . \tag{7.13}$$

Above, the last inequality uses the fact that $V_{x^{(0)}}(0) = \mathbb{1}^T x^{(0)} \leq 1$.

We now respectively lower and upper bound the two sides of (7.13) as follows. One one hand, using the definition of gradient, the left hand side of (7.13) is lower bounded as

$$\begin{aligned}\langle \nabla f_\mu(x^{(k)}), x^{(k)}\rangle &= \sum_{i\in[n]} x_i^{(k)} A_i \bullet e^{\frac{1}{\mu}\left(\sum_{i\in[n]} x_i^{(k)} A_i - I\right)} - \mathbb{1}^T x^{(k)} \\ &\geq (1-\varepsilon) I \bullet e^{\frac{1}{\mu}\left(\sum_{i\in[n]} x_i^{(k)} A_i - I\right)} - \mathbb{1}^T x^{(k)} - m\cdot(\frac{\varepsilon}{nm})^4 \\ &= (1-\varepsilon)\mathrm{Tr}(Y(x^{(k)})) - \mathbb{1}^T x^{(k)} - m\cdot(\frac{\varepsilon}{nm})^4\ .\end{aligned}$$

Above, the (only) inequality is because if $B \stackrel{\text{def}}{=} \sum_{i\in[n]} x_i^{(k)} A_i$ has eigenvalues $\lambda_1, \ldots, \lambda_m \geq 0$, then $\sum_{i\in[n]} x_i^{(k)} A_i \bullet e^{\frac{1}{\mu}\left(\sum_{i\in[n]} x_i^{(k)} A_i - I\right)} = \sum_{j\in[m]} \lambda_j \cdot e^{(\lambda_j - 1)/\mu}$. However, if there are some λ_j satisfying $\lambda_j < 1-\varepsilon$, the corresponding term $e^{\frac{1}{\mu}(\lambda_j - 1)} \leq e^{-\varepsilon/\mu} = (\frac{\varepsilon}{nm})^4$ is very small, and there are at most m such small terms. As a result, one must have $\sum_{j\in[m]} \lambda_j \cdot e^{(\lambda_j-1)/\mu} \geq (1-\varepsilon)\sum_{j\in[m]} \cdot e^{(\lambda_j-1)/\mu} - m\cdot(\frac{\varepsilon}{nm})^4 = (1-\varepsilon) I \bullet e^{\frac{1}{\mu}\left(\sum_{i\in[n]} x_i^{(k)} A_i - I\right)} - m\cdot(\frac{\varepsilon}{nm})^4$.

On the other hand, since $x_i^{(T)} A_i \leq (1+\varepsilon) I$ by Proposition 7.9.d, we must have $\mathbb{1}^T x^{(T)} \leq (1+\varepsilon)\mathsf{OPT}$ by the definition of OPT, and thus $f_\mu(x^{(T)}) \geq 0 - (1+\varepsilon)\mathsf{OPT}$. This gives an upper bound on the right hand side of (7.13) that is $\frac{4(1+\varepsilon)}{\alpha T}\mathsf{OPT} + \frac{2}{\alpha T} + 2\varepsilon\mathsf{OPT} \leq 3\varepsilon\mathsf{OPT}$, due to our choice of $T \geq \frac{8}{\alpha\varepsilon}$.

Together, we deduce from (7.13) that

$$(1-\varepsilon)\frac{1}{T}\sum_{k=0}^{T-1} \mathbb{E}\Big[\mathrm{Tr}(Y(x^{(k)})) - \mathbb{1}^T x^{(k)}\Big] - m\cdot(\frac{\varepsilon}{nm})^4 \leq 3\varepsilon\mathsf{OPT}$$
$$\implies \mathbb{E}[\mathrm{Tr}(\bar{Y})] = \mathrm{Tr}\mathbb{E}\Big[\frac{1}{T}\sum_k Y(x^{(k)})\Big] \leq \frac{1}{T}\sum_k \mathbb{E}[\mathbb{1}^T x^{(k)}] + 4\varepsilon\mathsf{OPT} \leq (1+\varepsilon)\mathsf{OPT} + 4\varepsilon\mathsf{OPT}\ ,$$

where the last inequality is from $\mathbb{1}^T x^{(k)} \leq (1+\varepsilon)\mathsf{OPT}$ for each k (see Proposition 7.9.d). □

As mentioned earlier, our proof for Lemma 7.16 below is fundamentally different from its much weaker version in [7].

Lemma 7.16. *For any $T \geq \frac{8}{\alpha\varepsilon}$, with probability at least $1 - \frac{\varepsilon}{100}$ we have $A_i \bullet \bar{Y} \geq 1 - 2\varepsilon$ for all $i \in [n]$.*

Proof. For each iteration $k = 0, \ldots, T-1$ and coordinate $i \in [n]$, we denote by

- $\gamma_i^{(k)} \stackrel{\text{def}}{=} \mathbb{T}^{(k)}(\nabla_i f_\mu(x^{(k)})) \in [-1, 1]$ the actual truncated gradient, and
- $\xi_i^{(k)} \stackrel{\text{def}}{=} \frac{1}{2}\big(\mathbb{T}_-(\nabla_i f_\mu(x^{(k)})) + \mathbb{T}_+(\nabla_i f_\mu(x^{(k)}))\big) \in [-1/2, 1/2]$ the expected truncated gradient.

It is easy to verify that $\mathbb{E}[\gamma^{(k)}] = \xi^{(k)}$, where the expectation is over the random choice of $\mathbb{T}^{(k)}$. In addition, since $\nabla_i f_\mu(x^{(k)}) = 2\xi_i^{(k)}$ whenever $\nabla_i f_\mu(x^{(k)}) \in [-1, 1] \setminus [-\varepsilon, \varepsilon]$ owing to the definition of the thresholding functions, we automatically have

$$\nabla_i f_\mu(x^{(k)}) \geq 2\xi_i^{(k)} - \varepsilon \ .$$

In the first step, recalling that $x_i^{(T)} = x_i^{(0)} \cdot e^{-\alpha \sum_{k=0}^{T-1} \gamma_i^{(k)}}$ by the definition of our update rule (Line 8 of `PosSDPSolver`), and recalling that $x_i^{(T)} A_i \preceq (1+\varepsilon) I \prec 1.5 I$ due to Proposition 7.9.d which implies $x_i^{(T)} \leq \frac{1.5}{\|A_i\|_{\mathsf{spe}}}$, we automatically have that for every $i \in [n]$, independent of the randomness of the algorithm, it always satisfies that

$$\frac{1}{T}\sum_{k=0}^{T-1} \gamma_i^{(k)} \geq -\frac{\log(1.5/(\|A_i\|_{\mathsf{spe}} \cdot x_i^{(0)}))}{\alpha T} \geq \frac{-\log(2n)}{\alpha T} \geq -\frac{\varepsilon}{8} \ .$$

Above, the second inequality is due to our choice of $x^{(0)}$, and the third inequality is due to our choice of T. Next, define $Z_{k,i} \stackrel{\text{def}}{=} \sum_{j=0}^{k-1} (\gamma_i^{(k)} - \xi_i^{(k)})$, we have that $\{Z_{k,i}\}_{k=1}^{T}$ is a martingale, satisfying that $\mathbb{E}[Z_{k,i} | Z_{1,i}, \ldots, Z_{k-1,i}] = Z_{k-1,i}$ and $|Z_{k,i} - Z_{k-1,i}| \leq 1/2$. By the Azuma-Hoeffding inequality, we have

$$\Pr\big[\frac{1}{T}\sum_{k=0}^{T-1}(\xi_i^{(k)} - \gamma_i^{(k)}) < -\frac{\varepsilon}{4}\big] = \Pr\big[\frac{Z_{T,i}}{T} > \frac{\varepsilon}{4}\big] \leq e^{\frac{-\varepsilon^2 T}{8}} \leq \frac{\varepsilon}{100n} \ .$$

By a union bound, with probability at least $1 - \varepsilon/100$, for every $i \in [n]$,

$$\begin{aligned}\frac{1}{T}\sum_{k=0}^{T-1} \nabla_i f_\mu(x^{(k)}) \geq \frac{1}{T}\sum_{k=0}^{T-1} 2\xi_i^{(k)} - \varepsilon &= 2\frac{1}{T}\sum_{k=0}^{T-1}(\xi_i^{(k)} - \gamma_i^{(k)}) + 2\frac{1}{T}\sum_{k=0}^{T-1}\gamma_i^{(k)} - \varepsilon \\ &\geq 2 \cdot (-\frac{\varepsilon}{4}) - \frac{\varepsilon}{4} - \varepsilon > -2\varepsilon \ .\end{aligned}$$

In other words, with probability at least $1 - \varepsilon/100$, for every $i \in [n]$,

$$A_i \bullet \bar{Y} - 1 = \frac{1}{T}\sum_{k=0}^{T-1}(A_i \bullet Y(x^{(k)}) - 1) = \frac{1}{T}\sum_{k=0}^{T-1} \nabla_i f_\mu(x^{(k)}) \geq -2\varepsilon \ . \qquad \square$$

Chapter 8

Spectral Sparsification and Regret Minimization Beyond Matrix Multiplicative Updates

This chapter is based on the result published in [4], and its further edits can be found at:

http://arxiv.org/abs/1506.04838.

In this paper, we provide a novel construction of the linear-sized spectral sparsifiers of Batson, Spielman and Srivastava [26]. While previous constructions required $\Omega(n^4)$ running time [26, 168], our sparsification routine can be implemented in almost-quadratic running time $O(n^{2+\varepsilon})$.

The fundamental conceptual novelty of our work is the leveraging of a strong connection between sparsification and a regret minimization problem over density matrices. This connection was known to provide an interpretation of the randomized sparsifiers of Spielman and Srivastava [151] via the application of matrix multiplicative weight updates (MWU) [39, 160]. In this paper, we explain how matrix MWU naturally arises as an instance of the Follow-the-Regularized-Leader framework and generalize this approach to yield a larger class of updates. This new class allows us to accelerate the construction of linear-sized spectral sparsifiers, and give novel insights on the motivation behind Batson, Spielman and Srivastava [26].

8.1 Introduction

A powerful tool to handle large-scaled graphs is to compress them by reducing their sizes, while preserving properties of interest such as the size of cuts [28, 29] or the routability of certain flows [41]. This *sparsification* procedures also play an important role as fundamental primitives behind many fast graph algorithms [88, 128]. In this paper, we consider the strong notion of *spectral sparsifier* put forward by Spielman

and Teng [152, 153]: G' is $(1+\varepsilon)$-spectral approximate to G if G' is a subgraph of G with possibly reweighted edges, and for every $x \in \mathbb{R}^n$,

$$x^T L_G x \leq x^T L_{G'} x \leq (1+\varepsilon) x^T L_G x \quad \text{or equivalently} \quad L_G \preceq L_{G'} \preceq (1+\varepsilon) L_G \ ,$$

where L_G and $L_{G'}$ are respectively the graph Laplacian matrices of G and G'.

The algorithm of Spielman and Srivastava [151] constructs $(1+\varepsilon)$-spectral sparsifiers with $O(n \log n/\varepsilon^2)$ edges in nearly linear time by randomly sampling edges proportionally to their effective resistance. In a seminal paper, Batson, Spielman and Srivastava [26] give $(1+\varepsilon)$-spectral sparsifiers with $O(n/\varepsilon^2)$ edges, but their construction and subsequent algorithm by [168] require $O(mn^3/\varepsilon^2)$ and $O(mn^2/\varepsilon^2 + n^4/\varepsilon^4)$ time respectively. We shall refer to their analysis and algorithm the BSS for short. The main contribution of this paper is to give an improved construction of linear-sized spectral sparsifiers that runs in almost-quadratic time.

Theorem 8.1. *For any even integer $q \geq 2$ and any $\varepsilon \in (0, \frac{1}{4\sqrt{q}})$, there is an algorithm that, for any weighted undirected graph G with n vertices and m edges, with probability at least $1 - n^{-\Omega(1)}$, constructs a $(1+\varepsilon)$-spectral sparsifier G' that has at most $O(\sqrt{q}n/\varepsilon^2)$ edges in time $\tilde{O}(mn^{1+1/q}/\varepsilon^5)$.*

Since q can be chosen as a large constant and the graph can be preprocessed to reduce the number of edges to $m = O(n \log n)$, the above running time is almost quadratic in terms of n.

Graph sparsification is a special case of sparsifying sums of rank-1 PSD matrices (see [26] and Appendix 8.B). Our algorithm for Theorem 8.1 also applies to this more general problem with an almost cubic running time, which is stil an improvement over the previous quartic running time.

Theorem 8.2. *For any even integer $q \geq 2$ and any $\varepsilon \in (0, \frac{1}{4\sqrt{q}})$, there is an algorithm that, for any decomposition $I = \sum_{i=1}^m v_i v_i^T \in \mathbb{R}^{n \times n}$ of rank-1 matrices, with probability at least $1 - n^{-\Omega(1)}$, constructs scalars $s_i \geq 0$ with $|\{i : s_i > 0\}| \leq O(\sqrt{q}n/\varepsilon^2)$ that satisfies $I \preceq \sum_{i=1}^m s_i v_i v_i^T \preceq (1+\varepsilon) I$ in time $\tilde{O}(n^{3+1/q}/\varepsilon^5 + mn/\varepsilon^4)$.*

The fundamental conceptual novelty of our work is the establishment of a deep connection between graph or matrix sparsifications and a regret minimization problem over PSD matrices (see Section 8.1.1). This relation was known [39, 160] for the randomized sparsifiers of Spielman and Srivastava [151], for which the underlying matrix concentration bound can be easily recovered as an application of the matrix version of Multiplicative Weight Updates (MWU) [11, 125], a standard online learning algorithm. However, it was not clear how this interpretation could be extended to BSS, despite a clear analogy was also noted by de Carli Silva, Harvey and Sato (see [39, Section 8]). Both the MWU and the BSS rely on potential function arguments, where the potential is essentially a robust version to capture of the maximum and minimum graph eigenvalues. In this paper, we provide the missing piece of this

interpretation: we consider a generalization of MWU to a larger class of updates, and show that the BSS can be recovered as an instance of this class. Beyond our faster implementation of sparsification, we believe that this interpretation is of independent interest and may be useful in other areas in which the argument of BSS has found application [111].

We focus on updates coming from the *follow-the-regularized-leader* (FTRL) framework. The choice of regularizer in this framework fully determines the update strategy and the corresponding potential function. See for example the recent survey by Hazan [72]. The standard MWU argument can be recovered as an instance of FTRL, where the regularizer is chosen to be the entropy function. In contrast, we choose a different class of regularizers consisting of all $\ell_{1-1/q}$ semi-norms for $q \geq 2$, and provide corresponding regret bounds in Section 8.3. In Section 8.4 and Section 8.5, we show that the choice $q = 2$ recovers an algorithm which is somewhat similar to BSS, and produces linear-sized spectral sparsifiers. This algorithm can be implemented to run in a $O(mn^{3/2})$ time. Finally, in Section 8.6, we consider regularizers corresponding to large, constant $q > 2$, which yield very different algorithms from BSS with almost quadratic running time.

8.1.1 Regret Minimization

In this subsection, we discuss our contribution on the problem of regret minimization in online linear optimization [72]. Our technical results apply to the more general case of online PSD linear optimization over the set of density matrices, but our key contributions are described more concisely in the scalar case.

Let $\Delta_n = \{x \in \mathbb{R}^n : x \geq 0 \wedge \mathbb{1}^T x = 1\}$ be the unit simplex in $\mathbb{R}^n$, and we call a vector in Δ_n an *action*. A player is going to play T actions $x_0, \ldots, x_{T-1} \in \Delta_n$ in a row; only after playing x_k, the player observes a feedback vector $f_k \in \mathbb{R}^n$, which may depend on x_k, and suffers the linear loss $\langle f_k, x_k \rangle$. The regret minimization problem asks us to device a strategy for the player that minimizes the *regret*, i.e., difference between the total loss suffered by the player and the loss suffered by the *a* posteriori best fixed action $u \in \Delta_n$:

$$\text{minimize} \quad \max_{u \in \Delta_n} R(u), \qquad \text{where } R(u) \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \langle f_k, x_k - u \rangle \ .$$

A well-known strategy for this problem is to update x_k in a multiplicative fashion: for each coordinate $i \in [n]$, define $x_{k+1,i}$ to be proportional to $x_{k,i} \cdot \exp^{-\alpha \cdot f_{k,i}}$ for some parameter $\alpha > 0$. This strategy is known as the *multiplicative weight update*. Its classical analysis [10] implies

$$\forall u \in \Delta_n, \quad R(u) = \sum_{k=0}^{T-1} \langle f_k, x_k - u \rangle \leq \frac{\alpha}{2} \sum_{k=0}^{T-1} \|f_k\|_\infty^2 + \frac{\log n}{\alpha} \ . \tag{8.1}$$

The first term on the righthand side contributes a regret of $\|f_k\|_\infty^2$ that is paid at every iteration, and we call it the *width term*. The second term is a fixed start-up

cost corresponding to 'how long it takes the update to explore the whole Δ_n', and we call it the *diameter term*. If for all iterations k, $\|f_k\|_\infty$ is upper bounded by ρ, known as the *width* of the problem, the trade-off between the width and diameter terms can be be optimized by the choice of $\alpha > 0$ to show that the total regret is at most $O(\rho\sqrt{T \log n})$.

Optimization Interpretation. We take an optimization perspective to describe MWU and its generalizations by characterizing our strategies as instances of the *follow-the-regularized-leader* and *mirror descent* frameworks. Let $w(\cdot)$ be a strongly convex function over the simplex, known as the *regularizer*. The follow-the-regularized-leader strategy with parameter $\alpha > 0$ can be described as a trade-off between minimizing the loss incurred so far and the value of the regularizer.

$$\texttt{FTRL}: \qquad x_{k+1} = \arg\min_{z \in \Delta_n} \left\{ w(z) + \alpha \textstyle\sum_{j=0}^{k} \langle f_j, z \rangle \right\} \ . \tag{8.2}$$

Similarly, the mirror-descent strategy optimizes a trade-off

$$\texttt{MirrorDescent}: \qquad \text{start with } x_0 = \left(\tfrac{1}{n}, \ldots, \tfrac{1}{n}\right); \quad x_{k+1} \leftarrow \arg\min_{z \in \Delta_n} \{ V_{x_k}(z) + \alpha \langle f_k, z \rangle \} \ , \tag{8.3}$$

where $V_x(y) \stackrel{\text{def}}{=} w(y) - w(x) - \langle \nabla w(x), y - x \rangle$ is the induced Bregman divergence. Under mild assumptions (which are satisfied in this paper, see Appendix 8.A), it is easy to check that `MirrorDescent` is equivalent to `FTRL`. We will therefore interchangeably use `MirrorDescent` and `FTRL` in the rest of the paper, because `FTRL` gives the cleaner description for the updates, while `MirrorDescent` provides a simpler analysis. The MWU strategy is an instance of the two equivalent strategies above, with the choice of regularizer $w(x) \stackrel{\text{def}}{=} \sum_i x_i \log x_i - x_i$, i.e. the (negative) entropy function.

Previous Work. The MWU is a simple but extremely powerful algorithmic tool that has been repeatedly discovered in theory of computation, machine learning, optimization, and game theory (see for instance the survey [10] and the book [40]). Since MWU has found numerous important applications in semidefinite programming [11, 9], constraint satisfaction problem [154], maximum flow [46], sparsest cut [149], balanced separator [126], small set expansion [23], traveling salesman problem [12], zero-sum games [51], and fractional packing problems [68]. The analysis of follow-the-regularized-leader can be found in the surveys [72, 142], while that of the mirror descent appears in the the book [27].

Beyond MWU. Historically, MWU has been extended at least from three orthogonal directions. In this paper, we pursue all these three directions simultaneously (see our summary in Table 8.1.)

1. **From vector to matrix.** Instead of studying actions x in the forms of n-dimensional probability distributions, one can study density matrices X in $\Delta_{n \times n}$, the set of PSD matrices whose trace equals to one. This is a generaliza-

tion from a set of "experts" corresponding to $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ to all combinations of the form $\sum_{i=1}^n t_i \mathbf{e}_i$ where t is on the n-dimensional unit sphere $\mathbb{S}^{n-1}$. Accordingly, each loss vector f_k can be generalized to a symmetric matrix $F_k \in \mathbb{R}^{n \times n}$, so the loss of any density matrix X becomes $F_k \bullet X = \mathrm{Tr}(F_k X)$. (If $X = vv^T$ is of rank one, then $F_k \bullet X = v^T F_k v$.) Among many applications, the matrix version of MWU has been used in designing algorithms for solving semidefinite programs [11] and finding balanced separators [126], and in the proof of QIP = PSPACE [81].

2. **Local norm convergence.** The width term $\|f_k\|_\infty^2$ in the regret upper bound (8.1) can be replaced with $\langle |f_k|, x_k \rangle \cdot \|f_k\|_\infty$. (Here, we have used $|f_k|$ to denote coordinate-wise absolute value of f_k.) This technique is known as the local-norm technique because $\langle |f_k|, x_k \rangle$ is a local way to measure the length of f_k with respect to x_k. Since $\langle |f_k|, x_k \rangle \cdot \|f_k\|_\infty$ is never larger than $\|f_k\|_\infty^2$, as well as $x_k \in \Delta_n$, this new upper bound can only be smaller than the original. Indeed, this tighter bound has proved useful in the multi-arm bandit problem [2], and in the solution of positive linear programs [7]. It also underpins the negative-width technique of [10].

3. **Change of regularizer.** If one replaces the entropy regularizer with the $\ell_{1-1/q}$-regularizer $w(x) = -\frac{q}{q-1} \sum_{i=1}^n x_i^{1-1/q}$ for any $q \geq 2$, the corresponding update rule changes

$$\text{from} \quad \boxed{x_{k+1,i} = \exp^{-\sum_{j=0}^k \alpha f_{j,i} + c}} \quad \text{to} \quad \boxed{x_{k+1,i} = \left(\sum_{j=0}^k \alpha f_{j,i} + c\right)^{-q}} \ ,$$

where in both cases c is the unique constant that ensures $x_{k+1} \in \Delta_n$. The FTRL framework is very powerful as the choice of regularizer $w(x)$ completely determines both the form and the analysis of the update strategy. Ultimately, different regularizers achieve different trade-offs between the width and diameter terms in Equation (8.1). For instance, the $\ell_{1/2}$-regularizer yields the following regret bound

$$\forall u \in \Delta_n, \quad R(u) \leq O(\alpha) \cdot \sum_{k=0}^{T-1} \langle |f_k|, x_k \rangle \cdot \max_{i \in [n]} |f_{k,i} \sqrt{x_{k,i}}| + \frac{2\sqrt{n}}{\alpha} \ .$$

The diameter term is now $2\sqrt{n}$, much worse than $\log n$ in the entropy case in (8.1). However, since (the local norm version of) the width term goes from $\langle |f_k|, x_k \rangle \cdot \|f_k\|_\infty$ to $\langle |f_k|, x_k \rangle \cdot \max_{i \in [n]} |f_{k,i} \sqrt{x_{k,i}}|$, the width term may become smaller.. This is exactly the case in the sparsification case, where the feedback vectors, corresponding to the edges added to the sparsifier, may be weighted up by a factor as large as n, so that we may have $\|f_k\|_\infty \geq n$. In this scenario, the use of a more stongly-convex regularizer, such as $\ell_{1/2}$, allows us to measure the width in a more convenient local norm and yields the BSS linear-sized sparsifier(see Figure 8-1 on page 222 for a visual comparison of different regularizers).

Paper	Allow Matrix?	Allow Local Norm?	Allow Non-Entropy Regularizer?
[131, 65] [9, 10]	no	no	no
[2, 7]	no	yes	no
[13, 36]	no	yes	yes
[11, 126]	yes	no	no
[74]	yes	yes	no
[this paper]	yes	yes	yes

Table 8.1: Comparisons among prior results on the regret minimization problem.

We point out that the $\ell_{1-1/q}$-regularizers have also been used, albeit solely in the scalar case, by the machine learning community to obtain asymptotically optimal strategies for the multi-arm bandit problem [13, 36].

8.1.2 Extensions

High Rank Sparsification. Our same algorithm of Theorem 8.1 and 8.2 also applies to sparsifying sums of PSD matrices, rather than just rank-1 PSD matrices. This recovers the same result of de Carli Silva, Harvey, and Sato [39]. Such an extension has been shown important for problems such as finding hypergraph sparsifiers, finding sparse SDP solutions, and finding sparsifiers on subgraphs. However, as in the rank-1 case, the detailed running time of our algorithm has to be examined separately for each specific sparsification problem.

As an example, given a weighted undirected graph G that is decomposed into edge-disjoint subgraphs, the goal of *linear-sized subgraph sparsification* is to construct a $(1+O(\varepsilon))$-spectral sparsifier G' to G, so that G' consists only of the reweighted versions of at most n/ε^2 given subgraphs. Our same algorithm for Theorem 8.1 runs in time $\tilde{O}(mn^{1+1/q}/\varepsilon^5)$ for this problem.

Weak Unweighted Graph Sparsification. Given $\kappa \in [1, m/n]$, consider the problem of finding a κ-spectral sparsifier of G containing $O(m/\kappa)$ distinct edges from E, *without* reweighting. This problem is very recently studied by Anderson, Gu and Melgaard [8], our regret minimization framework allows us to design a simple and almost-quadratic-time algorithm for this problem, improving from the quartic time complexity of [8].

8.2 Preliminaries

Throughout this paper, for a cleaner representation that depends on the context, we interchangeably use $X \bullet Y = \langle X, Y \rangle = \mathrm{Tr}(XY)$ to denote the inner product between two symmetric matrices. If X is symmetric, we use e^X to denote its matrix exponential

and $\log X$ to denote its matrix logarithm, when X is PSD. If X is symmetric with eigendecomposition $X = \sum_{i=1}^n \lambda_i v_i v_i^T$ we denote by $|X| \stackrel{\text{def}}{=} \sum_{i=1}^n |\lambda_i| v_i v_i^T$. For any symmetric X, we use $\|X\|_{\mathsf{spe}}$ to denote the spectral norm of X, and $\lambda_{\mathsf{max}}(X), \lambda_{\mathsf{min}}(X)$ to denote its largest and smallest eigenvalues. We define $\Delta_{n\times n} \stackrel{\text{def}}{=} \{X \in \mathbb{R}^{n\times n} : X \succeq 0, \mathrm{Tr}X = 1\}$ to be the set of positive semidefinite (PSD) matrices with trace 1. This should be seen as the matrix generalization of the n-dimensional simplex $\Delta_n \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n : x \geq 0, \mathbb{1}^T x = 1\}$.

Regularizers and Bregman Divergence. We are interested in two types of regularizers over $\Delta_{n\times n}$, namely, $w(X) \stackrel{\text{def}}{=} X \bullet (\log X - I)$, known as the entropy regularizer, and $w(X) \stackrel{\text{def}}{=} -\frac{q}{q-1}\mathrm{Tr}X^{1-1/q}$ for some $q > 1$, which we call the $\ell_{1-1/q}$-regularizer. The corresponding Bregman divergences $V_X(Y) \stackrel{\text{def}}{=} w(Y) - w(X) - \langle \nabla w(X), Y - X\rangle$ are the following.

$$\text{entropy case:} \quad V_X(Y) = Y \bullet (\log Y - \log X) - I \bullet (Y - X) \ ,$$

$$\ell_{1-1/q} \text{ case:} \quad V_X(Y) = X^{-1/q} \bullet Y + \frac{1}{q-1}\mathrm{Tr}X^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}Y^{1-1/q} \ .$$

Note that both regularizers above and their Bregman divergences are convex over the cone of PSD matrices.[1] We now state some classical properties of Bregman divergence. Their proofs are included in Appendix 8.D for completeness.

Lemma 8.3. *The Bregman divergence of a convex differentiable function $w(\cdot)$ has the properties:*

- *Non-negativity: $V_X(Y) \geq 0$ for all $X, Y \geq 0$.*
- *The "three-point equality": $\langle \nabla w(X) - \nabla w(Y), X - U\rangle = V_X(U) - V_Y(U) + V_Y(X)$.*
- *Given $\tilde{X} \succeq 0$ and $X = \arg\min_{Z\in\Delta_{n\times n}} V_{\tilde{X}}(Z)$ as the* Bregman projection*, we have the "generalized Pythagorean theorem" for all $U \in \Delta_{n\times n}$: $V_{\tilde{X}}(U) \geq V_X(U) + V_{\tilde{X}}(X) \geq V_X(U)$.*

8.3 Regret Minimization in Full Information

In this section, we consider the following setting of the regret minimization problem, known as the full information setting. At each iteration $k = 0, \ldots, T-1$, the player chooses an action $X_k \in \Delta_{n\times n}$, receives a symmetric loss matrix $F_k \in \mathbb{R}^{n\times n}$ and suffers a loss $\langle F_k, X_k\rangle$. At this point, the player is allowed to observe the full matrix F_k without any restriction.

Again, the goal of the player is to minimize the regret with respect to any fixed matrix $U \in \Delta_{n\times n}$:

$$R(U) \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \langle F_k, X_k - U\rangle \ .$$

[1] While this is easy to check by taking the second derivative for the entropy regularizer, it is less obvious for the $\ell_{1-1/q}$ regularizer. The latter follows easily from Lieb's concavity theorem [96, 31].

The best choice of U in hindsight can be taken as the rank-1 projection over a minimum eigenvector of $\sum_{k=0}^{T-1} F_k$. As a result, the total loss for the best choice of U is $\lambda_{\mathsf{min}}\big(\sum_{k=0}^{T-1} F_k\big)$.

Entropy Regularizer. If $w(\cdot)$ is the entropy regularizer, then (8.2) can be explicitly written as

$$\texttt{MirrorDescent}_{\mathsf{exp}}: \qquad X_k = \exp^{cI - \alpha \sum_{j=0}^{k-1} F_j} \ , \tag{8.4}$$

where $c \in \mathbb{R}$ is the unique constant that ensures $\mathrm{Tr} X_k = 1$. This is also known as the *matrix multiplicative weight update* method, and the following theorem gives its regret bound.[2]

Theorem 8.4. *In* $\texttt{MirrorDescent}_{\mathsf{exp}}$*, if the parameter* $\alpha > 0$ *satisfies* $\alpha F_k \succeq -I$ *for all iterations* $k = 0, 1, \ldots, T-1$*, then, for every* $U \in \Delta_{n \times n}$*,*

$$R(U) \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \langle F_k, X_k - U \rangle \leq \alpha \sum_{k=0}^{T-1} \big(X_k \bullet |F_k|\big) \cdot \|F_k\|_{\mathsf{spe}} + \frac{V_{X_0}(U)}{\alpha} \ .$$

We note that $V_{X_0}(U) \leq \log n$*.*

Our proof of Theorem 8.4 uses a technique known as the *tweaked version* of mirror descent (see [167, 133]). We define an intermediate point $\tilde{X}_{k+1} = \arg\min_{Z \succeq 0} \big\{ V_{X_k}(Z) + \alpha \langle F_k, Z \rangle \big\}$ as the minimizer over $Z \succeq 0$, rather than $Z \in \Delta_{n \times n}$ as in (8.3). Accordingly, the actual point X_{k+1} equals to $\arg\min_{Z \in \Delta_{n \times n}} \{ V_{\tilde{X}_{k+1}}(Z) \}$, the *Bregman projection* of $\tilde{X}_{k+1}$ back to the hyperplane $\mathrm{Tr} Z = 1$. This two-step interpretation of mirror descent gives a very clean proof to our regret bound, and we defer this full proof to Appendix 8.E.

$\ell_{1-1/q}$ regularizer. If $w(\cdot)$ is the $\ell_{1-1/q}$ regularizer, then (8.2) can be explicitly written as

$$\texttt{MirrorDescent}_{\ell_{1-1/q}}: \qquad X_k = \Big(cI + \alpha \textstyle\sum_{j=0}^{k-1} F_j \Big)^{-q} \ , \tag{8.5}$$

where $c \in \mathbb{R}$ is the unique constant that ensures $cI + \alpha \sum_{j=0}^{k-1} F_j \succ 0$ and $\mathrm{Tr} X_k = 1$.

If we focus on the special case of $q = 2$ and each F_k having rank 1, the following theorem gives the regret bound for $\texttt{MirrorDescent}_{\ell_{1/2}}$.

[2] The scalar version of this theorem was proved for instance in [1, 143, 7]. A slightly different matrix version of this theorem was proved in [74] (in particular, the authors of [74] have required $I \succeq \alpha F_k \succeq -I$ while in fact it suffices to only require $\alpha F_k \succeq -I$.

Theorem 8.5. *In* $\texttt{MirrorDescent}_{\ell_{1/2}}$*, if the parameter* $\alpha > 0$*, and the loss matrix* F_k *is rank one and satisfies* $X_k^{1/2} \bullet \alpha F_k > -1$ *for all* k*, then, for every* $U \in \Delta_{n\times n}$*,*

$$R(U) \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \langle F_k, X_k - U\rangle \le \alpha \cdot \sum_{k=0}^{T-1} \frac{(X_k \bullet F_k)(X_k^{1/2} \bullet F_k)}{1 + X_k^{1/2} \bullet \alpha F_k} + \frac{V_{X_0}(U)}{\alpha} \ .$$

If we instead have $X_k^{1/2} \bullet \alpha F_k \ge -\frac{1}{2}$*, the above bound can be simplified as*

$$R(U) \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \langle F_k, X_k - U\rangle \le 2\alpha \cdot \sum_{k=0}^{T-1} (X_k \bullet F_k)(X_k^{1/2} \bullet F_k) + \frac{V_{X_0}(U)}{\alpha} \ .$$

We note that $V_{X_0}(U) \le 2\sqrt{n}$*.*

We recommend the interested readers to see the proof of Theorem 8.5 in Appendix 8.E, as it provides a straightforward generalization of Theorem 8.4 using regularizers other than entropy.

Theorem 8.5 is only a special case of the following more general regret bound, which holds for arbitrary $q \ge 2$, and for F_k having arbitrary rank. At a first reading, one can skip Theorem 8.6 because its sole purpose in this paper is to improve the running time of graph sparsification from $\tilde{O}(mn^{3/2})$ to $\tilde{O}(mn^{1+1/q})$, as well as allowing one to sparsify sums of high rank PSDs.

Theorem 8.6. *In* $\texttt{MirrorDescent}_{\ell_{1-1/\mathsf{q}}}$ *with* $q \ge 2$ *and* $\alpha > 0$*, if the loss matrix* F_k *is either positive or negative semidefinite and satisfies* $\alpha X_k^{1/2q} F_k X_k^{1/2q} \succeq -\frac{1}{2q} I$ *for all* k*, then for every* $U \in \Delta_{n\times n}$*,*

$$R(U) \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \langle F_k, X_k - U\rangle \le O(q\alpha) \sum_{k=0}^{T-1} (X_k \bullet |F_k|) \cdot \|X_k^{1/2q} F_k X_k^{1/2q}\|_{\mathsf{spe}} + \frac{V_{X_0}(U)}{\alpha} \ .$$

We note that $V_{X_0}(U) \le \frac{q}{q-1} n^{1/q}$*.*

(The proof of Theorem 8.6 is deferred to Appendix 8.E.)

The key idea to prove Theorem 8.6 is to replace the use of the Sherman-Morrison formula in the proof of Theorem 8.5 with the Woodbury formula so as to allow F_k to be of high rank. It also uses the Lieb-Thirring trace inequality to handle arbitrary $q \ge 2$.)

8.4 Warm-Up: Upper-Sided Linear-Sized Sparsification

In this section and the next, we present our construction of linear-sized sparisifier in the general matrix setting. Its specialization to graph sparsification appears in Appendix 8.B, while its efficient implementation is discussed in Section 8.6. To showcase how the regret bounds of Section 8.3 can be useful in the construction of sparsifiers, we start by describing a warm-up example in which we are only interested in

obtaining a single side of the sparsification guarantee.

Suppose we are given a decomposition of the identity matrix $I = \sum_{e=1}^{m} w_e \hat{L}_e$, where each $\hat{L}_e$ satisfies

$0 \preceq \hat{L}_e \preceq I$ and is of rank 1 and trace 1, i.e. $\hat{L}_e = vv^t$ for some $v \in \mathbb{R}^n$ with $\|v\|_2 = 1$.

The weights $w_e > 0$ may be unknown, though the trace guarantee ensures that $\sum_e w_e = n$. In this section, we are interested in finding some $s \in \Delta_m$ satisfying $\sum_{e=1}^{m}(ns_e) \cdot \hat{L}_e \preceq (1+\varepsilon)I$, while the sparsity of s —that is, $|\{e \in [m] : s_e > 0\}|$— is at most $O(n/\varepsilon^2)$. We call this the *upper-sided linear-sized spectral sparsification* because it only gives an upper bound on the eigenvalues of $\sum_{e=1}^{m}(ns_e) \cdot \hat{L}_e$ and no lower bound.

Consider the following algorithm that invokes the regret minimization framework in Section 8.3 to solve this upper-sided sparsification. We choose

$$\text{the } \ell_{1/2} \text{ regularizer and } \alpha = \varepsilon/4\sqrt{n} \text{ for } \texttt{MirrorDescent}_{\ell_{1/2}}.$$

At iteration k, set the feedback matrix as $F_k = -n\hat{L}_{e_k}$, where e_k minimizes $\hat{L}_e \bullet X_k$ over $e \in [m]$. [3]

Before applying Theorem 8.5, let us first verify that the prerequisite $X_k^{1/2} \bullet \alpha F_k \geq -\frac{1}{2}$ holds. Because $\sum_{e\in[m]} \frac{w_e}{n} \hat{L}_e \bullet X_k = \frac{1}{n} I \bullet X_k = \frac{1}{n}$, by an averaging argument, we must have $\hat{L}_{e_k} \bullet X_k \leq \frac{1}{n}$. This further implies $-\alpha n \hat{L}_{e_k} \bullet X_k^{1/2} \geq -\alpha\sqrt{n} > -\frac{1}{2}$ due to the claim below.

Claim 8.7. *For every $X \in \Delta_{n\times n}$, we have $\hat{L}_e \bullet X^{1/2} \leq (\hat{L}_e \bullet X)^{1/2}$ for every $e \in [m]$.*

Proof. Without loss of generality, one can assume X to be diagonal. Next, since $\hat{L}_e = v_e v_e^T$ is of rank one, the desired inequality follows from Jensen's inequality $v_e^T X^{1/2} v_e \leq \sqrt{v_e^T X v_e}$ and the fact that $\|v_e\|_2^2 = \mathrm{Tr}\hat{L}_e \leq 1$. □

Now, applying Theorem 8.5, we obtain that for every $U \in \Delta_{n\times n}$,

$$\sum_{k=0}^{T-1} \langle -n\hat{L}_{e_k}, X_k - U\rangle \leq 2\alpha \cdot \sum_{k=0}^{T-1} (X_k \bullet n\hat{L}_{e_k})(X_k^{1/2} \bullet n\hat{L}_{e_k}) + \frac{2\sqrt{n}}{\alpha} \ .$$

After rearranging, and using $\hat{L}_{e_k} \bullet X_k \leq \frac{1}{n}$ and $n\hat{L}_{e_k} \bullet X_k^{1/2} \leq \sqrt{n}$ we deduced earlier,

$$\begin{aligned}\left\langle \frac{n}{T}\sum_{k=0}^{T-1} \hat{L}_{e_k}, U \right\rangle &\leq \frac{2\alpha}{T} \cdot \sum_{k=0}^{T-1} (X_k \bullet n\hat{L}_{e_k})(X_k^{1/2} \bullet n\hat{L}_{e_k}) + \frac{1}{T}\sum_k \langle n\hat{L}_{e_k}, X_k\rangle + \frac{2\sqrt{n}}{\alpha T} \\ &\leq \frac{2\alpha}{T} \cdot T \cdot 1 \cdot \sqrt{n} + 1 + \frac{2\sqrt{n}}{\alpha T} = \frac{\varepsilon}{2} + 1 + \frac{8n}{\varepsilon T} \ .\end{aligned}$$

Finally, choosing $T = 16n/\varepsilon^2$ and U to be the rank-1 projection over a maximum eigenvector, we conclude that $\lambda_{\mathsf{max}}(\frac{n}{T}\sum_{k=0}^{T-1} \hat{L}_{e_k}) \leq 1 + \varepsilon$.

[3] This choice naturally follows from a saddle-point interpretation of the problem, because it is the subgradient of the function $f(X) \stackrel{\text{def}}{=} \min_{s\in\Delta_m} \sum_{e=1}^{m}(ns_e\hat{L}_e) \bullet X$ at $X = X_k$. We have skipped the explanation of this choice due to the space limitation.

This completes the description of our upper-sided linear-sized sparsification algorithm. The full sparsification algorithm, in the next section, will essentially consists of playing out this analysis on the lower and upper side at the same time.

We emphasize here that if one chooses the entropy regularizer by using $\texttt{MirrorDescent}_{\exp}$, and chooses $e_k = e$ with probability proportional to w_e, a similar analysis from the one above recovers the sparsification result of Spielman and Srivastava [151].

8.5 Linear-Sized Sparsification

As before, suppose we are given a decomposition of the identity matrix $I = \sum_{e=1}^{m} w_e \hat{L}_e$, where each $\hat{L}_e$ satisfies $0 \preceq \hat{L}_e \preceq I$ and is of rank 1 and trace 1. The weights $w_e > 0$ may be unknown and satsify $\sum_e w_e = n$. In this section, we are interested in finding scalars $s_e \geq 0$ satisfying

$$I \preceq \sum_{e=1}^{m} s_e \cdot \hat{L}_e \preceq (1 + 8\varepsilon + O(\varepsilon^2))I \ , \tag{8.6}$$

while the sparsity of s —that is, $|\{e \in [m] : s_e > 0\}|$— is at most $O(n/\varepsilon^2)$.

Instead of maintaining one sequence X_k like in Section 8.4, we maintain two sequences $X_k, Y_k \in \Delta_{n\times n}$. At each iteration $k \in 0, 1, \ldots, T-1$, find an arbitrary $e_k \in [m]$ such that

$$\hat{L}_{e_k} \bullet X_k \leq \hat{L}_{e_k} \bullet Y_k \ .$$

This is always possible by an averaging argument with weights w_e. Next, we choose the $\ell_{1/2}$ regularizer and some parameter $\alpha < 1/2$ (in fact, we will choose $\alpha = \varepsilon$ later), and updates

$$\begin{aligned} X_{k+1} &= \operatorname*{arg\,min}_{Z\in\Delta_{n\times n}} \left\{ V_{X_k}(Z) + \left\langle \frac{-\alpha \hat{L}_{e_k}}{(X_k \bullet \hat{L}_{e_k})^{1/2}}, Z \right\rangle \right\} \quad \text{and} \\ Y_{k+1} &= \operatorname*{arg\,min}_{Z\in\Delta_{n\times n}} \left\{ V_{Y_k}(Z) + \left\langle \frac{\alpha \hat{L}_{e_k}}{(Y_k \bullet \hat{L}_{e_k})^{1/2}}, Z \right\rangle \right\} \ . \end{aligned} \tag{8.7}$$

In other words, we have picked feedback matrices $F_k = \frac{-\hat{L}_{e_k}}{(X_k \bullet \hat{L}_{e_k})^{1/2}}$ for the X_k sequence and $F_k = \frac{\hat{L}_{e_k}}{(Y_k \bullet \hat{L}_{e_k})^{1/2}}$ for the Y_k sequence in our $\texttt{MirrorDescent}_{\ell_{1/2}}$.[4]

Notice that $X_k^{1/2} \bullet \frac{-\alpha \hat{L}_{e_k}}{(X_k \bullet \hat{L}_{e_k})^{1/2}} \geq -\frac{1}{2}$ due to Claim 8.7, so we always have $X_k^{1/2} \bullet \alpha F_k \geq -\frac{1}{2}$ which satisfies the prerequisite of Theorem 8.5. Applying Theorem 8.5 on the X_k sequence, we obtain that for every $U_X \in \Delta_{n\times n}$,

$$\begin{aligned} &\sum_{k=0}^{T-1} \left\langle \frac{-\hat{L}_{e_k}}{(X_k \bullet \hat{L}_{e_k})^{1/2}}, X_k - U_X \right\rangle \\ &\leq 2\alpha \cdot \sum_{k=0}^{T-1} (X_k \bullet \frac{\hat{L}_{e_k}}{(X_k \bullet \hat{L}_{e_k})^{1/2}})(X_k^{1/2} \bullet \frac{\hat{L}_{e_k}}{(X_k \bullet \hat{L}_{e_k})^{1/2}}) + \frac{V_{X_0}(U_X)}{\alpha} \end{aligned}$$

[4] In fact, the denominator $(X_k \bullet \hat{L}_{e_k})^{1/2}$ is defined so as to make sure that F_k is the 'maximally aggressive' loss matrix we can have for $\texttt{MirrorDescent}_{\ell_{1/2}}$.

$$= 2\alpha \cdot \sum_{k=0}^{T-1} X_k^{1/2} \bullet \hat{L}_{e_k} + \frac{V_{X_0}(U_X)}{\alpha} \leq 2\alpha \cdot \sum_{k=0}^{T-1} (X_k \bullet \hat{L}_{e_k})^{1/2} + \frac{V_{X_0}(U_X)}{\alpha} \ .$$

Above, the last inequality uses Claim 8.7. If we denote by $M_X \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \frac{\hat{L}_{e_k}}{(\hat{L}_{e_k} \bullet X_k)^{1/2}}$ and rearrange the inequality above, we get

$$M_X \bullet U_X \leq \frac{V_{X_0}(U_X)}{\alpha} + (1+2\alpha) \sum_{k=0}^{T-1} (\hat{L}_{e_k} \bullet X_k)^{1/2} \ . \tag{8.8}$$

Similarly, applying Theorem 8.5 on the Y_k sequence, and define $M_Y \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \frac{\hat{L}_{e_k}}{(\hat{L}_{e_k} \bullet Y_k)^{1/2}}$, we obtain that for every $U_Y \in \Delta_{n \times n}$,

$$M_Y \bullet U_Y \geq -\frac{V_{Y_0}(U_Y)}{\alpha} + (1-2\alpha) \sum_{k=0}^{T-1} (\hat{L}_{e_k} \bullet Y_k)^{1/2} \ . \tag{8.9}$$

In the rest of the proof, we will use (8.8) and (8.9) to deduce

$$\lambda_{\mathsf{max}}(M_Y) - \lambda_{\mathsf{min}}(M_Y) \leq 8\varepsilon(1+O(\varepsilon))\lambda_{\mathsf{min}}(M_Y) \ . \tag{8.10}$$

Finally, since $M_Y = \sum_{k=0}^{T-1} \frac{\hat{L}_{e_k}}{(\hat{L}_{e_k} \bullet Y_k)^{1/2}}$ is a matrix that is a summation of at most $T = n/\varepsilon^2$ rank-1 matrices, dividing it by $\lambda_{\mathsf{min}}(M_Y)$ gives the desired sparsification for (8.6).

We prove (8.10) in two steps.

Lowerbounding $\lambda_{\mathsf{min}}(M_Y)$. Recall that we have $\mathrm{Tr}(M_X) = \sum_{k=0}^{T-1} \frac{1}{(\hat{L}_e \bullet X_k)^{1/2}}$ because we have assumed each $\hat{L}_e$ to be of trace 1. Denoting by $a_k = (\hat{L}_e \bullet X_k)^{1/2}$, we have that $\mathrm{Tr}(M_X) = \sum_{k=0}^{T-1} \frac{1}{a_k}$. We apply (8.8) here with $U_X = \frac{1}{n} I = X_0$, and obtain

$$\frac{1}{n} \sum_{k=0}^{T-1} \frac{1}{a_k} = \frac{1}{n} \mathrm{Tr}(M_X) \leq (1+2\alpha) \sum_{k=0}^{T-1} (\hat{L}_e \bullet X_k)^{1/2} \leq (1+2\alpha) \sum_{k=0}^{T-1} a_k \ .$$

Applying Cauchy-Schwarz, we have

$$(\sum_{k=0}^{T-1} a_k)^2 \geq \frac{1}{n(1+2\alpha)} (\sum_{k=0}^{T-1} a_k)(\sum_{k=0}^{T-1} \frac{1}{a_k}) \geq \frac{T^2}{n(1+2\alpha)} \ . \tag{8.11}$$

If we choose $T = \frac{n}{\varepsilon^2}$, we immediately have[5]

$$\textstyle\sum_{k=0}^{T-1} (\hat{L}_e \bullet Y_k)^{1/2} \geq \sum_{k=0}^{T-1} a_k \geq \frac{\sqrt{n}}{\varepsilon^2}(1-O(\alpha)) \ .$$

Substituting the above lower bound into (8.9), and choosing $U_Y \in \Delta_{n \times n}$ to be the rank-1 projection matrix over the smallest eigenvector of M_Y, and choosing $\alpha = \varepsilon$,

[5]In fact, it suffices to stop our algorithm at the earliest iteration T so that inequality (8.11) is satisfied. Our analysis here only represents the most pessimistic scenario; in practice, this early termination implies we can choose less than n/ε^2 matrices for certain inputs. This is in contrast to [26], as their algorithm uses n/ε^2 rank-1 matrices for all inputs.

we have

$$\lambda_{\mathsf{min}}(M_Y) \geq -\frac{2\sqrt{n}}{\alpha} + (1-2\alpha)\sum_{k=0}^{T-1}(\hat{L}_e \bullet Y_k)^{1/2} \geq (1-O(\varepsilon))\frac{\sqrt{n}}{\varepsilon^2} \tag{8.12}$$

Upperbounding $\boldsymbol{\lambda_{\mathsf{max}}(M_Y) - \lambda_{\mathsf{min}}(M_Y)}$. This time, we use our choice of $\hat{L}_{e_k} \bullet X_k \leq \hat{L}_{e_k} \bullet Y_k$ to combine (8.8) and (8.9) and derive that

$$\frac{1}{1+2\alpha} M_Y \bullet U_X \leq \frac{1}{1+2\alpha} M_X \bullet U_X \leq \frac{1}{1-2\alpha} M_Y \bullet U_Y + \frac{2\sqrt{n}}{\alpha}\Big(\frac{1}{1+2\alpha} + \frac{1}{1-2\alpha}\Big) \ .$$

Choosing U_X to be the rank-1 matrix projection matrix over the largest eigenvector of M_Y, U_Y to be that over the smallest eigenvector of M_Y, and recalling that $\alpha = \varepsilon$, we have

$$\lambda_{\mathsf{max}}(M_Y) \leq \frac{1+2\varepsilon}{1-2\varepsilon}\lambda_{\mathsf{min}}(M_Y) + \frac{4\sqrt{n}}{\varepsilon}(1+O(\varepsilon)) \ .$$

After rearranging and substituting in the lower bound (8.12), we finish the proof of (8.10)

$$\lambda_{\mathsf{max}}(M_Y) - \lambda_{\mathsf{min}}(M_Y) \leq \frac{4\varepsilon}{1-2\varepsilon}\lambda_{\mathsf{min}}(M_Y) + \frac{4\sqrt{n}}{\varepsilon}(1+O(\varepsilon)) \leq 8\varepsilon(1+O(\varepsilon))\lambda_{\mathsf{min}}(M_Y) \ .$$

□

8.6 Efficient Implementation for Graph Sparsification

The update rules described in (8.7) imply that X_k and Y_k are of the form (see Section 8.3)

$$X_k = \Big(c^X \cdot I - \textstyle\sum_{j=0}^{k-1} s_j^X \hat{L}_{e_j}\Big)^{-2} \quad \text{and} \quad Y_k = \Big(\textstyle\sum_{j=0}^{k-1} s_j^Y \hat{L}_{e_j} - c^Y \cdot I\Big)^{-2} \ . \tag{8.13}$$

Here, c^X is the unique (positive) constant that satisfies $c^X I - \sum_{j=0}^{k-1} s_j^X \hat{L}_{e_j} \succ 0$ and $\mathrm{Tr} X_k = 1$, while c^Y is the unique (possibly negative) constant that satisfies $\sum_{j=0}^{k-1} s_j^Y \hat{L}_{e_j} - c^Y I \succ 0$ and $\mathrm{Tr} Y_k = 1$. The coefficients s_j^X and s_j^Y are always positive. (It is worth noting that c^X is initially $\sqrt{n}$ at X_0 and keeps increasing, while c^Y is initially $-\sqrt{n}$ and keeps increasing as well.)

Recall that $\texttt{MirrorDescent}_{\ell_{1/2}}$ requires one to compute c^X and c^Y for each iteartion, and this can be done via binary search. One way to perform binary search is to first compute $\lambda_{\mathsf{max}} = \lambda_{\mathsf{max}}(\sum_{j=0}^{k-1} s_j^X \hat{L}_{e_j})$. Then, one can binary search c^X in the range $[\lambda_{\mathsf{max}} + 1, \lambda_{\mathsf{max}} + \sqrt{n}]$ to find the correct one satisfying $\mathrm{Tr}\big(c^X \cdot I - \sum_{j=0}^{k-1} s_j^X \hat{L}_{e_j}\big)^{-2} = 1$. Similarly, one can binary search c^Y in the range of $[\lambda_{\mathsf{min}} - \sqrt{n}, \lambda_{\mathsf{min}} - 1]$ where $\lambda_{\mathsf{min}} = \lambda_{\mathsf{min}}(\sum_{j=0}^{k-1} s_j^Y \hat{L}_{e_j})$.[6]

If one performs the binary search to an accuracy that is small enough, this gives

[6] λ_{max} and λ_{min} can be computed via power methods, and it suffices to compute them up to an additive error of, say, 0.1. In Appendix 8.G, we propose an alternative approach to compute c^X and c^Y, avoiding the use of power methods.

an algorithm whose running time is $\tilde{O}(n^3 m/\varepsilon^2)$, dominated by the computation of $X_k \bullet \hat{L}_e = \big(c^X \cdot I - \sum_{j=0}^{k-1} s_j^X \hat{L}_{e_j}\big)^{-2} \bullet \hat{L}_e$ for each $k \in [T]$ and $e \in [m]$.

Running Time Improvement. For the graph sparsification problem described in Theorem 8.1, we sketch the key ideas needed to improve the running time to $\tilde{O}\big(mn^{1+1/q}/\varepsilon^5\big)$ for any even integer $q \geq 2$. The details can be found in Appendix 8.F and 8.G. In particular, we first describe how to achieve a running time of $\tilde{O}\big(mn^{1+1/2}/\varepsilon^5\big)$.

Recall that in Section 8.5, we have constructed M_X and M_Y and proved that $\lambda_{\mathsf{min}}(M_X)$ and $\lambda_{\mathsf{min}}(M_Y)$ are both at least $\Omega(\sqrt{n}/\varepsilon^2)$. In fact, it is not hard to ensure that $\lambda_{\mathsf{max}}(M_X)$ and $\lambda_{\mathsf{max}}(M_Y)$ are at most $O(\sqrt{n}/\varepsilon^2)$ as well.[7] Since $\sum_{j=0}^{k-1} s_j^X \hat{L}_{e_j} \preceq \alpha M_X$, we conclude that the eigenvalues of $\sum_{j=0}^{k-1} s_j^X \hat{L}_{e_j}$ are all upper bounded by $\alpha \cdot O(\sqrt{n}/\varepsilon^2) = O(\sqrt{n}/\varepsilon)$. Therefore, throughout the algorithm, the encountered choices of c^X are always upper bounded by $O(\sqrt{n}/\varepsilon)$.

For this reason, we only need to compute matrix inversions of the form $(cI - A)^{-1}$, with the guarantee that $c = O(\sqrt{n}/\varepsilon)$. Since we always have $cI - A \succeq I$ —as otherwise $\mathrm{Tr}(cI - A)^{-2}$ is strictly larger than 1— we can approximate this matrix inverse by

$$(cI - A)^{-1} = c^{-1}\Big(I - \frac{A}{c}\Big)^{-1} \approx c^{-1}\Big(I + \frac{A}{c} + \frac{A^2}{c^2} + \cdots \frac{A^d}{c^d}\Big) \ , \tag{8.14}$$

and it suffices to choose the maximum degree $d = O(\sqrt{n}/\varepsilon)$. This is formally proved in Lemma 8.21. In other words, when computing X_k, it suffices to replace the matrix inversion with some matrix polynomial of degree $d = O(\sqrt{n}/\varepsilon)$. Similar idea also holds for the Y_k sequence.

So far, we managed avoiding the computationally expensive matrix inversion. Next, we want to further accelerate the procedure of computing $(cI - A)^{-2} \bullet \hat{L}_e$ for all edges $e \in [m]$ simultaneously. Recall that $\hat{L}_e = v_e v_e^T$ is of rank 1, and one can rewrite

$$(cI - A)^{-2} \bullet \hat{L}_e = v_e^T (cI - A)^{-2} v_e = \|(cI - A)^{-1} v_e\|_2^2 \ .$$

For this reason, as in [151], one can apply the Johnson-Lindenstrauss dimension reduction [86]: there exists random matrix Q with $\tilde{O}(1/\varepsilon^2)$ rows, satisfying that $\|(cI - A)^{-1} v_e\|_2^2 \approx \|Q(cI - A)^{-1} v_e\|_2^2$ for for all v_e.

Using this dimension reduction, one can precompute $T = Q(cI - A)^{-1}$ in time $\tilde{O}(m/\varepsilon^2) \times \tilde{O}(\sqrt{n}/\varepsilon) = \tilde{O}(m\sqrt{n}/\varepsilon^3)$, with the help from the approximate matrix inversion (8.14), and the nearly-linear time Laplacian system solvers [152]. After the precomputation, each $(cI - A)^{-2} \bullet \hat{L}_e \approx \|T v_e\|_2^2$ can be computed in $\tilde{O}(1/\varepsilon^2)$ time, totaling $\tilde{O}(m/\varepsilon^2)$ per iteration, which is negligible.

In sum, taking into account that we have $T = n/\varepsilon^2$ iterations, the total running time is $\tilde{O}(mn^{1+1/2}/\varepsilon^5)$. To turn this $\tilde{O}(mn^{1+1/2}/\varepsilon^5)$ into $\tilde{O}(mn^{1+1/q}/\varepsilon^5)$ for any constant q, we need to replace the use of the $\ell_{1/2}$ regularizer with the $\ell_{1-1/q}$ regularizer. This requires one to use Theorem 8.6 in replacement of Theorem 8.5.

[7] This may require one to stop the algorithm earlier than $T = n/\varepsilon^2$ iterations, which is even better!

We wish to emphasize here that our analysis in Section 8.5 needs to be *strengthened* in order to tolerate all the errors incurred from the approximate computations (most notably from Laplacian linear solvers, from Johnson-Lindenstrauss, and from (8.14)). This is only rountinary thanks to the optimization motivation behind our argument, and we have done this carefully in Appendix 8.F.

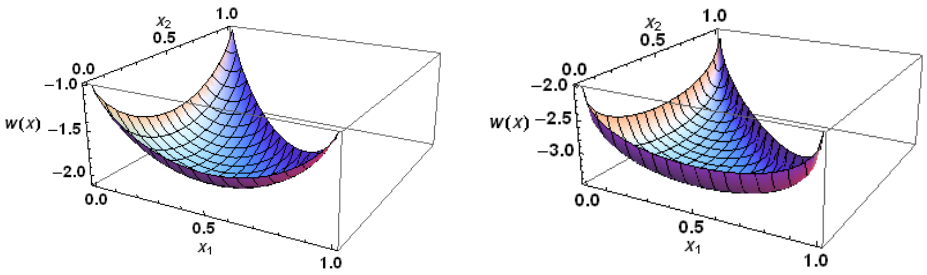
Acknowledgement

We thank Richard Peng, Nikhil Srivastava, and Nisheeth Vishnoi for helpful conversations. This material is based upon work partly supported by the National Science Foundation under Grant CCF-1319460 and by a Simons Graduate Student Award under grant no. 284059.

Appendix

Appendix roadmap.

- In Figure 8-1, we plot the entropy and the $\ell_{1/2}$ regularizers of the 3-dimensional scalar case for a visual comparison.
- In Appendix 8.A, we verify the equivalence between `FTRL` and `MirrorDescent` for our choices of the regularizers.
- In Appendix 8.B, we provide notations for graphs, and state the reduction from the sparsifying graphs to sparsifying sums of rank-1 matrices.
- In Appendix 8.C, we provide our unweighted sparsification result.
- In Appendix 8.D and 8.E we provide missing proofs for Section 8.2 and 8.3 respectively.
- In Appendix 8.F, we generalize our sparsification algorithm of Section 8.5 to allow arbitrary $q \geq 2$, high rank matrices, and approximate computations.
- In Appendix 8.G, we provide the details of how to implement linear-sized graph sparsifications in almost-quadratic time, thus finishing the running time claim of Theorem 8.1.
- In Appendix 8.H, we sketch how to generalize our running time improvement to other problems, including sparsifying sums of rank-1 PSD matrices (i.e., Theorem 8.14), as well as subgraph sparsifications.



(a) The entropy regularizer (b) The $\ell_{1/2}$ regularizer

Figure 8-1: Two regularizers in $n = 3$. The first two axes represent x_1, x_2 so $x_3 = 1 - x_1 - x_2$. The third axes represent $w(x)$.

8.A Partial Equivalence Between FTRL and Mirror Descent

In this section, we show the equivalence between mirror descent and follow-the-regularized-leader for our choices of the regularizers. In fact, this equivalence holds more generally for all regularizers $w(\cdot)$ that are convex function of Legendre type with domain Q (see for instance [22, 136]).

Letting $A_i \in \mathbb{R}^n$ be any symmetric matrix for each iteration i, the follow-the-regularized-leader method can be described as

$$\forall k = 0, 1, \ldots, T-1, \quad X_k = \operatorname*{arg\,min}_{Z \in \Delta_{n \times n}} \left\{ w(Z) + \sum_{i=0}^{k-1} \langle A_i, Z \rangle \right\} . \tag{8.15}$$

The mirror descent method (with starting point $\tilde{X}_0 = \frac{1}{n} I$) can be described as

$$\forall k = 0, 1, \ldots, T-1, \quad \tilde{X}_k = \operatorname*{arg\,min}_{Z \in \Delta_{n \times n}} \left\{ V_{\tilde{X}_{k-1}}(Z) + \langle A_{k-1}, Z \rangle \right\} , \tag{8.16}$$

where as before, $V_X(Y) \stackrel{\text{def}}{=} w(Y) - \langle \nabla w(X), Y - X \rangle - w(X)$ is the Bregman divergence of $w(\cdot)$.

Recall that when $w(X) = X \bullet (\log X - I)$ is the entropy regularizer, then $\nabla w(X) = \log X$ and therefore $(\nabla w)^{-1}(A) = e^A$. When $w(X) = -\frac{q}{q-1} \mathrm{Tr} X^{1-1/q}$ is the $\ell_{1-1/q}$ regularizer, then $\nabla w(X) = X^{-1/q}$ and therefore $(\nabla w)^{-1}(A) = A^{-q}$. The rest of the proof holds for both these two types of regularizers.

To compute the minimizer X_k for (8.15), one can take the derivative and demand that $\nabla w(X_k) + \sum_{i=0}^{k-1} A_i - c_k \cdot I = 0$. Here, the extra term $-c_k \cdot I$ comes from the Lagrange multipliers of the linear constraint $\mathrm{Tr}(Z) = I \bullet Z = 1$. (We do not have Lagrange multipliers for the other constraint $Z \succeq 0$ because our gradient $\nabla w(Z)$ is a barrier function and tends to infinite as any eigenvalue of Z tends to zero.) It is now easy to see that c_k is the unique constant that ensures $\sum_{i=0}^{k-1} A_i - c_k I \preceq 0$ (because $\nabla w(X_k) \succeq 0$) and that $\mathrm{Tr} X_k = \mathrm{Tr}\big((\nabla w)^{-1}(c_k I - \sum_{i=0}^{k-1} A_i)\big) = 1$.

To compute the minimizer X_k for (8.16), one can take the derivative and demand

that $\nabla w(\tilde{X}_k) - \nabla w(\tilde{X}_{k-1}) + A_i - d_k \cdot I = \nabla V_{\tilde{X}_{k-1}}(\tilde{X}_k) + A_i - d_k \cdot I = 0$. Here, the extra term $-d_k \cdot I$ again comes from the Lagrange multipliers of the linear constraint $\mathrm{Tr}(Z) = I \bullet Z = 1$. It is now easy to see that d_k is the unique constant that ensures $-\nabla w(\tilde{X}_{k-1}) + A_i - d_k \cdot I \preceq 0$ (because $\nabla w(\tilde{X}_k) \succeq 0$) and that $\mathrm{Tr}\tilde{X}_k = \mathrm{Tr}\big((\nabla w)^{-1}(\nabla w(\tilde{X}_{k-1}) + d_k I - A_{k-1})\big) = 1$.

To show the equivalence between (8.15) and (8.16), we perform a simple induction. Suppose that $\tilde{X}_{k-1} = X_{k-1}$, and we wish to prove $\tilde{X}_k = X_k$.

In this case, we have

$$\begin{aligned}\tilde{X}_k &= (\nabla w)^{-1}\big(\nabla w(\tilde{X}_{k-1}) + d_k I - A_{k-1}\big) = (\nabla w)^{-1}\big(\nabla w(X_{k-1}) + d_k I - A_{k-1}\big) \\ &= (\nabla w)^{-1}\Big(c_{k-1} I + d_k I - \sum_{i=0}^{k-1} A_i\Big) \ , \text{ and} \\ X_k &= (\nabla w)^{-1}\Big(c_k I - \sum_{i=0}^{k-1} A_i\Big) \ .\end{aligned}$$

Finally, since d_k is the unique constant that ensures $c_{k-1} I + d_k I - \sum_{i=0}^{k-1} A_i \succeq 0$ and $\mathrm{Tr}\big((\nabla w)^{-1}\big(c_{k-1} I + d_k I - \sum_{i=0}^{k-1} A_i\big)\big) = 1$, while c_k is the unique constant that ensures $c_k I - \sum_{i=0}^{k-1} A_i \succeq 0$ and $\mathrm{Tr}\big((\nabla w)^{-1}\big(c_k I - \sum_{i=0}^{k-1} A_i\big)\big) = 1$, it is obvious to see that $c_k = c_{k-1} + d_k$ and therefore $\tilde{X}_k = X_k$.

8.B Graph Notations

Let $G = (V, E, w)$ be a undirected weighted graph with n vertices and m edges, and each $w_e > 0$ is the weight of edge e. Without loss of generality, we study only connected graphs throughout this paper. For every edge $e = (a, b) \in E$, we orient it arbitrarily and denote by $\chi_e \stackrel{\text{def}}{=} \mathbf{e}_a - \mathbf{e}_b \in \mathbb{R}^n$ the characteristic (column) vector of edge e.

Let $L_e \stackrel{\text{def}}{=} w_e \chi_e \chi_e^T \in \mathbb{R}^{n \times n}$ be the graph Laplacian of edge e, or the edge Laplacian. Let $B \in \mathbb{R}^{m \times n}$ be the incidence matrix where its row corresponding to edge e is the characteristic (row) vector χ_e^T. Define $W = \mathrm{diag}\{w_e\}_{e \in E}$ to be the diagonal matrix of edge weights. The Laplacian with respect to graph G is $L_G \stackrel{\text{def}}{=} B^T W B \in \mathbb{R}^{n \times n}$. It is clear from the definition that $L_G \succeq 0$ is PSD and $L_G = \sum_{e \in E} L_e$. Notice that $\ker(L_G) = \ker(W^{1/2} B) = \mathrm{span}(\mathbb{1})$, and therefore $x^T L_G x = 0$ if and only if x is a constant vector.

Since L_G is symmetric, one can diagonalize it and write $L_G = \sum_{i=1}^{n-1} \lambda_i v_i v_i^T$, where λ_i's are the positive eigenvalues of L_G and v_i's are the corresponding set of orthogonal eigenvectors. The Moore-Penrose pseudoinverse of L_G is denoted by $L_G^\dagger \stackrel{\text{def}}{=} \sum_{i=1}^{n-1} \lambda_i^{-1} v_i v_i^T$. For notational convenience, we will stick to L_G^{-1} to denote this pseudoinverse, and often use L_G^{-2} to denote $(L_G^\dagger)^2$, and $L_G^{-1/2}$ to denote $(L_G^\dagger)^{1/2}$, and so on. We remark here that $L_G L_G^{-1} = L_G^{-1} L_G = \sum_i v_i v_i^T = I_{\mathsf{im}(L_G)}$. Here, $I_{\mathsf{im}(L_G)}$ is the identity matrix on the image space of L_G, which is just the space spanned by

all the vectors orthogonal to $\mathbb{1}$. For notational convenience, we will often abbreviate $I_{\mathsf{im}(L_G)}$ as I.

Throughout this paper, whenever related to graph sparsifications, we denote by

$$\check{L}_e \stackrel{\text{def}}{=} L_G^{-1/2} L_e L_G^{-1/2} \quad \text{and} \quad \hat{L}_e \stackrel{\text{def}}{=} \frac{L_G^{-1/2} L_e L_G^{-1/2}}{L_G^{-1} \bullet L_e} = \frac{\check{L}_e}{L_G^{-1} \bullet L_e} \ .$$

Above, $\check{L}_e$ is the *normalized edge Laplacian*, and $\hat{L}_e$ is the *normalized edge Laplacian scaled by the effective resistance*. ($L_G^{-1} \bullet L_e$ is the "effective resistance" of the edge e, see for instance [151]). Both of them have rank 1, and it satisfies $\mathrm{Tr}(\check{L}_e) \leq 1$ and $\check{L}_e \preceq I$, while $\mathrm{Tr}(\hat{L}_e) = 1$ and $\hat{L}_e \preceq I$.

It is easy to check from the above definition that $\sum_e \check{L}_e = I_{\mathsf{im}(L_G)}$. In addition, letting $w_e = L_G^{-1} \bullet L_e$ be the effective resistence of edge e, then $\sum_e w_e \hat{L}_e = I_{\mathsf{im}(L_G)}$ as well. Notice that $\sum_e w_e = \mathrm{Tr} I_{\mathsf{im}(L_G)} = n - 1$, the dimension of $I_{\mathsf{im}(L_G)}$ (see [151]).

From Graph Sparsification to Rank-1 Decomposition Sparsification. As originally shown in [26], one can easily translate the problem of graph spectral sparsification (see Theorem 8.1) into that of sparsifying sums of rank-1 matrices (see Theorem 8.2). Indeed, because $I_{\mathsf{im}(L_G)} = \sum_{e \in [m]} \check{L}_e$ is a summation of rank-1 matrices, if one can find scalars $s_e \geq 0$ (as per Theorem 8.2) that satisfies $I_{\mathsf{im}(L_G)} \preceq \sum_{e \in [m]} s_e \check{L}_e \preceq (1+\varepsilon) I_{\mathsf{im}(L_G)}$, this immediately implies, by the definition of $\check{L}_e$, that $L_G \preceq \sum_{e \in [m]} s_e L_e \preceq (1+\varepsilon) L_G$.

8.C Weak Unweighted Sparsifier

In this section, we consider the weak *unweighted spectral sparsification* problem very recently studied by Anderson, Gu and Melgaard [8]: for any value $\kappa \in [1, m/n]$, find a κ-spectral sparsifier of G containing $O(m/\kappa)$ distinct edges from E, *without reweighting*. We show that our regret minimization framework allows us to design a simple and almost-quadratic-time algorithm for this problem, improving from the quartic time complexity of [8].

Formally, given any weighted undirected graph $G = (V, E, w)$ with n vertices and m edges, and any value $\kappa \in [1, m/n]$, the task it to find a subset $E_0 \subseteq E$ containing $O(m/\kappa)$ distinct edges such that

$$\frac{1}{\kappa} L_G \preceq \sum_{e \in E_0} L_e \preceq L_G \ .$$

This is an unweighted sparsification problem because one is not allowed to reweight the edges in E_0, in contrast to Theorem 8.1; and we call it a weak sparsifier because κ is usually large.

Similar to Appendix 8.B, one can easily reduce this graph sparsification problem to sparsifying sums of rank-1 matrices. Given m rank-1 PSD matrices $\check{L}_1, \ldots, \check{L}_m \in \mathbb{R}^{n \times n}$ that satisfies $I = \sum_{e \in [m]} \check{L}_e$, and given some $\kappa \in [1, m/n]$, find a subset $E_0 \subseteq [m]$ with $O(m/\kappa)$ distinct elements satisfying $\sum_{e \in E_0} \check{L}_e \succeq \frac{1}{\kappa} I$.

(In this section, one should feel free to coincide this $\check{L}_e$ with the 'normalized edge Laplacian' introduced in Section 8.B; but $\hat{L}_e$ needs not coincide with any graph Laplacian in general.)

We solve this weak unweighted sparsification problem via the following reduction to regret minimization.

If $\kappa \leq 9$, we output $E_0 = E$ and are done. Otherwise, we choose the $\ell_{1/2}$ regularizer and parameter $\alpha = 4\sqrt{n}\kappa$ for $\texttt{MirrorDescent}_{\ell_{1/2}}$. At each iteration $k = 0, 1, \ldots, T-1$, we define $e_k = e$ to be the index $e \in [m]$ that maximizes the quantity $\frac{X_k \bullet \check{L}_e}{1+X_k^{1/2} \bullet \alpha \check{L}_e}$ among all edges not chosen before —i.e., all edges in $E \setminus \{e_0, e_1, \ldots, e_{k-1}\}$. Next, we feed $F_k = \check{L}_{e_k}$ as the feedback matrix to $\texttt{MirrorDescent}_{\ell_{1/2}}$, and compute X_{k+1} of the next iteration.

Let us now state a simple property for the selected matrix $\check{L}_{e_k}$ using an averaging argument:

Claim 8.8. *For each $k = 0, 1, \ldots, T-1$, we either have $\sum_{j=0}^{k-1} \check{L}_{e_j} \succeq \frac{1}{\kappa} I$ or $\frac{X_k \bullet \check{L}_{e_k}}{1+X_k^{1/2} \bullet \alpha \check{L}_{e_k}} \geq \frac{1}{6m}$.*

Proof. Let us recall that by the definition of $\texttt{MirrorDescent}_{\ell_{1/2}}$, we have

$$X_k = \Big(\alpha \sum_{j=0}^{k-1} \check{L}_{e_j} - c_k I\Big)^{-2} \ ,$$

where $c_k > 0$ is the unique constant that makes $\alpha \sum_{j=0}^{k-1} \check{L}_{e_j} \succ c_k I$ and $\mathrm{Tr} X_k = 1$. Note that if $c_k/\alpha \geq \frac{1}{\kappa}$ then we already have $\sum_{j=0}^{k-1} \check{L}_{e_j} \succ \frac{c_k}{\alpha} I \succeq \frac{1}{\kappa} I$. Therefore, we can assume $c_k/\alpha < \frac{1}{\kappa}$ for the rest of the proof.

One one hand, we have

$$\begin{aligned} \sum_{e \notin \{e_0, \ldots, e_{k-1}\}} X_k \bullet \check{L}_e = X_k \bullet \Big(I - \sum_{j=0}^{k-1} \check{L}_{e_j}\Big) &= X_k \bullet \Big(I - \frac{c_k}{\alpha} I - \frac{X_k^{-1/2}}{\alpha}\Big) \\ &= (1 - \frac{c_k}{\alpha}) - \frac{\mathrm{Tr} X_k^{1/2}}{\alpha} > 1 - \frac{1}{\kappa} - \frac{\sqrt{n}}{\alpha} > \frac{5}{6} \ , \end{aligned} \tag{8.17}$$

where the first inequality is due to $\mathrm{Tr} X_k^{1/2} \leq \sqrt{n}$ and the second inequality is due to our choice of $\alpha = 4\sqrt{n}\kappa$ and the fact that $\kappa > 9$.

On the other hand, we have

$$\begin{aligned} \sum_{e \notin \{e_0, \ldots, e_{k-1}\}} \frac{1}{6m}\big(1 + X_k^{1/2} \bullet \alpha \check{L}_e\big) &\leq \frac{1}{6} + \frac{\alpha}{6m} X_k^{1/2} \bullet \sum_{e \notin \{e_0, \ldots, e_{k-1}\}} \alpha \check{L}_e \\ &\leq \frac{1}{6} + \frac{\alpha}{6m} X_k^{1/2} \bullet I \leq \frac{1}{6} + \frac{\alpha\sqrt{n}}{6m} = \frac{1}{6} + \frac{4n\kappa}{6m} \leq \frac{5}{6} \ , \end{aligned} \tag{8.18}$$

where the second inequality is because $\sum_{e \notin \{e_0, \ldots, e_{k-1}\}} \check{L}_e \preceq \sum_{e \in [m]} \check{L}_e = I$, the third inequality is because $\mathrm{Tr} X_k^{1/2} \leq \sqrt{n}$, and the fourth inequality is because $\kappa \leq m/n$.

Combining (8.17) and (8.18), we conclude that there exists at least some index

$e \in [m] \setminus \{e_0, \ldots, e_{k-1}\}$ satisfying that $X_k \bullet \check{L}_e \geq \frac{1}{7m}\big(1 + X_k^{1/2} \bullet \alpha \check{L}_e\big)$, finishing the proof of the claim. □

Now we are ready to apply Theorem 8.5, the regret bound, with our choice of $F_k = \check{L}_{e_k}$:

$$\begin{aligned}
\forall U \in \Delta_{n\times n}, \qquad \sum_{k=0}^{T-1} \langle \check{L}_{e_k}, U\rangle &\geq \sum_{k=0}^{T-1} \langle \check{L}_{e_k}, X_k\rangle - \alpha \frac{\mathrm{Tr}(X_k \check{L}_{e_k} X_k^{1/2} \check{L}_{e_k})}{1 + X_k^{1/2} \bullet \alpha \check{L}_{e_k}} - \frac{2\sqrt{n}}{\alpha} \\
&= \sum_{k=0}^{T-1} \langle \check{L}_{e_k}, X_k\rangle \Big(1 - \frac{X_k^{1/2} \bullet \alpha \check{L}_{e_k}}{1 + X_k^{1/2} \bullet \alpha \check{L}_{e_k}}\Big) - \frac{2\sqrt{n}}{\alpha} \\
&= \sum_{k=0}^{T-1} \frac{\check{L}_{e_k} \bullet X_k}{1 + X_k^{1/2} \bullet \alpha \check{L}_{e_k}} - \frac{2\sqrt{n}}{\alpha} \ . \qquad (8.19)
\end{aligned}$$

We will now choose $T = 9m/\kappa$. (Notice that $T < m$ because $\kappa > 9$.) There are two possibilities according to Claim 8.8.

In the first case, we have $\sum_{j=0}^{k-1} \check{L}_{e_j} \succeq \frac{1}{\kappa} I$ for some $k = 0, 1, \ldots, T-1$ and we are done: that is, defining $E_0 \stackrel{\text{def}}{=} \{e_0, e_1, \ldots, e_{k-1}\}$, we have that $|E_0| \leq T = O(m/\kappa)$ and $I \succeq \sum_{e\in E_0} \check{L}_e \succeq \frac{1}{\kappa} I$.

In the second case, we have $\frac{X_k \bullet \check{L}_{e_k}}{1 + X_k^{1/2} \bullet \alpha \check{L}_{e_k}} \geq \frac{1}{6m}$ for all $k = 0, 1, \ldots, T-1$. Substituting this into (8.19), and choosing U to be the rank 1 matrix corresponding to the smallest eigenvalue of $\sum_{k=0}^{T-1} \check{L}_{e_k}$, we conclude that

$$\lambda_{\min}\Big(\sum_{k=0}^{T-1} \check{L}_{e_k}\Big) \geq \sum_{k=0}^{T-1} \frac{1}{6m} - \frac{1}{2\kappa} = \frac{1}{\kappa} \ .$$

Therefore, defining $E_0 \stackrel{\text{def}}{=} \{e_0, e_1, \ldots, e_{T-1}\}$, we also have $|E_0| = T = O(m/\kappa)$ and $I \succeq \sum_{e\in E_0} \check{L}_e \succeq \frac{1}{\kappa} I$. In sum,

Theorem 8.9. *Given a decomposition $I = \sum_{e\in[m]} \check{L}_e$ of rank-1 PSD matrices, and given some $\kappa \in [1, m/n]$, the above algorithm finds a subset $E_0 \subseteq [m]$ with $O(\frac{m}{\kappa})$ distinct elements satisfying $I \succeq \sum_{e\in E_0} \check{L}_e \succeq \frac{1}{\kappa} I$.*

We remark here that for graph sparsification, the above algorithm can be implemented to run in time $\tilde{O}(m^{3/2}n)$, and can be improved to $\tilde{O}(m^{1+1/q}n)$ for any even integer constant $q \geq 2$ if the $\ell_{1-1/q}$ regularizer is used instead of $\ell_{1/2}$. We ignore the implementation details in this version of the paper because it is very similar to the details discussed in Section 8.6.

8.D Proof of Lemma 8.3

We state some classical properties for Bregman divergence, which are classical and can be found in for instance [40].

Lemma 8.3. *The following properties hold for Bregman divergence.*

- *Non-negativity:* $V_X(Y) \geq 0$ *for all* $X, Y \geq 0$.
- *The "three-point equality":* $\langle \nabla w(X) - \nabla w(Y), X - U \rangle = V_X(U) - V_Y(U) + V_Y(X)$.
- *Given* $\tilde{X} \succeq 0$ *and* $X = \arg\min_{Z \in \Delta_{n\times n}} V_{\tilde{X}}(Z)$ *as the* Bregman projection*, we have the "generalized Pythagorean theorem" for all* $U \in \Delta$*:* $V_{\tilde{X}}(U) \geq V_X(U) + V_{\tilde{X}}(X) \geq V_X(U)$.

Proof. The non-negativity follows by definition from the convexity of $w(X)$. For every $U \succeq 0$, the "three-point equality" follows from the following inequality.

$$\begin{aligned}
&\langle \nabla w(Y) - \nabla w(Y), Y - U \rangle \\
&= (w(U) - w(Y) - \langle \nabla w(Y), U - Y \rangle) - (w(U) - w(Y) - \langle w(Y), U - Y) \rangle) \\
&\quad - (w(Y) - w(Y) - \langle \nabla w(Y), Y - Y \rangle) \\
&= V_Y(U) - V_Y(U) - V_Y(Y) \ .
\end{aligned}$$

For the generalized Pythagorean theorem, we only need to prove $V_{\tilde{X}}(U) \geq V_X(U) + V_{\tilde{X}}(X)$ because the second inequality follows from the non-negativity of $V_{\tilde{X}}(X)$. To provide the simplest proof, we only focus on the special case when $w(X) = -\frac{q}{q-1}\mathrm{Tr}X^{1-1/q}$. (The proof for the entropy regularizer is similar, while the proof for the most general Legendre function case is more involved.)

By definition,

$$\begin{aligned}
V_X(U) + V_{\tilde{X}}(X) = X^{-1/q} \bullet U + \frac{1}{q-1}\mathrm{Tr}X^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}U^{1-1/q} \\
+ \tilde{X}^{-1/q} \bullet X + \frac{1}{q-1}\mathrm{Tr}\tilde{X}^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}X^{1-1/q} \\
V_{\tilde{X}}(U) = \tilde{X}^{-1/q} \bullet U + \frac{1}{q-1}\mathrm{Tr}\tilde{X}^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}U^{1-1/q} \ .
\end{aligned}$$

Therefore,

$$\begin{aligned}
V_{\tilde{X}}(U) - (V_X(U) + V_{\tilde{X}}(X)) &= \tilde{X}^{-1/q} \bullet U - X^{-1/q} \bullet U - \tilde{X}^{-1/q} \bullet X + \mathrm{Tr}X^{1-1/q} \\
&= (\tilde{X}^{-1/q} - X^{-1/q}) \bullet (U - X) \ .
\end{aligned}$$

Since $V_{\tilde{X}}(U)$ is a convex function and $X = \arg\min_{z \in \Delta} V_{\tilde{X}}(z)$, for any $U \in \Delta_{n\times n}$ we must have

$$\langle \nabla V_{\tilde{X}}(X), U - X \rangle \geq 0 \Longleftrightarrow \langle -X^{-1/q} + \tilde{X}^{-1/q}, U - X \rangle \geq 0 \ .$$

This concludes the proof of the lemma. □

8.E Missing Proofs in Section 8.3

Theorem 8.4. *In* $\texttt{MirrorDescent}_{\mathsf{exp}}$*, if the parameter* $\alpha > 0$ *satisfies* $\alpha F_k \succeq -I$ *for all iterations* $k = 0, 1, \ldots, T-1$*, then, for every* $U \in \Delta_{n\times n}$*,*

$$R(U) \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \langle F_k, X_k - U\rangle \le \alpha \sum_{k=0}^{T-1} \big(X_k \bullet |F_k|\big) \cdot \|F_k\|_{\mathsf{spe}} + \frac{V_{X_0}(U)}{\alpha} \ .$$

We note that $V_{X_0}(U) \le \log n$*.*

Proof. We prove the theorem by using a two-step description of the mirror descent. For every $k \ge 0$, define $\tilde{X}_{k+1} \stackrel{\text{def}}{=} \arg\min_{Z\succeq 0}\{V_{X_k}(Z) + \alpha\langle F_k, Z\rangle\}$, where the minimization is over all $Z \succeq 0$, rather than $Z \in \Delta_{n\times n}$. This minimizer $\tilde{X}_{k+1}$ certainly exists (and equals to $\exp^{\log X_k - \alpha F_k}$), and it is not hard to verify that $X_{k+1} = \arg\min_{Z\in\Delta_{n\times n}}\{V_{\tilde{X}_{k+1}}(Z)\}$. In other words, one can describe the update $X_k \to X_{k+1}$ by adding an intermediate stage $X_k \to \tilde{X}_{k+1} \to X_{k+1}$. We also assume that initially we have $\tilde{X}_0 \stackrel{\text{def}}{=} X_0$.

Noticing that the definition of $\tilde{X}_{k+1}$ implies that $\nabla V_{X_k}(\tilde{X}_{k+1}) + \alpha F_k = 0$, which by the definition of $V_X(Y)$ is equivalent to $\nabla w(X_k) - \nabla w(\tilde{X}_{k+1}) = \alpha F_k$. Therefore,

$$\begin{aligned}\langle \alpha F_k, X_k - U\rangle = \langle \nabla w(X_k) - \nabla w(\tilde{X}_{k+1}), X_k - U\rangle &= V_{X_k}(U) - V_{\tilde{X}_{k+1}}(U) + V_{\tilde{X}_{k+1}}(X_k) \\ &\le V_{\tilde{X}_k}(U) - V_{\tilde{X}_{k+1}}(U) + V_{\tilde{X}_{k+1}}(X_k) \ .\end{aligned} \tag{8.20}$$

Above, the second equality is due to the three-point equality and the only inequality is due to the generalized Pythagorean theorem of Bregman divergence (see Lemma 8.3). Now,

$$\begin{aligned} V_{\tilde{X}_{k+1}}(X_k) &= X_k \bullet (\log X_k - \log \tilde{X}_{k+1}) + \mathrm{Tr}\tilde{X}_{k+1} - \mathrm{Tr}X_k \\ &= X_k \bullet \alpha F_k + \mathrm{Tr}\big(e^{\log X_k - \alpha F_k}\big) - \mathrm{Tr}X_k \overset{①}{\le} X_k \bullet \alpha F_k + X_k \bullet e^{-\alpha F_k} - \mathrm{Tr}X_k \\ &\overset{②}{\le} X_k \bullet \alpha F_k + X_k \bullet (I - \alpha F_k + \alpha^2 F_k^2) - \mathrm{Tr}X_k = \alpha^2 \cdot X_k \bullet F_k^2 \\ &\overset{③}{\le} \alpha^2 \cdot (X_k \bullet |F_k|)\|F_k\|_{\mathsf{spe}} \ . \end{aligned}$$

Above, ① is due to the Golden-Thompson inequality. ② follows because $e^{-\alpha A} \preceq I - \alpha A + \alpha^2 A^2$, which can be proved after transforming into its eigenbasis, and then using the fact that $e^{-a} \le 1 - a + a^2$ for all $a \ge -1$. ③ follows because $F_k^2 \preceq \|F_k\|_{\mathsf{spe}} \cdot |F_k|$.

Finally, substituting the above upper bound into (8.20) and telescoping it for $k = 0, \ldots, T-1$, we obtain

$$\sum_{k=0}^{T-1} \langle F_k, X_k - U\rangle \le \frac{V_{\tilde{X}_0}(U) - V_{\tilde{X}_T}(U)}{\alpha} + \alpha \sum_{k=0}^{T-1} (X_k \bullet |F_k|)\|F_k\|_{\mathsf{spe}} \ .$$

The desired result of this theorem now follows from the above inequality and the simple upper bound $V_{\tilde{X}_0}(U) = V_{X_0}(U) \le \log n$ and the nonnegativity $V_{\tilde{X}_T}(U) \ge 0$. □

Theorem 8.5. *In* $\texttt{MirrorDescent}_{\ell_{1/2}}$*, if the parameter* $\alpha > 0$*, and the loss matrix* F_k *is rank one and satisfies* $X_k^{1/2} \bullet \alpha F_k > -1$ *for all* k*, then, for every* $U \in \Delta_{n\times n}$*,*

$$R(U) \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \langle F_k, X_k - U\rangle \le \alpha \cdot \sum_{k=0}^{T-1} \frac{(X_k \bullet F_k)(X_k^{1/2} \bullet F_k)}{1 + X_k^{1/2} \bullet \alpha F_k} + \frac{V_{X_0}(U)}{\alpha} \ .$$

If we instead have $X_k^{1/2} \bullet \alpha F_k \ge -\frac{1}{2}$*, the above bound can be simplified as*

$$R(U) \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \langle F_k, X_k - U\rangle \le 2\alpha \cdot \sum_{k=0}^{T-1} (X_k \bullet F_k)(X_k^{1/2} \bullet F_k) + \frac{V_{X_0}(U)}{\alpha} \ .$$

We note that $V_{X_0}(U) \le 2\sqrt{n}$*.*

Proof. We prove the theorem by using a two-step description of the mirror descent. For every $k \ge 0$, define $\tilde{X}_{k+1} \stackrel{\text{def}}{=} \arg\min_{Z \succeq 0}\{V_{X_k}(Z) + \alpha\langle F_k, Z\rangle\}$, where the minimization is over all $Z \succeq 0$, rather than $Z \in \Delta_{n\times n}$. We claim that this minimizer $\tilde{X}_{k+1}$ exists and is strictly positive definite, because one can choose $Z = \tilde{X}_{k+1} = (X_k^{-1/2} + \alpha F_k)^{-2} \succ 0$ to make the gradient zero:

$$\nabla V_{X_k}\big(\tilde{X}_{k+1}\big) + \alpha F_k = \nabla w(\tilde{X}_{k+1}) - \nabla w(X_k) + \alpha F_k = -\tilde{X}_{k+1}^{-1/2} + X_k^{-1/2} + \alpha F_k = 0 \ . \tag{8.21}$$

This uses our assumption $X_k^{1/2} \bullet \alpha F_k > -1$ which is equivalent to $\alpha F_k \succ -X_k^{-1/2}$,[8] so as to ensure that $\tilde{X}_{k+1}$ is well defined.

Next, it is easy to verify that $X_{k+1} = \arg\min_{Z\in\Delta_{n\times n}}\{V_{\tilde{X}_{k+1}}(Z)\}$. In other words, one can describe the update $X_k \to X_{k+1}$ by adding an intermediate stage $X_k \to \tilde{X}_{k+1} \to X_{k+1}$. We assume for notational simplicity that $\tilde{X}_0 \stackrel{\text{def}}{=} X_0$.

Using (8.21), we easily obtain that

$$\begin{aligned}\langle \alpha F_k, X_k - U\rangle = \langle \nabla w(X_k) - \nabla w(\tilde{X}_{k+1}), X_k - U\rangle &= V_{X_k}(U) - V_{\tilde{X}_{k+1}}(U) + V_{\tilde{X}_{k+1}}(X_k) \\ &\le V_{\tilde{X}_k}(U) - V_{\tilde{X}_{k+1}}(U) + V_{\tilde{X}_{k+1}}(X_k) \ .\end{aligned} \tag{8.22}$$

Above, the second equality is due to the three-point equality and the only inequality is due to the generalized Pythagorean theorem of Bregman divergence (see Lemma 8.3).

We now exactly compute $V_{\tilde{X}_{k+1}}(X_k)$ in two cases.

- If $\alpha F_k = -uu^T$ is negative semidefinite, using the Sherman-Morrison formula,

$$\mathrm{Tr}\tilde{X}_{k+1}^{1/2} = \mathrm{Tr}\big((X_k^{-1/2} - uu^T)^{-1}\big) = \mathrm{Tr}\Big(X_k^{1/2} + \frac{X_k^{1/2}uu^TX_k^{1/2}}{1 - u^TX_k^{1/2}u}\Big) \ .$$

  Therefore,

$$V_{\tilde{X}_{k+1}}(X_k) = \tilde{X}_{k+1}^{-1/2} \bullet X_k + \mathrm{Tr}\tilde{X}_{k+1}^{1/2} - 2\mathrm{Tr}X_k^{1/2}$$

[8] This is because, if $F_k = -uu^T$, then $X_k^{1/2} \bullet (-\alpha uu^T) > -1$ is equivalent to $\alpha u^T X_k^{1/2} u < 1$, which is further equivalent to $\alpha \mathrm{Tr} X_k^{1/4} uu^T X_k^{1/4} < 1$. However, since $X_k^{1/4} uu^T X_k^{1/4}$ is a rank-1 matrix, this is finally equivalent to $\alpha uu^T \prec X_k^{-1/2}$.

$$
\begin{aligned}
&= (X_k^{-1/2} - uu^T) \bullet X_k + \mathrm{Tr}\tilde{X}_{k+1}^{1/2} - 2\mathrm{Tr}X_k^{1/2} \\
&= -uu^T \bullet X_k + \big(\mathrm{Tr}\tilde{X}_{k+1}^{1/2} - \mathrm{Tr}X_k^{1/2}\big) = -u^T X_k u + \frac{u^T X_k u}{1 - u^T X_k^{1/2} u} \\
&= \frac{u^T X_k u \cdot u^T X_k^{1/2} u}{1 - u^T X_k^{1/2} u} = \alpha^2 \frac{(X_k \bullet F_k)(X_k^{1/2} \bullet F_k)}{1 + X_k^{1/2} \bullet \alpha F_k} \ .
\end{aligned}
$$

- If $\alpha F_k = uu^T$ is positive semidefinite, using the Sherman-Morrison formula,

$$
\mathrm{Tr}\tilde{X}_{k+1}^{1/2} = \mathrm{Tr}\big((X_k^{-1/2} + uu^T)^{-1}\big) = \mathrm{Tr}\Big(X_k^{1/2} - \frac{X_k^{1/2} uu^T X_k^{1/2}}{1 + u^T X_k^{1/2} u}\Big) \ .
$$

  Therefore,

$$
\begin{aligned}
V_{\tilde{X}_{k+1}}(X_k) &= \tilde{X}_{k+1}^{-1/2} \bullet X_k + \mathrm{Tr}\tilde{X}_{k+1}^{1/2} - 2\mathrm{Tr}X_k^{1/2} \\
&= (X_k^{-1/2} + uu^T) \bullet X_k + \mathrm{Tr}\tilde{X}_{k+1}^{1/2} - 2\mathrm{Tr}X_k^{1/2} \\
&= uu^T \bullet X_k + \big(\mathrm{Tr}\tilde{X}_{k+1}^{1/2} - \mathrm{Tr}X_k^{1/2}\big) = u^T X_k u + \frac{u^T X_k u}{1 + u^T X_k^{1/2} u} \\
&= \frac{u^T X_k u \cdot u^T X_k^{1/2} u}{1 + u^T X_k^{1/2} u} = \alpha^2 \frac{(X_k \bullet F_k)(X_k^{1/2} \bullet F_k)}{1 + X_k^{1/2} \bullet \alpha F_k} \ .
\end{aligned}
$$

Finally, substituting the above computation of $V_{\tilde{X}_{k+1}}(X_k)$ into (8.22) and telescoping it for $k = 0, \ldots, T-1$, we obtain

$$
\sum_{k=0}^{T-1} \langle F_k, X_k - U \rangle \le \frac{V_{\tilde{X}_0}(U) - V_{\tilde{X}_T}(U)}{\alpha} + \alpha \sum_{k=0}^{T-1} \frac{(X_k \bullet F_k)(X_k^{1/2} \bullet F_k)}{1 + X_k^{1/2} \bullet \alpha F_k} \ .
$$

The desired result of this theorem now follows from the above inequality and the simple upper bound $V_{\tilde{X}_0}(U) = V_{X_0}(U) \le 2\sqrt{n}$ and the nonnegativity $V_{\tilde{X}_T}(U) \ge 0$. □

The next theorem generalizes Theorem 8.5 to high rank loss matrices and $\ell_{1-1/q}$-regularizers with $q \ge 2$. The key idea is to replace the use of the Sherman-Morrison formula in the proof of Theorem 8.5 with the Woodbury formula so as to allow F_k to be of high rank. It also uses the Lieb-Thirring trace inequality to handle arbitrary $q \ge 2$.

Theorem 8.6. *In* $\texttt{MirrorDescent}_{\ell_{1-1/q}}$ *with* $q \ge 2$ *and* $\alpha > 0$*, if the loss matrix* F_k *is either positive or negative semidefinite and satisfies* $\alpha X_k^{1/2q} F_k X_k^{1/2q} \succeq -\frac{1}{2q} I$ *for all* k*, then,*

$$
\forall U \in \Delta_{n\times n}, \quad R(U) \stackrel{\mathrm{def}}{=} \sum_{k=0}^{T-1} \langle F_k, X_k - U \rangle \le O(q\alpha) \sum_{k=0}^{T-1} (X_k \bullet |F_k|) \cdot \|X_k^{1/2q} F_k X_k^{1/2q}\|_{\mathsf{spe}} + \frac{V_{X_0}(U)}{\alpha} \ .
$$

We note that $V_{X_0}(U) \le \frac{q}{q-1} n^{1/q}$.

Proof. We prove the theorem by using a two-step description of the mirror descent. For every $k \geq 0$, define $\tilde{X}_{k+1} \stackrel{\text{def}}{=} \arg\min_{Z \succeq 0}\{V_{X_k}(Z) + \alpha\langle F_k, Z\rangle\}$, where the minimization is over all $Z \succeq 0$, rather than $Z \in \Delta_{n\times n}$. We claim that this minimizer $\tilde{X}_{k+1}$ exists and is strictly positive definite, because one can choose $Z = \tilde{X}_{k+1} = (X_k^{-1/q} + \alpha F_k)^{-q} \succ 0$ to make the gradient zero:

$$\nabla V_{X_k}\big(\tilde{X}_{k+1}\big) + \alpha F_k = \nabla w(\tilde{X}_{k+1}) - \nabla w(X_k) + \alpha F_k = -\tilde{X}_{k+1}^{-1/q} + X_k^{-1/q} + \alpha F_k = 0 \ . \tag{8.23}$$

This uses our assumption $\alpha X_k^{1/2q} F_k X_k^{1/2q} \succeq -\frac{1}{2q} I$ which certainly implies $\alpha F_{k,i} \succeq -\frac{1}{2} X_k^{-1/q}$, so as to ensure that $\tilde{X}_{k+1}$ is well defined.

Next, it is easy to verify that $X_{k+1} = \arg\min_{Z\in\Delta}\{V_{\tilde{X}_{k+1}}(Z)\}$. In other words, one can describe the update $X_k \to X_{k+1}$ by adding an intermediate stage $X_k \to \tilde{X}_{k+1} \to X_{k+1}$. We assume for notational simplicity that $\tilde{X}_0 \stackrel{\text{def}}{=} X_0$.

Using (8.23), we easily obtain that

$$\begin{aligned}\langle \alpha F_k, X_k - U\rangle = \langle \nabla w(X_k) - \nabla w(\tilde{X}_{k+1}), X_k - U\rangle &= V_{X_k}(U) - V_{\tilde{X}_{k+1}}(U) + V_{\tilde{X}_{k+1}}(X_k) \\ &\leq V_{\tilde{X}_k}(U) - V_{\tilde{X}_{k+1}}(U) + V_{\tilde{X}_{k+1}}(X_k) \ .\end{aligned} \tag{8.24}$$

Above, the second equality is due to the three-point equality and the only inequality is due to the generalized Pythagorean theorem of Bregman divergence (see Lemma 8.3).

We now upper bound $V_{\tilde{X}_{k+1}}(X_k)$ in two cases: the case when $\alpha F_k = -PP^T \preceq 0$ and the case when $\alpha F_k = PP^T \succeq 0$. In both cases, we denote by $\beta \stackrel{\text{def}}{=} \alpha\|X_k^{1/2q} F_k X_k^{1/2q}\|_{\mathsf{spe}} = \|X_k^{1/2q} PP^T X_k^{1/2q}\|_{\mathsf{spe}}$. Notice that this implies [9]

$$X_k^{1/2q} PP^T X_k^{1/2q} \preceq \beta I \quad \text{and} \quad P^T X_k^{1/q} P \preceq \beta I \ . \tag{8.25}$$

- If $\alpha F_k = -PP^T$, we have $X_k^{-1/q} \succ PP^T$ and $\beta \leq \frac{1}{2q}$ by our assumption, so using the Sherman-Morrison-Woodbury formula,

$$\begin{aligned}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} &= \mathrm{Tr}\big((X_k^{-1/q} - PP^T)^{-1}\big)^{q-1} \\ &= \mathrm{Tr}\Big(X_k^{1/q} + X_k^{1/q} P\big(I - P^T X_k^{1/q} P\big)^{-1} P^T X_k^{1/q}\Big)^{q-1} \\ &\leq \mathrm{Tr}\Big(X_k^{1/q} + \frac{X_k^{1/q} PP^T X_k^{1/q}}{1-\beta}\Big)^{q-1} \ ,\end{aligned}$$

  where the last inequality follows because $(I - P^T X_k^{1/q} P)^{-1} \preceq \frac{1}{1-\beta} I$ owing to (8.25), as well as $A \preceq B \implies \mathrm{Tr}A^n \leq \mathrm{Tr}B^n$. We continue and write

$$\begin{aligned}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} &\leq \mathrm{Tr}\Big(X_k^{1/q} + \frac{X_k^{1/q} PP^T X_k^{1/q}}{1-\beta}\Big)^{q-1} \\ &= \mathrm{Tr}\Big(X_k^{1/2q}\big(I + \frac{X_k^{1/2q} PP^T X_k^{1/2q}}{1-\beta}\big) X_k^{1/2q}\Big)^{q-1}\end{aligned}$$

[9] The second inequality is because $P^T X_k^{1/q} P = (P^T X_k^{1/2q})(P^T X_k^{1/2q})^T$ and has the same largest eigenvalue as $(P^T X_k^{1/2q})^T (P^T X_k^{1/2q}) = X_k^{1/2q} PP^T X_k^{1/2q}$.

$$\leq \mathrm{Tr}\Big(X_k^{(q-1)/2q}\big(I+\frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\beta}\big)^{q-1}X_k^{(q-1)/2q}\Big)$$
$$= \mathrm{Tr}\Big(X_k^{1-1/q}\big(I+\frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\beta}\big)^{q-1}\Big) \ ,$$

where the inequality uses the Lieb-Thirring trace inequality (which relies on the fact that $q-1\geq 1$). Finally, denoting by $D \stackrel{\rm def}{=} \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\beta} \preceq \frac{\beta}{1-\beta}I$ (which uses (8.25) again), we have

$$\big(I+D\big)^{q-1} \preceq I+(q-1)D+O(q^2\beta)\cdot D \ .$$

This matrix inequality can be proved by first turning into its eigenbasis, and then verifying that $(1+x)^{q-1}\leq 1+(q-1)x+O(q^2\beta)x$ for all $x\in[0,\frac{\beta}{1-\beta}]$ (which uses the fact that $\beta\leq 1/2q$). Using this inequality, we conclude that

$$\begin{aligned}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} &\leq \mathrm{Tr}\Big(X_k^{1-1/q}\big(I+\frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\beta}\big)^{q-1}\Big)\\ &\leq \mathrm{Tr}\Big(X_k^{1-1/q}\Big(I+\Big((q-1)+O(q^2\beta)\Big)\frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\beta}\Big)\Big)\\ &= \mathrm{Tr}X_k^{1-1/q}+(q-1)(1+O(q\beta))X_k\bullet PP^T \ .\end{aligned}$$

Therefore,

$$\begin{aligned}V_{\tilde{X}_{k+1}}(X_k) &= \tilde{X}_{k+1}^{-1/q}\bullet X_k+\frac{1}{q-1}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q}-\frac{q}{q-1}\mathrm{Tr}X_k^{1-1/q}\\ &= (X_k^{-1/q}-PP^T)\bullet X_k+\frac{1}{q-1}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q}-\frac{q}{q-1}\mathrm{Tr}X_k^{1-1/q}\\ &= -PP^T\bullet X_k+\frac{1}{q-1}\big(\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q}-\mathrm{Tr}X_k^{1-1/q}\big)\\ &= O(q\beta)\cdot PP^T\bullet X_k = O(q\alpha^2)(X_k\bullet|F_k|)\cdot\|X_k^{1/2q}F_kX_k^{1/2q}\|_{\mathsf{spe}} \ .\end{aligned}$$

- If $\alpha F_k = PP^T$, using the Sherman-Morrison-Woodbury formula,

$$\begin{aligned}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} &= \mathrm{Tr}\big((X_k^{-1/q}+PP^T)^{-1}\big)^{q-1}\\ &= \mathrm{Tr}\Big(X_k^{1/q}-X_k^{1/q}P(I+P^TX_k^{1/q}P)^{-1}P^TX_k^{1/q}\Big)^{q-1}\\ &\leq \mathrm{Tr}\Big(X_k^{1/q}-\frac{X_k^{1/q}PP^TX_k^{1/q}}{1+\beta}\Big)^{q-1} \ ,\end{aligned}$$

where the last inequality follows because $(I+P^TX_k^{1/q}P)^{-1}\succeq\frac{1}{1+\beta}I$ owing to (8.25), as well as $A\preceq B \implies \mathrm{Tr}A^n\leq\mathrm{Tr}B^n$. We continue and write

$$\begin{aligned}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} &\leq \mathrm{Tr}\Big(X_k^{1/q}-\frac{X_k^{1/q}PP^TX_k^{1/q}}{1+\beta}\Big)^{q-1}\\ &= \mathrm{Tr}\Big(X_k^{1/2q}\big(I-\frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1+\beta}\big)X_k^{1/2q}\Big)^{q-1}\end{aligned}$$

$$\leq \mathrm{Tr}\Big(X_k^{(q-1)/2q}\big(I - \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1+\beta}\big)^{q-1}X_k^{(q-1)/2q}\Big)$$
$$= \mathrm{Tr}\Big(X_k^{1-1/q}\big(I - \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1+\beta}\big)^{q-1}\Big) \ ,$$

where the inequality again uses the Lieb-Thirring trace inequality. Denoting by $D \stackrel{\text{def}}{=} \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1+\beta} \preceq \frac{\beta}{1+\beta}I$ (which uses (8.25) again), we see that

$$\big(I-D\big)^{q-1} \preceq I - (q-1)D + O(q^2\beta)\cdot D \ .$$

This matrix inequality can be proved by first turning into its eigenbasis, and then verifying that $(1-x)^{q-1} \leq 1-(q-1)x + O(q^2\beta)x$ for all $x \in [0, \frac{\beta}{1+\beta}]$ (which uses the fact that $\beta \leq 1/2q$). This concludes that

$$\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} \leq \mathrm{Tr}\Big(X_k^{1-1/q}\big(I - \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1+\beta}\big)^{q-1}\Big)$$
$$\leq \mathrm{Tr}\Big(X_k^{1-1/q}\Big(I - (q-1)\big(1-O(q\beta)\big)\frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1+\beta}\Big)\Big)$$
$$= \mathrm{Tr}X_k^{1-1/q} - (q-1)\big(1-O(q\beta)\big)X_k \bullet PP^T \ .$$

Therefore,

$$V_{\tilde{X}_{k+1}}(X_k) = \tilde{X}_{k+1}^{-1/q} \bullet X_k + \frac{1}{q-1}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}X_k^{1-1/q}$$
$$= (X_k^{-1/q} + PP^T) \bullet X_k + \frac{1}{q-1}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}X_k^{1-1/q}$$
$$= PP^T \bullet X_k + \frac{1}{q-1}\big(\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} - \mathrm{Tr}X_k^{1-1/q}\big)$$
$$= O(q\beta)\cdot PP^T \bullet X_k = O(q\alpha^2)(X_k \bullet |F_k|)\cdot \|X_k^{1/2q}F_kX_k^{1/2q}\|_{\mathsf{spe}} \ .$$

Finally, substituting the above upper bound on $V_{\tilde{X}_{k+1}}(X_k)$ into (8.24) and telescoping it for $k = 0, \ldots, T-1$, we obtain

$$\sum_{k=0}^{T-1}\langle F_k, X_k - U\rangle \leq \frac{V_{\tilde{X}_0}(U) - V_{\tilde{X}_T}(U)}{\alpha} + O(q\alpha)\sum_{k=0}^{T-1}(X_k \bullet |F_k|)\cdot\|X_k^{1/2q}F_kX_k^{1/2q}\|_{\mathsf{spe}} \ .$$

The desired result of this theorem now follows from the above inequality and the simple upper bound $V_{\tilde{X}_0}(U) = V_{X_0}(U) \leq \frac{q}{q-1}n^{1/q}$ and the nonnegativity $V_{\tilde{X}_T}(U) \geq 0$. □

8.F Robust Linear-Sized Sparsification

In this section, we deduce the more generalized version of the same result presented in Section 8.5, with the following major differences.

- **Regularizer.** In this section, we allow the general $\ell_{1-1/q}$ regularizer to be used, for any even integer $q \geq 2$, rather than just the $\ell_{1/2}$ regularizer. (The assumption

on q being even integer rather than all reals no less than 2 is only for the sake of proof convenience.)

- **High rank.** In this section, we allow $\hat{L}_e$ to be possibly of high rank, rather than just rank 1.

- **Approximate computations.** In this section, we allow many computations to be approximate rather than exact. This will enable the algorithm to be more efficiently implemented in the next section (Appendix 8.G). In particular, we allow the following quantities to be approximately computed.

  - We only need $\mathrm{Tr}\hat{L}_e$ to be in $[1-\varepsilon_1, 1]$ rather than exactly one.
  - We only need $\mathrm{Tr}X_k$ and $\mathrm{Tr}Y_k$ to be in $[1, 1+\varepsilon_1]$ rather than exactly one.
  - We only need $\hat{L}_e \bullet X_k$ and $\hat{L}_e \bullet Y_k$ to be computed only up to a $(1+\varepsilon_1)$ multiplicative error.

  We will assume throughout this paper that $\varepsilon_1 < 1/2$.

8.F.1 The Problem

Suppose we are given a decomposition of the identity matrix $I = \sum_{e=1}^m w_e \hat{L}_e$, where each $\hat{L}_e$ satisfies ① $0 \preceq \hat{L}_e \preceq I$, ② $\mathrm{Tr}\hat{L}_e \in [1-\varepsilon_1, 1]$, and ③ $\hat{L}_e$ may be of high rank. The weights $w_e > 0$ may be unknown.

In this section, we are interested in using the $\ell_{1-1/q}$ regularizer for $\texttt{MirrorDescent}_{\ell_{1-1/q}}$ in order to find scalars $s_e \geq 0$ satisfying

$$I \preceq \sum_{e=1}^m s_e \cdot \hat{L}_e \preceq \left(1 + \sqrt{\frac{8q^2}{q-1}} \cdot \varepsilon + O(\varepsilon_1 + q\varepsilon^2 + \varepsilon_1\varepsilon\sqrt{q})\right) I \ , \tag{8.26}$$

while the sparsity of s —that is, $|\{e \in [m] : s_e > 0\}|$— is at most n/ε^2. We will not worry about the running time in this section, and defer all the implementation details to Appendix 8.G.

Throughout this section, we pick $w(X)$ to be the $\ell_{1-1/q}$ regularizer and $V_X(Y)$ to be its induced Bregman divergence.

8.F.2 Our Algorithm

Maintain two sequences $X_k, Y_k \succeq 0$ satisfying $\mathrm{Tr}X_k, \mathrm{Tr}Y_k \in [1, 1+\varepsilon_1]$. At the very beginning we choose $X_0 = \frac{1}{n}I$ and $Y_0 = \frac{1}{n}I$ as before.

At each iteration $k = 0, 1, \ldots, T-1$, find an arbitrary e_k such that

$$\mathsf{Dot}(\hat{L}_{e_k}, X_k) \leq (1+\varepsilon_1)^2 \mathsf{Dot}(\hat{L}_{e_k}, Y_k) \ ,$$

where $\mathsf{Dot}(\hat{L}_e, X)$ is some algorithm[10] that approximately computes $\hat{L}_e \bullet X$ and sat-

[10]The implementation of this algorithm will be described in Appendix 8.G.

isfies

$$\hat{L}_e \bullet X \le \mathsf{Dot}(\hat{L}_e, X) \le (1+\varepsilon_1) \cdot \hat{L}_e \bullet X \ .$$

We can always do so because after averaging,

$$\sum_e w_e \mathsf{Dot}(\hat{L}_e, X_k) \le (1+\varepsilon_1) \sum_e (w_e \hat{L}_e) \bullet X_k = (1+\varepsilon_1) \mathrm{Tr} X_k$$
$$\le (1+\varepsilon_1)^2 \mathrm{Tr} Y_k = (1+\varepsilon_1)^2 \sum_e (w_e \hat{L}_e) \bullet Y_k \le (1+\varepsilon_1)^2 \sum_e w_e \mathsf{Dot}(\hat{L}_e, Y_k) \ .$$

At each iteration $k = 0, 1, \ldots, T-1$, we perform updates by finding[11] arbitrary $\delta_X, \delta_Y \ge 0$ satisfying

$$Y_k^{-1/q} + \frac{\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, Y_k)^{1/q}} - \delta_Y I \succeq 0 \quad \text{and} \quad \mathrm{Tr} X_{k+1}, \mathrm{Tr} Y_{k+1} \in [1, 1+\varepsilon_1] \ ,$$

where

$$X_{k+1} \stackrel{\text{def}}{=} \left(X_k^{-1/q} + \frac{-\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}} + \delta_X I \right)^{-q}$$

and

$$Y_{k+1} \stackrel{\text{def}}{=} \left(Y_k^{-1/q} + \frac{\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, Y_k)^{1/q}} - \delta_Y I \right)^{-q} \ .$$

Above, $\alpha > 0$ is some parameter that will be specified at the end of this section. Note that this corresponds to performing updates

$$\text{“} \quad X_{k+1} \leftarrow \operatorname*{arg\,min}_{Z \in \Delta_{n \times n}} \left\{ V_{X_k}(Z) + \left\langle \frac{-\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}}, Z \right\rangle \right\} \quad \text{”} \quad \text{and}$$

$$\text{“} \quad Y_{k+1} \leftarrow \operatorname*{arg\,min}_{Z \in \Delta_{n \times n}} \left\{ V_{Y_k}(Z) + \left\langle \frac{\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, Y_k)^{1/q}}, Z \right\rangle \right\} \quad \text{”}$$

however, we have **not** required $\mathrm{Tr} X_{k+1} = \mathrm{Tr} Y_{k+1}$ to be precisely equal to 1.

For analysis purpose only, we also define $\tilde{X}_{k+1}$ and $\tilde{Y}_{k+1}$ to be similar updates but without δ_X or δ_Y:

$$\tilde{X}_{k+1} \stackrel{\text{def}}{=} \left(X_k^{-1/q} + \frac{-\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}} \right)^{-q} \quad \text{and} \quad \tilde{Y}_{k+1} \stackrel{\text{def}}{=} \left(Y_k^{-1/q} + \frac{\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, Y_k)^{1/q}} \right)^{-q} \ .$$

We assume also $\tilde{X}_0 \stackrel{\text{def}}{=} X_0$.

Note that $\tilde{Y}_{k+1}$ is always well defined. Claim 8.10 below shows that as long as $\alpha < 1$, it always satisfies $X_k^{-1/q} + \frac{-\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}} \succeq 0$, so $\tilde{X}_{k+1}$ is also well defined.

Claim 8.10. *For every $e \in [m]$, we have $X_k^{-1/q} \succeq \frac{\hat{L}_e}{(\hat{L}_e \bullet X_k)^{1/q}} \succeq \frac{\hat{L}_e}{\mathsf{Dot}(\hat{L}_e, X_k)^{1/q}}$. In addition, denoting by $\frac{\alpha \hat{L}_e}{\mathsf{Dot}(\hat{L}_e, X_k)^{1/q}} = PP^T$, we have $0 \preceq P^T X_k^{1/q} P \preceq \alpha I$.*

Similarly, for every $e \in [m]$, we have $Y_k^{-1/q} \succeq \frac{\hat{L}_e}{(\hat{L}_e \bullet Y_k)^{1/q}} \succeq \frac{\hat{L}_e}{\mathsf{Dot}(\hat{L}_e, Y_k)^{1/q}}$. In addition,

[11] The existence of such δ_X and δ_Y shall become soon (due to Claim 8.10). The implementation of these updates will be described in Appendix 8.G.

denoting by $\frac{\alpha \hat{L}_e}{\mathsf{Dot}(\hat{L}_e, Y_k)^{1/q}} = PP^T$, *we have* $0 \preceq P^T Y_k^{1/q} P \preceq \alpha I$.

Proof. We only prove the X_k part because the Y_k part is similar. We first compute

$$\|X_k^{1/2q} \hat{L}_e X_k^{1/2q}\|_{\mathsf{spe}}^q \le \mathrm{Tr}((X_k^{1/2q} \hat{L}_e X_k^{1/2q})^q) \le \mathrm{Tr}(X_k^{1/2} (\hat{L}_e)^q X_k^{1/2}) \ ,$$

where the last inequality follows from the Lieb-Thirring trace inequality.

Next, using the fact that $\hat{L}_e \preceq I$, we obtain that $(\hat{L}_e)^q \preceq \hat{L}_e$. Therefore,

$$\|X_k^{1/2q} \hat{L}_e X_k^{1/2q}\|_{\mathsf{spe}}^q \le \mathrm{Tr}(X_k^{1/2} \hat{L}_e X_k^{1/2}) = \hat{L}_e \bullet X_k \ .$$

In other words, we have $X_k^{1/2q} \hat{L}_e X_k^{1/2q} \preceq (\hat{L}_e \bullet X_k)^{1/q} \cdot I$ which means $X_k^{-1/q} \succeq \frac{\hat{L}_e}{(\hat{L}_e \bullet X_k)^{1/q}}$. We automatically have $\frac{\hat{L}_e}{(\hat{L}_e \bullet X_k)^{1/q}} \succeq \frac{\hat{L}_e}{\mathsf{Dot}(\hat{L}_e, X_k)^{1/q}}$ because $\mathsf{Dot}(\hat{L}_e, X_k) \ge \hat{L}_e \bullet X_k$.

To prove the second half, beginning from $X_k^{-1/q} \succeq \frac{1}{\alpha} \cdot PP^T$, we left multiply it with $P^T X_k^{1/q}$ and right multiply it with $X_k^{1/q} P$, and obtain $P^T X_k^{1/q} P \succeq \frac{1}{\alpha} \cdot P^T X_k^{1/q} P P^T X_k^{1/q} P$. Denoting by $D \stackrel{\text{def}}{=} P^T X_k^{1/q} P$, we have $D \succeq \frac{1}{\alpha} D^2$, which immediately implies $0 \preceq D \preceq \alpha I$ as desired. □

We have now finished the description of the algorithm. We remark here that $\mathrm{Tr}\tilde{X}_{k+1} < \mathrm{Tr} X_k$ and $\mathrm{Tr}\tilde{Y}_{k+1} > \mathrm{Tr} Y_k$. Therefore, since $\mathrm{Tr} X_{k+1}$ increases as δ_X increases, while $\mathrm{Tr} Y_{k+1}$ decreases as δ_Y increase, we conclude the existence of $\delta_X, \delta_Y \ge 0$ so that $\mathrm{Tr} X_{k+1}, \mathrm{Tr} Y_{k+1} \in [1, 1+\varepsilon_1]$.

8.F.3 Our Analysis

We begin by reproving essentially the first half of Theorem 8.5: that is, to prove (8.22). We need to pay extra attention here since our $\mathrm{Tr} X_k$ and $\mathrm{Tr} Y_k$ do not precisely equal to 1.

Lemma 8.11. *For every* $U_X \succeq 0$ *satisfying* $\mathrm{Tr} U_X \le 1$, *and every* $U_Y \succeq 0$ *satisfying* $\mathrm{Tr} U_Y \ge 1 + \varepsilon_1$,

$$\Big\langle \frac{-\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}}, X_k - U_X \Big\rangle \le V_{\tilde{X}_{k+1}}(X_k) + V_{\tilde{X}_k}(U_X) - V_{\tilde{X}_{k+1}}(U_X) \ , \quad \textit{and}$$

$$\Big\langle \frac{\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, Y_k)^{1/q}}, Y_k - U_Y \Big\rangle \le V_{\tilde{Y}_{k+1}}(Y_k) + V_{\tilde{Y}_k}(U_Y) - V_{\tilde{Y}_{k+1}}(U_Y) \ .$$

Proof. We first prove the X_k part. By our choice of the regularizer, we have

$$0 = \nabla w(\tilde{X}_{k+1}) - \nabla w(X_k) + \frac{-\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}} = -\tilde{X}_{k+1}^{-1/q} + X_k^{-1/q} + \frac{-\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}} \ .$$

Next, we obtain that

$$\begin{aligned} \Big\langle \frac{-\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}}, X_k - U_X \Big\rangle &= \langle \nabla w(X_k) - \nabla w(\tilde{X}_{k+1}), X_k - U_X \rangle \\ &\stackrel{①}{=} V_{X_k}(U_X) - V_{\tilde{X}_{k+1}}(U_X) + V_{\tilde{X}_{k+1}}(X_k) \end{aligned}$$

$$\overset{②}{\leq} V_{\tilde{X}_k}(U_X) - V_{\tilde{X}_{k+1}}(U_X) + V_{\tilde{X}_{k+1}}(X_k) \ .$$

Above, ① is due to the three-point equality of Bregman divergence, and ② comes from

$$\begin{aligned} V_{X_k}(U_X) - V_{\tilde{X}_k}(U_X) &\overset{③}{=} \big(X_k^{-1/q} - \tilde{X}_k^{-1/q}\big) \bullet U_X + \frac{1}{q-1}\big(\mathrm{Tr}X_k^{1-1/q} - \mathrm{Tr}\tilde{X}_k^{1-1/q}\big) \\ &\overset{④}{=} \delta_X \mathrm{Tr}U_X + \frac{1}{q-1}\sum_i \frac{1}{\lambda_i^{q-1}} - \frac{1}{(\lambda_i - \delta_X)^{q-1}} \\ &\overset{⑤}{\leq} \delta_X \mathrm{Tr}U_X - \delta_X \sum_i \frac{1}{\lambda_i^q} \overset{⑥}{\leq} 0 \ . \end{aligned}$$

Here, ③ is owing to the definition of Bregman divergence. ④ comes from the fact that $\tilde{X}_{k+1}^{-1/q} = X_{k+1}^{-1/q} - \delta_X I$, and the definition of choosing λ_i to be the i-th eigenvalue of $X_{k+1}^{-1/q}$. ⑤ follows from the convexity of $f(x) = x^{1-q}$ which implies $f(\lambda_i) - f(\lambda_i - \delta_X) \leq \nabla f(\lambda_i) \cdot \delta_X$. ⑥ is by our assumption of $\mathrm{Tr}U_X \leq 1$ as well as $\mathrm{Tr}X_{k+1} = \sum_i \frac{1}{\lambda_i^q} \geq 1$.

Similarly, for the Y_k part, we can compute

$$\begin{aligned} \langle \frac{\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, Y_k)^{1/q}}, Y_k - U_Y \rangle &= \langle \nabla w(Y_k) - \nabla w(\tilde{Y}_{k+1}), Y_k - U_Y \rangle \\ &\overset{①}{=} V_{Y_k}(U_Y) - V_{\tilde{Y}_{k+1}}(U_Y) + V_{\tilde{Y}_{k+1}}(Y_k) \\ &\overset{②}{\leq} V_{\tilde{Y}_k}(U_Y) - V_{\tilde{Y}_{k+1}}(U_Y) + V_{\tilde{Y}_{k+1}}(Y_k) \ . \end{aligned}$$

Above, ① is due to the three-point equality, and inequality ② comes from

$$\begin{aligned} V_{Y_k}(U_Y) - V_{\tilde{Y}_k}(U_Y) &\overset{③}{=} \big(Y_k^{-1/q} - \tilde{Y}_k^{-1/q}\big) \bullet U_Y + \frac{1}{q-1}\big(\mathrm{Tr}Y_k^{1-1/q} - \mathrm{Tr}\tilde{Y}_k^{1-1/q}\big) \\ &\overset{④}{=} -\delta_Y \mathrm{Tr}U_Y + \frac{1}{q-1}\sum_i \frac{1}{\lambda_i^{q-1}} - \frac{1}{(\lambda_i + \delta_Y)^{q-1}} \\ &\overset{⑤}{\leq} -\delta_Y \mathrm{Tr}U_Y + \delta_Y \sum_i \frac{1}{\lambda_i^q} \overset{⑥}{\leq} 0 \ . \end{aligned}$$

Here, ③ is owing to the definition of Bregman divergence. ④ comes from the fact that $\tilde{Y}_{k+1}^{-1/q} = Y_{k+1}^{-1/q} + \delta_Y I$, and the definition of choosing λ_i to be the i-th eigenvalue of $Y_{k+1}^{-1/q}$. ⑤ follows from the convexity of $f(x) = x^{1-q}$ which implies $f(\lambda_i) - f(\lambda_i + \delta_Y) \leq \nabla f(\lambda_i) \cdot (-\delta_Y)$. ⑥ is by our assumption of $\mathrm{Tr}U_Y \geq 1 + \varepsilon_1$ as well as $\mathrm{Tr}Y_{k+1} = \sum_i \frac{1}{\lambda_i^q} \leq 1 + \varepsilon_1$. □

In a next step, we reprove essentially the second half of Theorem 8.5: that is, to provide upper bounds on $V_{\tilde{X}_{k+1}}(X_k)$ and $V_{\tilde{Y}_{k+1}}(Y_k)$ in Lemma 8.12 and Lemma 8.13.

Lemma 8.12. *As long as $q \geq 2$ and $\alpha \leq 1/2q$, we have*

$$V_{\tilde{X}_{k+1}}(X_k) \leq \frac{q}{2}(\alpha^2 + O(q\alpha^3)) \cdot \big(\hat{L}_{e_k} \bullet X_k\big)^{1-1/q} \ .$$

Proof. Suppose $\frac{\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}} = PP^T$. Then, using the Sherman-Morrison-Woodbury

formula,

$$\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} = \mathrm{Tr}\big((X_k^{-1/q} - PP^T)^{-1}\big)^{q-1} = \mathrm{Tr}\Big(X_k^{1/q} + X_k^{1/q}P(I - P^TX_k^{1/q}P)^{-1}P^TX_k^{1/q}\Big)^{q-1}$$
$$\leq \mathrm{Tr}\Big(X_k^{1/q} + \frac{X_k^{1/q}PP^TX_k^{1/q}}{1-\alpha}\Big)^{q-1} \quad ,$$

where the last inequality follows because $(I - P^TX_k^{1/q}P)^{-1} \preceq \frac{1}{1-\alpha}I$ owing to Claim 8.10, as well as $A \preceq B \implies \mathrm{Tr}A^n \leq \mathrm{Tr}B^n$. We continue and write

$$\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} \leq \mathrm{Tr}\Big(X_k^{1/q} + \frac{X_k^{1/q}PP^TX_k^{1/q}}{1-\alpha}\Big)^{q-1} = \mathrm{Tr}\Big(X_k^{1/2q}\big(I + \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\alpha}\big)X_k^{1/2q}\Big)^{q-1}$$
$$\leq \mathrm{Tr}\Big(X_k^{(q-1)/2q}\big(I + \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\alpha}\big)^{q-1}X_k^{(q-1)/2q}\Big)$$
$$= \mathrm{Tr}\Big(X_k^{1-1/q}\big(I + \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\alpha}\big)^{q-1}\Big) \quad ,$$

where the inequality uses the Lieb-Thirring trace inequality (which relies on the fact that $q-1 \geq 1$). Finally, denoting by $D \stackrel{\mathrm{def}}{=} \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\alpha} \preceq \frac{\alpha}{1-\alpha}I$, we see that

$$\big(I + D\big)^{q-1} \preceq I + (q-1)D + \Big(\frac{(q-1)(q-2)}{2}\alpha + O(q^3\alpha^2)\Big)D \quad .$$

This above matrix inequality can be proved by first turning into its eigenbasis, and then verifying that $(1+x)^{q-1} \leq 1+(q-1)x+\frac{(q-1)(q-2)}{2}\alpha x+O(q^3\alpha^2)x$ for all $x \in [0, \frac{\alpha}{1-\alpha}]$. (This uses the fact that $\alpha \leq 1/2q$). Next, using the above matrix inequality, we conclude that

$$\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} \leq \mathrm{Tr}\Big(X_k^{1-1/q}\big(I + \frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\alpha}\big)^{q-1}\Big)$$
$$\leq \mathrm{Tr}\Big(X_k^{1-1/q}\Big(I + \Big((q-1) + \frac{(q-1)(q-2)}{2}\alpha + O(q^3\alpha^2)\Big)\frac{X_k^{1/2q}PP^TX_k^{1/2q}}{1-\alpha}\Big)\Big)$$
$$= \mathrm{Tr}X_k^{1-1/q} + (q-1)\frac{1 + \frac{q-2}{2}\alpha + O(q^2\alpha^2)}{1-\alpha}X_k \bullet PP^T \quad .$$

Therefore,

$$V_{\tilde{X}_{k+1}}(X_k) = \tilde{X}_{k+1}^{-1/q} \bullet X_k + \frac{1}{q-1}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}X_k^{1-1/q}$$
$$= (X_k^{-1/q} - PP^T) \bullet X_k + \frac{1}{q-1}\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}X_k^{1-1/q}$$
$$= -PP^T \bullet X_k + \frac{1}{q-1}\big(\mathrm{Tr}\tilde{X}_{k+1}^{1-1/q} - \mathrm{Tr}X_k^{1-1/q}\big)$$
$$\leq PP^T \bullet X_k\Big(-1 + \frac{1 + \frac{q-2}{2}\alpha + O(q^2\alpha^2)}{1-\alpha}\Big)$$
$$= \frac{q}{2}(\alpha + O(q\alpha^2)) \cdot PP^T \bullet X_k$$
$$\leq \frac{q}{2}(\alpha^2 + O(q\alpha^3)) \cdot \big(\hat{L}_{e_k} \bullet X_k\big)^{1-1/q} \quad . \qquad \square$$

Lemma 8.13. *As long as $q \geq 2$ and $\alpha \leq 1/2q$, we have*

$$V_{\tilde{Y}_{k+1}}(Y_k) \leq \frac{q}{2}\big(\alpha^2 + O(\alpha^3)\big) \cdot \big(\hat{L}_{e_k} \bullet Y_k\big)^{1-1/q} \ .$$

Proof. Suppose $\frac{\alpha \hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, Y_k)^{1/q}} = PP^T$. Then, using the Sherman-Morrison-Woodbury formula,

$$\begin{aligned}
\mathrm{Tr}\tilde{Y}_{k+1}^{1-1/q} = \mathrm{Tr}\big((Y_k^{-1/q} + PP^T)^{-1}\big)^{q-1} &= \mathrm{Tr}\Big(Y_k^{1/q} - Y_k^{1/q}P(I + P^TY_k^{1/q}P)^{-1}P^TY_k^{1/q}\Big)^{q-1} \\
&\leq \mathrm{Tr}\Big(Y_k^{1/q} - \frac{Y_k^{1/q}PP^TY_k^{1/q}}{1+\alpha}\Big)^{q-1} \ ,
\end{aligned}$$

where the last inequality follows because $(I + P^TY_k^{1/q}P)^{-1} \succeq \frac{1}{1+\alpha}I$ owing to Claim 8.10, as well as $A \preceq B \implies \mathrm{Tr}A^n \leq \mathrm{Tr}B^n$. We continue and write

$$\begin{aligned}
\mathrm{Tr}\tilde{Y}_{k+1}^{1-1/q} &\leq \mathrm{Tr}\Big(Y_k^{1/q} - \frac{Y_k^{1/q}PP^TY_k^{1/q}}{1+\alpha}\Big)^{q-1} = \mathrm{Tr}\Big(Y_k^{1/2q}\big(I - \frac{Y_k^{1/2q}PP^TY_k^{1/2q}}{1+\alpha}\big)Y_k^{1/2q}\Big)^{q-1} \\
&\leq \mathrm{Tr}\Big(Y_k^{(q-1)/2q}\big(I - \frac{Y_k^{1/2q}PP^TY_k^{1/2q}}{1+\alpha}\big)^{q-1}Y_k^{(q-1)/2q}\Big) \\
&= \mathrm{Tr}\Big(Y_k^{1-1/q}\big(I - \frac{Y_k^{1/2q}PP^TY_k^{1/2q}}{1+\alpha}\big)^{q-1}\Big) \ ,
\end{aligned}$$

where the inequality again uses the Lieb-Thirring trace inequality (which relies on the fact that $q - 1 \geq 1$). Denoting by $D \stackrel{\mathrm{def}}{=} \frac{Y_k^{1/2q}PP^TY_k^{1/2q}}{1+\alpha} \preceq \frac{\alpha}{1+\alpha}I$, we see that

$$\big(I - D\big)^{q-1} \preceq I - (q-1)D + \frac{(q-1)(q-2)\alpha}{2(1+\alpha)}D \ .$$

This above matrix inequality can be proved by first turning into its eigenbasis, and then verifying that $(1-x)^{q-1} \leq 1-(q-1)x + \frac{(q-1)(q-2)}{2}\frac{\alpha}{1+\alpha}x$ for all $x \in [0, \frac{\alpha}{1+\alpha}]$. (This uses the fact that $\alpha \leq 1/2q$). Next, using the above matrix inequality, we conclude that

$$\begin{aligned}
\mathrm{Tr}\tilde{Y}_{k+1}^{1-1/q} &\leq \mathrm{Tr}\Big(Y_k^{1-1/q}\big(I - \frac{Y_k^{1/2q}PP^TY_k^{1/2q}}{1+\alpha}\big)^{q-1}\Big) \\
&\leq \mathrm{Tr}\Big(Y_k^{1-1/q}\Big(I - (q-1)\big(1 - \frac{(q-2)\alpha}{2(1+\alpha)}\big)\frac{Y_k^{1/2q}PP^TY_k^{1/2q}}{1+\alpha}\Big)\Big) \\
&= \mathrm{Tr}Y_k^{1-1/q} - (q-1)\big(1 - \frac{(q-2)\alpha}{2(1+\alpha)}\big)\frac{1}{1+\alpha}Y_k \bullet PP^T \ .
\end{aligned}$$

Therefore,

$$\begin{aligned}
V_{\tilde{Y}_{k+1}}(Y_k) &= \tilde{Y}_{k+1}^{-1/q} \bullet Y_k + \frac{1}{q-1}\mathrm{Tr}\tilde{Y}_{k+1}^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}Y_k^{1-1/q} \\
&= (Y_k^{-1/q} + PP^T) \bullet Y_k + \frac{1}{q-1}\mathrm{Tr}\tilde{Y}_{k+1}^{1-1/q} - \frac{q}{q-1}\mathrm{Tr}Y_k^{1-1/q} \\
&= PP^T \bullet Y_k + \frac{1}{q-1}\big(\mathrm{Tr}\tilde{Y}_{k+1}^{1-1/q} - \mathrm{Tr}Y_k^{1-1/q}\big)
\end{aligned}$$

$$
\begin{aligned}
&\le PP^T \bullet Y_k \Big(1 - \frac{\big(1 - \frac{(q-2)\alpha}{2(1+\alpha)}\big)}{1+\alpha}\Big) \\
&= \frac{q}{2}\big(\alpha + O(\alpha^2)\big) \cdot PP^T \bullet Y_k \\
&\le \frac{q}{2}\big(\alpha^2 + O(\alpha^3)\big) \cdot \big(\hat{L}_{e_k} \bullet Y_k\big)^{1-1/q} \ .
\end{aligned}
$$

□

Theorem 8.14. *Suppose* $\varepsilon < \frac{1}{4\sqrt{q}}$ *and* $\varepsilon_1 < \frac{1}{2}$*, and we choose* $\alpha = \frac{\varepsilon\sqrt{2}}{\sqrt{q-1}}$ *and* $T = \frac{n}{\varepsilon^2}$*. Then, the matrix* $M_Y \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \frac{\hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, Y_k)^{1/q}}$ *satisfies that*

$$
\lambda_{\mathsf{max}}(M_Y) - \lambda_{\mathsf{min}}(M_Y) \le \lambda_{\mathsf{min}}(M_Y) \cdot \Big(\sqrt{\frac{8q^2}{q-1}} \cdot \varepsilon + O(\varepsilon_1 + q\varepsilon^2)\Big) \ .
$$

This theorem provides the sparsification guarantee to our Theorem 8.1 and 8.2. We shall provide its running time guarantee in the next section.

Proof. Define matrices $M_X \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \frac{\hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}}$ and $M_Y \stackrel{\text{def}}{=} \sum_{k=0}^{T-1} \frac{\hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, Y_k)^{1/q}}$. Also, denote by $\xi \stackrel{\text{def}}{=} \frac{q}{2}(\alpha + O(q\alpha^2))$.

We are now ready to rededuce (8.8) and (8.9) in Section 8.5.

Combining Lemma 8.11 and Lemma 8.12, and telescoping for $k = 0, 1, \ldots, T-1$, we have

$$
\forall U_X \succeq 0 \text{ satisfying } \mathrm{Tr}U_X = 1, \quad M_X \bullet U_X \le \frac{V_{\tilde{X}_0}(U_X)}{\alpha} + (1+\xi)\sum_{k=0}^{T-1}\big(\hat{L}_{e_k} \bullet X_k\big)^{1-1/q} \tag{8.27}
$$

$$
\le \frac{qn^{1/q}}{(q-1)\alpha} + (1+\xi)\sum_{k=0}^{T-1}(\hat{L}_{e_k} \bullet X_k)^{1-1/q} \ . \tag{8.28}
$$

Above, the second inequality uses the fact that $V_{\tilde{X}_0}(U_X) \le \frac{q}{q-1}n^{1/q}$.

Combining Lemma 8.11 and Lemma 8.13, and telescoping for $k = 0, 1, \ldots, T-1$, we have

$$
\begin{aligned}
\forall U_Y \succeq 0, \mathrm{Tr}U_Y = 1+\varepsilon_1, \quad M_Y \bullet U_Y &\ge -\frac{V_{\tilde{Y}_0}(U_Y)}{\alpha} + (1-\xi)\sum_{k=0}^{T-1}(\hat{L}_{e_k} \bullet Y_k)^{1-1/q} \\
&\ge -\frac{q(1+\varepsilon_1)n^{1/q}}{(q-1)\alpha} + (1-\xi)\sum_{k=0}^{T-1}\big(\hat{L}_{e_k} \bullet Y_k\big)^{1-1/q} \ .
\end{aligned} \tag{8.29}
$$

Above, the second inequality uses the fact that $V_{\tilde{Y}_0}(U_Y) \le \frac{q(1+\varepsilon_1)}{q-1}n^{1/q}$.

Similar to the proof in Section 8.5, we provide deduce our eigenvalue inequality in two steps.

Lowerbounding $\lambda_{\mathsf{min}}(M_Y)$. Since we have assumed each $\mathrm{Tr}\hat{L}_e$ to be at least $1-\varepsilon_1$, we have

$$\mathrm{Tr}(M_X) = \sum_{k=0}^{T-1} \frac{\mathrm{Tr}\hat{L}_{e_k}}{\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1/q}} \geq \frac{1-\varepsilon_1}{(1+\varepsilon_1)^{1/q}} \sum_{k=0}^{T-1} \frac{1}{(\hat{L}_{e_k} \bullet X_k)^{1/q}} \ .$$

Denoting by $a_k = \hat{L}_{e_k} \bullet X_k$, we can write $\mathrm{Tr}(M_X) \geq \frac{1-\varepsilon_1}{(1+\varepsilon_1)^{1/q}} \sum_{k=0}^{T-1} \frac{1}{a_k^{1/q}}$. Applying (8.27) with the choice of $U_X = \frac{1}{n} I = X_0$, we have

$$\frac{1-\varepsilon_1}{n(1+\varepsilon_1)^{1/q}} \sum_{k=0}^{T-1} \frac{1}{a_k^{1/q}} \leq \frac{1}{n}\mathrm{Tr}M_X = M_X \bullet U_X \leq (1+\xi) \sum_{k=0}^{T-1} (\hat{L}_{e_k} \bullet X_k)^{1-1/q} \leq (1+\xi) \sum_{k=0}^{T-1} a_k^{1-1/q} \ .$$

Using the above inequality we obtain

$$\begin{aligned} \sum_{k=0}^{T-1} a_k^{1-1/q} &\geq \frac{1}{\left(n(1+\xi)(1+\varepsilon_1)^{1/q}(1-\varepsilon_1)^{-1}\right)^{1-1/q}} (\sum_{k=0}^{T-1} a_k^{1-1/q})^{1/q} (\sum_{k=0}^{T-1} \frac{1}{a_k^{1/q}})^{1-1/q} \\ &\geq \frac{T}{n^{1-1/q}(1+\xi)^{1-1/q}(1+\varepsilon_1)^{1/q-1/q^2}(1-\varepsilon_1)^{1/q-1}} \ , \end{aligned}$$

where the last inequality follows from Hölder's inequality. If we choose $T = \frac{n}{\varepsilon^2}$, this immediately gives

$$\sum_{k=0}^{T-1} (\hat{L}_{e_k} \bullet X_k)^{1-1/q} = \sum_{k=0}^{T-1} a_k^{1-1/q} \geq \frac{n^{1/q}}{\varepsilon^2}(1 - O(q\alpha + \varepsilon_1)) \ . \tag{8.30}$$

Finally, substituting (8.30) into (8.29), and choosing U_Y so that $M_Y \bullet U_Y = (1+\varepsilon_1)\lambda_{\mathsf{min}}(M_Y)$, we have

$$\begin{aligned} (1+\varepsilon_1)\lambda_{\mathsf{min}}(M_Y) &\geq -\frac{q(1+\varepsilon_1)n^{1/q}}{(q-1)\alpha} + (1-\xi)\frac{1}{(1+\varepsilon_1)^{3-3/q}} \sum_{k=0}^{T-1} (\hat{L}_{e_k} \bullet X_k)^{1-1/q} \\ &\geq -\frac{2qn^{1/q}}{(q-1)\alpha} + (1-\xi)\frac{1}{(1+\varepsilon_1)^{3-3/q}} \frac{n^{1/q}}{\varepsilon^2}(1 - O(q\alpha+\varepsilon_1)) \\ &\geq -\frac{2qn^{1/q}}{(q-1)\alpha} + \frac{n^{1/q}}{\varepsilon^2}(1 - O(q\alpha+\varepsilon_1)) \\ &\geq \frac{n^{1/q}}{\varepsilon^2}(1 - O(q\alpha + \varepsilon_1 + \varepsilon^2/\alpha)) \ . \end{aligned} \tag{8.31}$$

Above, the first inequality is due to our choice of e_k which satisfies

$$(1+\varepsilon_1)^3 \hat{L}_{e_k} \bullet Y_k \geq (1+\varepsilon_1)^2 \mathsf{Dot}(\hat{L}_{e_k}, Y_k) \geq \mathsf{Dot}(\hat{L}_{e_k}, X_k) \geq \hat{L}_{e_k} \bullet X_k \ . \tag{8.32}$$

Upper bounding $\lambda_{\mathsf{max}}(M_Y) - \lambda_{\mathsf{min}}(M_Y)$. This time, combining (8.28) and (8.29), as well as using (8.32), we compute that

$$\begin{aligned} \frac{1}{1+\xi}\big(M_Y \bullet U_X - \frac{qn^{1/q}}{(q-1)\alpha}\big) &\leq \frac{1}{1+\xi}\big(M_X \bullet U_X - \frac{qn^{1/q}}{(q-1)\alpha}\big) \\ &\leq \frac{(1+\varepsilon_1)^{3-3/q}}{1-\xi}\big(M_Y \bullet U_Y + \frac{q(1+\varepsilon_1)n^{1/q}}{(q-1)\alpha}\big) \ . \end{aligned}$$

Choosing U_X so that $M_Y \bullet U_X = \lambda_{\mathsf{max}}(M_Y)$, and U_Y so that $M_Y \bullet U_Y = (1+\varepsilon_1)\lambda_{\mathsf{min}}(M_Y)$, we can rewrite the above inequality as

$$\frac{1}{1+\xi}\big(\lambda_{\mathsf{max}}(M_Y) - \frac{qn^{1/q}}{(q-1)\alpha}\big) \leq \frac{(1+\varepsilon_1)^{3-3/q}}{1-\xi}(1+\varepsilon_1)\big(\lambda_{\mathsf{min}}(M_Y) + \frac{qn^{1/q}}{(q-1)\alpha}\big) \ . \quad (8.33)$$

To turn this joint multiplicative-additive error into a purely multiplicative one, we further rewrite it as

$$\begin{aligned}
\lambda_{\mathsf{max}}(M_Y) - \lambda_{\mathsf{min}}(M_Y) &\leq \frac{2\xi + O(\varepsilon_1)}{1-\xi}\lambda_{\mathsf{min}}(M_Y) + \frac{1+\xi+O(\varepsilon_1)}{1-\xi}\frac{qn^{1/q}}{(q-1)\alpha} + \frac{qn^{1/q}}{(q-1)\alpha} \\
&= \frac{2\xi + O(\varepsilon_1)}{1-\xi}\lambda_{\mathsf{min}}(M_Y) + \frac{2q}{q-1}\frac{1+O(\varepsilon_1)}{1-\xi}\frac{n^{1/q}}{\alpha} \\
&\leq \frac{2\xi + O(\varepsilon_1)}{1-\xi}\lambda_{\mathsf{min}}(M_Y) + \frac{2q}{q-1}\cdot\lambda_{\mathsf{min}}(M_Y)\frac{\varepsilon^2}{\alpha}(1+O(q\alpha+\varepsilon_1+\varepsilon^2/\alpha)) \\
&= \lambda_{\mathsf{min}}(M_Y)\cdot\Big(q\alpha + \frac{2q}{q-1}\frac{\varepsilon^2}{\alpha} + O(\varepsilon_1 + q\varepsilon^2 + \varepsilon_1\varepsilon^2/\alpha + \varepsilon^4/\alpha^2 + q^2\alpha^2)\Big) \ .
\end{aligned}$$

Above, the second inequality uses (8.31). Now, it is clear that by choosing $\alpha = \frac{\varepsilon\sqrt{2}}{\sqrt{q-1}} \leq \frac{1}{2q}$, we have

$$\begin{aligned}
\lambda_{\mathsf{max}}(M_Y) - \lambda_{\mathsf{min}}(M_Y) &\leq \lambda_{\mathsf{min}}(M_Y)\cdot\Big(\sqrt{\frac{8q^2}{q-1}}\cdot\varepsilon + O(\varepsilon_1 + q\varepsilon^2 + \varepsilon_1\varepsilon\sqrt{q})\Big) \\
&\leq \lambda_{\mathsf{min}}(M_Y)\cdot\Big(\sqrt{\frac{8q^2}{q-1}}\cdot\varepsilon + O(\varepsilon_1 + q\varepsilon^2)\Big) \ . \qquad \square
\end{aligned}$$

8.F.4 An Additional Property

Recall that in the previous subsection, we have constructed M_X and M_Y and proved that $\lambda_{\mathsf{min}}(M_Y)$ (and in fact $\lambda_{\mathsf{min}}(M_Y)$ as well) is at least $\Omega(n^{1/q}/\varepsilon^2)$. In this subsection, we shall show that $\lambda_{\mathsf{max}}(M_X)$ and $\lambda_{\mathsf{max}}(M_Y)$ can be made at most $O(n^{1/q}/\varepsilon^2)$ as well. While this additional property is not needed for proving Theorem 8.14, it shall become useful for proving the desired running time in the next section (see Appendix 8.G).

The following lemma ensures that if we stop the algorithm "whenever we are done", and thus choose possibly less than n/ε^2 matrices, then, $\lambda_{\mathsf{max}}(M_X)$ and $\lambda_{\mathsf{max}}(M_Y)$ can be properly upper bounded.

Lemma 8.15. *If one stops the algorithm either when $T = \frac{n}{\varepsilon^2}$ iterations are performed, or when the first time that $\sum_{k=0}^{T-1}\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1-1/q} \geq \frac{n^{1/q}}{\varepsilon^2}$ is satisfied, then the same result of Theorem 8.14 can be obtained, while we have an extra guarantee*

$$\lambda_{\mathsf{max}}(M_X), \lambda_{\mathsf{max}}(M_Y) \leq O\big(\frac{n^{1/q}}{\varepsilon^2}\big) \ .$$

Proof. Recall that in the proof of Theorem 8.14, we have only used the choice of $T = \frac{n}{\varepsilon^2}$ to deduce (8.30). For this reason, if instead of choosing exactly $T = \frac{n}{\varepsilon^2}$

matrices, we

stop the algorithm at the first time T such that $\sum_{k=0}^{T-1} \mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1-1/q} \geq \frac{n^{1/q}}{\varepsilon^2}$ *is satisfied,*

then we automatically have

$$\sum_{k=0}^{T-1} (\hat{L}_{e_k} \bullet X_k)^{1-1/q} \geq \frac{n^{1/q}}{\varepsilon^2}(1 - O(\varepsilon_1)) \ .$$

Replacing (8.30) with the above lower bound, all results claimed in Theorem 8.14 remain true.

In the rest of the proof, we will show that this early termination rule ensures a good upper bound on $\lambda_{\mathsf{max}}(M_X)$ and $\lambda_{\mathsf{max}}(M_Y)$. Indeed, at the time the algorithm is terminated, we must have

$$\sum_{k=0}^{T-1} (\hat{L}_{e_k} \bullet X_k)^{1-1/q} \leq \sum_{k=0}^{T-1} \mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1-1/q} \leq \frac{n^{1/q}}{\varepsilon^2} + O(1) \ . \tag{8.34}$$

This is because, since $\hat{L}_{e_k} \bullet X_k \leq I \bullet X_k = 1$ and thus $\mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1-1/q} \leq O(1)$, the value $\sum_{k=0}^{T-1} \mathsf{Dot}(\hat{L}_{e_k}, X_k)^{1-1/q}$ is incremented by at most $O(1)$ at each iteration. As a consequence, at the first iteration it exceeds $n^{1/q}/\varepsilon^2$, the summation must be at least $n^{1/q}/\varepsilon^2 + O(1)$.

Next, substituting (8.34) into (8.28), and choosing U_X so that $M_X \bullet U_X = \lambda_{\mathsf{max}}(M_X)$, we have

$$\lambda_{\mathsf{max}}(M_X) \leq \frac{q n^{1/q}}{(q-1)\alpha} + (1+\xi)\frac{n^{1/q}}{\varepsilon^2} + O(1) = O\big(\frac{n^{1/q}}{\varepsilon^2}\big) \ .$$

Finally, recalling that we have chosen $\mathsf{Dot}(\hat{L}_{e_k}, X_k) \leq (1+\varepsilon_1)^2 \mathsf{Dot}(\hat{L}_{e_k}, Y_k)$, this ensures that $(1+\varepsilon_1)^2 M_X \succeq M_Y$. In sum, we obtain that $\lambda_{\mathsf{max}}(M_Y) \leq O(\lambda_{\mathsf{max}}(M_X)) \leq O\big(\frac{n^{1/q}}{\varepsilon^2}\big)$. □

8.G Efficient Implementation for Graph Sparsifications

Recall from Appendix 8.F that in order to implement the algorithm described in Theorem 8.14, we need to

(C1) Ensure that each $\mathrm{Tr}\hat{L}_e$ is in $[1-\varepsilon_1, 1]$.

(C2) Compute at each iteration two reals $c^X, c^Y \in \mathbb{R}$ satisfying that $\mathrm{Tr}X_k \in [1, 1+\varepsilon_1]$ and $\mathrm{Tr}Y_k \in [1, 1+\varepsilon_1]$, where

$$X_k \stackrel{\text{def}}{=} \Big(c^X \cdot I - \sum_{j=0}^{k-1} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, X_j)^{1/q}}\Big)^{-q} \quad \text{and} \quad Y_k \stackrel{\text{def}}{=} \Big(\sum_{j=0}^{k-1} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, Y_j)^{1/q}} - c^Y \cdot I\Big)^{-q} \ .$$

(C3) Compute at each iteration $\mathsf{Dot}(\hat{L}_e, X_k)$ and $\mathsf{Dot}(\hat{L}_e, Y_k)$ which satisfy

$$\hat{L}_e \bullet X_k \le \mathsf{Dot}(\hat{L}_e, X_k) \le (1+\varepsilon_1)\hat{L}_e \bullet X_k \quad \text{and} \quad \hat{L}_e \bullet Y_k \le \mathsf{Dot}(\hat{L}_e, Y_k) \le (1+\varepsilon_1)\hat{L}_e \bullet Y_k \ .$$

In this section, we suppose that we are dealing with a spectral graph sparsification instance (see Appendix 8.B). In other words, we use I to denote $I_{\mathsf{im}(L_G)}$, and have $\hat{L}_e = \frac{L_G^{-1/2} L_e L_G^{-1/2}}{w_e}$, where $w_e = L_G^{-1} \bullet L_e$ is the effective resistance of edge $e \in [m]$.

Knowing this scaling factor w_e is somewhat important, because we need to ensure that $\mathrm{Tr}\hat{L}_e$ is between $1-\varepsilon_1$ and 1 according to (C1). Fortunately, Spielman and Srivastava [151] have given an algorithm that runs in nearly-linear time, and produces the effective resistances $L_G^{-1} \bullet L_e$ up to a multiplicative error of $1+\varepsilon_1$ for all edges $e \in [m]$, with probability at least $1 - n^{-\Omega(1)}$.

In other words, we can denote by $\hat{L}_e = \frac{L_G^{-1/2} L_e L_G^{-1/2}}{w_e}$, where each w_e only needs to be between $(1-\varepsilon_1) \cdot L_G^{-1} \bullet L_e$ and $L_G^{-1} \bullet L_e$.

We next wish to show how to implement (C2) and (C3) efficiently. Before that, let us claim that

Lemma 8.16. *Regardingless of how (C2) and (C3) are implemented, for all iterations, $c^X, c^Y \le O(\alpha \frac{n^{1/q}}{\varepsilon^2}) = O(\frac{n^{1/q}}{\sqrt{q}\varepsilon})$.*

Proof. It is first easy to see that $c^Y \le \alpha \cdot \lambda_{\mathsf{max}}(M_Y) \le O(\alpha \frac{n^{1/q}}{\varepsilon^2})$ owing to Lemma 8.15. Next, since $\mathrm{Tr}X_k \ge 1$, we must have

$$c^X \le \lambda_{\mathsf{max}}\Big(\sum_{j=0}^{k-1} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, X_j)^{1/q}}\Big) + n^{1/q} \le \alpha \cdot \lambda_{\mathsf{max}}(M_X) + n^{1/q} \le O(\alpha \frac{n^{1/q}}{\varepsilon^2}) \ . \qquad \square$$

Now, we are ready to prove the main theorem of this section.

Theorem 8.17. *In an amortized[a] running time of $\tilde{O}(\sqrt{q} n^{1/q} m / \varepsilon_1^2 \varepsilon)$ per iteration, we can implement (C2) and (C3) with probability at least $1 - n^{-\Omega(1)}$.*

Combining this with the fact that there are at mots $\frac{n}{\varepsilon^2}$ iterations, the total running time of our graph sparsification algorithm is

$$\tilde{O}\Big(\frac{\sqrt{q} n^{1+1/q} m}{\varepsilon_1^2 \varepsilon^3}\Big) \ .$$

[a] This amortization can be removed, but will result in a slightly more involved implementation to analyze.

Our proof below will make frequent uses of Lemma 8.18 and Lemma 8.19, two independent lemmas regarding how to efficiently compute matrix inversions of the form $(cI - A)^{-q}$ as well as $(A - cI)^{-q}$. The statements and proofs of these two lemmas are deferred to Appendix 8.G.1.

Proof. Both (C2) and (C3) are trivially implementable when $k = 0$, because $X_0 = Y_0 = \frac{1}{n} I$.

Suppose that both of them are implementable at iteration $k-1$. We proceed in 4 steps to prove that they are implementable at iteration k as well.

- **Step I**: prove (C3) for computing $\mathsf{Dot}(\hat{L}_e, X_k)$.

  Suppose X_k is given in the form of $X_k \stackrel{\text{def}}{=} \left(c^X \cdot I - \sum_{j=0}^{k-1} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, X_j)^{1/q}}\right)^{-q}$ for some $c^X > 0$, and it satisfies $\mathrm{Tr} X_k \in [1, 1+\varepsilon_1]$. (This is done by the inductive assumption.)

  Since $\mathrm{Tr} X_k \leq 1 + \varepsilon_1 \leq 3/2$, we must have

  $$X_k^{-1/q} = c^X \cdot I - \sum_{j=0}^{k-1} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, X_j)^{1/q}} \succeq \frac{2}{3} I \ .$$

  This inequality ensures that we can compute $X_k \bullet \hat{L}_e$ approximately (up to $1+\varepsilon_1$ error) using Lemma 8.18. Since c^X is no more than $O(n^{1/q}/\sqrt{q}\varepsilon)$ owing to Lemma 8.16, the running time for computing $X_k \bullet \hat{L}_e$ for all edges $e \in E$ is $\tilde{O}(c^X q m/\varepsilon_1^2) = \tilde{O}(\sqrt{q} n^{1/q} m/\varepsilon_1^2 \varepsilon)$.

- **Step II**: prove (C3) for computing $\mathsf{Dot}(\hat{L}_e, Y_k)$.

  Suppose Y_k is given in the form of $Y_k \stackrel{\text{def}}{=} \left(\sum_{j=0}^{k-1} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, Y_j)^{1/q}} - c^Y \cdot I\right)^{-q}$ for some real c^Y, and it satisfies $\mathrm{Tr} Y_k \in [1, 1+\varepsilon_1]$. (This is done by the inductive assumption.) Since $\mathrm{Tr} Y_k \leq 1 + \varepsilon_1 \leq 3/2$, we must have

  $$Y_k^{-1/q} = \sum_{j=0}^{k-1} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, Y_j)^{1/q}} - c^Y \cdot I \succeq \frac{2}{3} I \ .$$

  This inequality ensures that we can compute $Y_k \bullet \hat{L}_e$ approximately (up to $1+\varepsilon_1$ error) using Lemma 8.19. Since c^Y is no more than $O(n^{1/q}/\sqrt{q}\varepsilon)$ owing to Lemma 8.16, the running time for computing $Y_k \bullet \hat{L}_e$ for all edges $e \in E$ is $\tilde{O}(c^Y q m/\varepsilon_1^2) = \tilde{O}(\sqrt{q} n^{1/q} m/\varepsilon_1^2 \varepsilon)$.

- **Step III**: prove (C2) for X_k.

  Suppose that $X_{k-1} \stackrel{\text{def}}{=} \left(b^X \cdot I - \sum_{j=0}^{k-2} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, X_j)^{1/q}}\right)^{-q}$. Since $\mathrm{Tr} X_{k-1} \leq 1 + \varepsilon_1 \leq 3/2$, we must have

  $$X_{k-1}^{-1/q} = b^X \cdot I - \sum_{j=0}^{k-2} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, X_j)^{1/q}} \succeq \frac{2}{3} I \ .$$

  Recall that we have proved that $X_{k-1}^{-1/q} \succeq \frac{\hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, X_j)^{1/q}}$ (see Claim 8.10), combining it with the inequality above and the fact that $\alpha < 1/4$, we have

  $$b^X \cdot I - \sum_{j=0}^{k-1} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, X_j)^{1/q}} \succeq \frac{1}{2} I \ . \tag{8.35}$$

Now, we are ready to perform a binary search to find c^X. If one selects $c^X = b^X$, he will get $\mathrm{Tr}X_k \geq \mathrm{Tr}X_{k-1} \geq 1$, and therefore $c^X = b^X$ is a good lower bound for the choice of c^X. On the other hand, if one selects $c^X = b^X + n^{1/q}$, he will get $\mathrm{Tr}X_k \leq \mathrm{Tr}(n^{1/q}I)^{-q} = 1$, so $b^X + n^{1/q}$ is a good upper bound for the choice of c^X.

In sum, we can binary search c^X in the interval of $[b^X, b^X + n^{1/q}]$. For each such value of c^X in the process of the binary search, since c^X is no more than $O(n^{1/q}/\sqrt{q}\varepsilon)$ as per Lemma 8.16, one can apply Lemma 8.18 and approximately compute $\mathrm{Tr}(X_k) = \sum_e X_k \bullet \hat{L}_e$ up to a multiplicative error of $1 + \varepsilon_1$, in time $\tilde{O}(c^X qm/\varepsilon_1^2) = \tilde{O}(\sqrt{q}n^{1/q}m/\varepsilon_1^2\varepsilon)$.

Since the overhead for the binary search is $\tilde{O}(1)$, the total running time to compute c^X at an iteration is $\tilde{O}(\sqrt{q}n^{1/q}m/\varepsilon_1^2\varepsilon)$.

- **Step IV**: prove (C2) for Y_k.

  Suppose that $Y_{k-1} \stackrel{\text{def}}{=} \left(\sum_{j=0}^{k-2} \frac{\alpha\hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, Y_j)^{1/q}} - b^Y \cdot I\right)^{-q}$. Since $\mathrm{Tr}Y_{k-1} \leq 1+\varepsilon_1 \leq 3/2$, we must have

  $$Y_{k-1}^{-1/q} = \sum_{j=0}^{k-2} \frac{\alpha\hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, Y_j)^{1/q}} - b^Y \cdot I \succeq \frac{2}{3}I \ . \tag{8.36}$$

  It is clear from now that it suffices for us to search for $c^Y \geq b^Y$, because if one selects $c^Y = b^Y$, he will get $\mathrm{Tr}Y_k \leq \mathrm{Tr}Y_{k-1} \leq 1 + \varepsilon_1$, and therefore $c^Y = b^Y$ is a good lower bound. However, unlike Step III, one cannot perform a simple binary search on c^Y because there is no good upper bound for c^Y.[12]

  Instead, consider the following increment-and-binary-search algorithm. Beginning from b^Y, we first choose $c^Y = b^Y + \frac{1}{6}$. This choice of c^Y ensures that, according to (8.36),

  $$Y_k^{-1/q} = \sum_{j=0}^{k-1} \frac{\alpha\hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, Y_j)^{1/q}} - c^Y \cdot I \succeq \frac{1}{2}I \ .$$

  Therefore, we can compute $\mathrm{Tr}(Y_k) = \sum_e Y_k \bullet \hat{L}_e$ approximately using Lemma 8.19. If the approximation computation from Lemma 8.19 tells us that $\mathrm{Tr}(Y_k) \geq 1$, we stop the increment of c^Y. Otherwise, we conclude that $\mathrm{Tr}(Y_k)$ is still less than or equal to $1 + \varepsilon_1$, and continue to try $c^Y = b^T + \frac{i}{6}$ for $i = 2, 3, 4, \ldots$. We stop this increment until we find some integer i so that $\mathrm{Tr}(Y_k) \geq 1$.

[12] In fact, if one is allowed to compute the smallest eigenvalue of $\sum_{j=0}^{k-1} \frac{\alpha\hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, Y_j)^{1/q}}$, he can perform a binary search as described in Section 8.6. However, we have chosen not to implement that algorithm because the running time analysis for the max/min eigenvalue computation is only longer than the current one.

At this moment, we have that

$$\mathrm{Tr}\Big(\sum_{j=0}^{k-1}\frac{\alpha\hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j},Y_j)^{1/q}}-(b^Y+\frac{i-1}{6})\cdot I\Big)^{-q}\le 1+\varepsilon_1 \quad \text{and}$$
$$\mathrm{Tr}\Big(\sum_{j=0}^{k-1}\frac{\alpha\hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j},Y_j)^{1/q}}-(b^Y+\frac{i}{6})\cdot I\Big)^{-q}\ge 1 \ .$$

Therefore, we can perform a binary search for c^Y between $b^Y+\frac{i-1}{6}$ and $b^Y+\frac{i}{6}$ for, and in $\tilde{O}(1)$ time we can find some value in this interval which satisfies $\mathrm{Tr}(Y_k)\in[1,1+\varepsilon_1]$.

Again, since we always have $c^Y\le O(n^{1/q}/\sqrt{q}\varepsilon)$ owing to Lemma 8.16, the binary search step costs a running time that is at most $\tilde{O}(c^Y qm/\varepsilon_1^2)=\tilde{O}(\sqrt{q}n^{1/q}m/\varepsilon_1^2\varepsilon)$ owing to Lemma 8.19.

The incrementation procedure takes a running time $\tilde{O}(\sqrt{q}n^{1/q}m/\varepsilon_1^2\varepsilon)$ for each increment of $\frac{1}{6}$. However, throughout the algorithm, we increment c^Y by $1/6$ at most $O(n^{1/q}/\sqrt{q}\varepsilon)$ times in total as per Lemma 8.16. This running time, after amortization, is going to be dominated by that of the binary search.

Overall, we have shown that (C2) and (C3) can be implemented to run in $\tilde{O}(\sqrt{q}n^{1/q}m/\varepsilon_1^2\varepsilon)$ time (in amortization) per iteration. Since there are a total of at most $\frac{n}{\varepsilon^2}$ iterations, the desired running time is obtained. □

8.G.1 Missing Lemmas

In this subsection, we state and prove Lemma 8.18 and Lemma 8.19 for the efficient computations of the matrix inverses needed for the previous subsection.

Lemma 8.18. *Suppose that we are given positive reals c and $s_0,\ldots,s_{k-1}$ satisfying $cI-\sum_{j=0}^{k-1}s_j\check{L}_{e_j}\succeq\frac{1}{2}I$, where each $\check{L}_e$ is the normalized edge Laplacian and $k=O(m)$. Let q be any positive even integer. Then, we can compute a matrix $T\in\mathbb{R}^{m'\times m}$ in time $\tilde{O}(cqm/\varepsilon_1^2)$, where T has $m'=\Theta(\log n/\varepsilon_1^2)$ rows and satisfies that, with probability at least $1-n^{-\Omega(1)}$,*

$$\forall e\in E,\qquad X\bullet\check{L}_e\le\|T\chi_e\|_2^2\le(1+\varepsilon_1)X\bullet\check{L}_e\ ,\qquad \text{where } X\stackrel{\text{def}}{=}\Big(cI-\sum_{j=0}^{k-1}s_j\check{L}_{e_j}\Big)^{-q}\ .$$

Lemma 8.19. *Suppose we are given positive $s_0, \ldots, s_{k-1}$ and a possibly negative real c satisfying that $\sum_{j=0}^{k-1} s_j \check{L}_{e_j} - cI \succeq \frac{1}{2} I$, where each $\check{L}_e$ is the normalized edge Laplacian and $k = O(m)$. Let q be any positive even integer. Then, we can compute a matrix $T \in \mathbb{R}^{m' \times m}$ in time $\tilde{O}(cqm/\varepsilon_1^2)$, where T has $m' = \Theta(\log n/\varepsilon_1^2)$ rows and satisfies that, with probability at least $1 - n^{-\Omega(1)}$,*

$$\forall e \in E, \qquad Y \bullet \check{L}_e \le \|T\chi_e\|_2^2 \le (1+\varepsilon_1) Y \bullet \check{L}_e \ , \qquad \text{where } Y \stackrel{\text{def}}{=} \Big(\sum_{j=0}^{k-1} s_j \check{L}_{e_j} - cI \Big)^{-q} .$$

Our proofs to the above lemmas rely on the following auxiliary tools.

Auxiliary Tools

The first one is the famous Laplacian linear system solver, written in the matrix language.

Theorem 8.20. *For parameter $\alpha \in [0,1]$. Given any Laplacian matrix L that corresponds to a graph with m edges, there exist an approximation $\bar{L}^{-1}$ which satisfies that, with probability at least $1 - n^{-\Omega(1)}$, $(1-\delta)L^{-1} \preceq \bar{L}^{-1} \preceq (1+\delta)L^{-1}$, and for every vector $v \in \mathbb{R}^n$, $\bar{L}^{-1}v$ can be computed in time $\tilde{O}(m \log(1/\delta))$.*

Proof. The algorithms presented in [152] can be expressed as matrices $\bar{L}^{-1}$ which satisfy that, with high probability, for every $x \in \mathbb{R}^n$, the vectors $L^{-1}x$ and $\bar{L}^{-1}$ are close under the so-called L-norm, or in symbols, $\|\bar{L}^{-1}x - L^{-1}x\|_L^2 \le \delta^2 \|L^{-1}x\|_L^2$. After expanding this out using the definition of the L-norm, we have

$$\begin{aligned}
& x^T \big(\bar{L}^{-1} - L^{-1}\big) L \big(\bar{L}^{-1} - L^{-1}\big) x \le \delta^2 \cdot x^T L^{-1} L L^{-1} x \\
\Longrightarrow \quad & \big(\bar{L}^{-1} - L^{-1}\big) L \big(\bar{L}^{-1} - L^{-1}\big) \preceq \delta^2 \cdot L^{-1} \\
\Longrightarrow \quad & (L^{1/2} \bar{L}^{-1} L^{1/2} - I)^2 \preceq \delta^2 I \\
\Longrightarrow \quad & -\delta I \preceq L^{1/2} \bar{L}^{-1} L^{1/2} - I \preceq \delta I \\
\Longrightarrow \quad & (1-\delta) L^{-1} \preceq \bar{L}^{-1} \preceq (1+\delta) L^{-1} \ .
\end{aligned}$$

The running time $\tilde{O}(m \log(1/\delta))$ follows from that of [152]. □

The next two lemmas are the classical results on approximating $(I - A)^{-q}$ and $(A - I)^{-q}$ using Taylor expansions.

Lemma 8.21. *The polynomial $\mathsf{P}(A) = I + A + \cdots + A^{d-1}$ satisfies that for all $0 \preceq A \preceq (1-\delta)I$,*

$$0 \preceq (I - A)^{-1} - \mathsf{P}(A) \preceq (1-\delta)^d \cdot (I - A)^{-1} \ .$$

As a consequence, for every integer $q \ge 1$,

$$(1 - q(1-\delta)^d) \cdot (I - A)^{-q} \preceq \mathsf{P}^q(A) \preceq (I - A)^{-q} \ .$$

Proof. We first note that for every $x \in [0, 1-\delta]$, we have

$$0 \le \frac{1}{1-x} - (1 + x + \cdots + x^{d-1}) = x^d + x^{d+1} + \cdots = \frac{x^d}{1-x} \le \frac{(1-\delta)^d}{1-x} \ . \quad (8.37)$$

As a consequence, we have that

$$0 \preceq (I-A)^{-1} - (1 + A + \cdots + A^{d-1}) \preceq (1-\delta)^d \cdot (I-A)^{-1} \ ,$$

which can be proved by first assuming (without loss of generality) that A is diagonal, and then analyzing each diagonal entry using (8.37).

To prove the result for $(I-A)^{-q}$, we first notice that $(I-A)^{-1}$ and $\mathsf{P}(A)$ are commutable. Therefore, $\mathsf{P}(A) \preceq (I-A)^{-1}$ directly implies $\mathsf{P}^q(A) \preceq (I-A)^{-q}$, which gives one side of the inequality. To see the other side, we rewrite

$$(1 - (1-\delta)^d) \cdot (I-A)^{-1} \preceq \mathsf{P}(A) \ ,$$

and then take the q-th power on both sides. This yields

$$\big(1 - q(1-\delta)^d\big) \cdot (I-A)^{-q} \preceq (1 - (1-\delta)^d)^q \cdot (I-A)^{-q} \preceq \mathsf{P}^q(A) \ ,$$

which finishes the proof of the lemma. □

Lemma 8.22. *The polynomial* $\mathsf{P}(A) = A + A^2 + \cdots + A^d$ *satisfies that for all* $(1+\delta)I \preceq A$*,*

$$0 \preceq (A-I)^{-1} - \mathsf{P}(A^{-1}) \preceq (1+\delta)^{-d} \cdot (A-I)^{-1} \ .$$

As a consequence, for every integer $q \ge 1$*,*

$$(1 - q(1+\delta)^{-d}) \cdot (A-I)^{-q} \preceq \mathsf{P}^q(A^{-1}) \preceq (A-I)^{-q} \ .$$

Proof. We first note that for every $x \ge 1+\delta$, we have

$$0 \le \frac{1}{x-1} - (x^{-1} + x^{-2} + \cdots + x^{-d}) = x^{-d-1} + x^{-d-2} + \cdots = \frac{1}{x^d}\frac{1}{x-1} \le \frac{1}{(1+\delta)^d}\frac{1}{x-1} \ . \quad (8.38)$$

As a consequence, we have that

$$0 \preceq (A-I)^{-1} - (A^{-1} + A^{-2} + \cdots + A^{-d}) \preceq (1+\delta)^{-d} \cdot (A-I)^{-1} \ ,$$

which can be proved by first assuming (without loss of generality) that A is diagonal, and then analyzing each diagonal entry using (8.38).

To prove the result for $(A-I)^{-q}$, we first notice that $(A-I)^{-1}$ and $\mathsf{P}(A^{-1})$ are commutable. Therefore, $\mathsf{P}(A^{-1}) \preceq (A-I)^{-1}$ directly implies $\mathsf{P}^q(A^{-1}) \preceq (A-I)^{-q}$, which gives one side of the inequality. To see the other side, we rewrite

$$(1 - (1+\delta)^{-d}) \cdot (A-I)^{-1} \preceq \mathsf{P}(A^{-1}) \ ,$$

and then take the q-th power on both sides. This yields

$$\big(1 - q(1+\delta)^{-d}\big) \cdot (A-I)^{-q} \preceq (1 - (1+\delta)^{-d})^q \cdot (A-I)^{-q} \preceq \mathsf{P}^q(A^{-1}) \ ,$$

which finishes the proof of the lemma. □

Missing Proofs of Lemma 8.18 and 8.19

Lemma 8.18. *Suppose that we are given positive reals c and $s_0, \ldots, s_{k-1}$ satisfying $cI - \sum_{j=0}^{k-1} s_j \check{L}_{e_j} \succeq \frac{1}{2} I$, where each $\check{L}_e$ is the normalized edge Laplacian and $k = O(m)$. Let q be any positive even integer. Then, we can compute a matrix $T \in \mathbb{R}^{m' \times m}$ in time $\tilde{O}(cqm/\varepsilon_1^2)$, where T has $m' = \Theta(\log n/\varepsilon_1^2)$ rows and satisfies that, with probability at least $1 - n^{-\Omega(1)}$,*

$$\forall e \in E, \qquad X \bullet \check{L}_e \leq \|T\chi_e\|_2^2 \leq (1+\varepsilon_1) X \bullet \check{L}_e \ , \qquad \text{where } X \stackrel{\text{def}}{=} \Big(cI - \sum_{j=0}^{k-1} s_j \check{L}_{e_j}\Big)^{-q} \ .$$

Proof. Denoting by $A = \frac{1}{c}\sum_{j=0}^{k-1} s_j \check{L}_{e_j}$, we have $0 \preceq A \preceq (1 - \frac{1}{2c})I$ by the assumption. Now we apply Lemma 8.21, and let $\mathsf{P}(A)$ be the matrix polynomial of degree $d = \Theta(c \log(q/\varepsilon_1))$ from Lemma 8.21. By the approximation guarantee, we have for every edge $e \in E$,

$$X \bullet \check{L}_e = \Big(cI - \sum_{j=0}^{k-1} s_j \check{L}_{e_j}\Big)^{-q} \bullet \check{L}_e = \Big(1 \pm \frac{\varepsilon_1}{10}\Big) \cdot c^{-q} \cdot \mathsf{P}^q(A) \bullet \check{L}_e \ . \tag{8.39}$$

Therefore, it suffices for us to compute $\mathsf{P}^q(A) \bullet \check{L}_e$ for each possible edge e.

Next, let $\bar{L_G}^{-1}$ be the approximation of L_G^{-1} from Theorem 8.20 that satisfies

$$(1 - \frac{\varepsilon_1}{10dq}) L_G^{-1} \preceq \bar{L_G}^{-1} \preceq (1 + \frac{\varepsilon_1}{10dq}) L_G^{-1} \ .$$

Denoting by $L_s \stackrel{\text{def}}{=} \sum_{j=0}^{k-1} \frac{s_j}{c} L_{e_j}$, we have $A = L_G^{-1/2} L_s L_G^{-1/2}$. Accordingly, for every edge $e \in E$,

$$\begin{aligned}
\mathsf{P}^q(A) \bullet \check{L}_e &= \mathrm{Tr}\Big(\mathsf{P}^q\big(L_G^{-1/2} L_s L_G^{-1/2}\big)\ L_G^{-1/2} L_e L_G^{-1/2}\Big) \\
&= \mathrm{Tr}\Big(\mathsf{P}^q\big(L_G^{-1} L_s\big)\ L_G^{-1} L_e\Big) \\
&= \mathrm{Tr}\Big(\mathsf{P}^{q/2}\big(L_G^{-1} L_s\big)\ L_G^{-1}\ \mathsf{P}^{q/2}\big(L_s L_G^{-1}\big) L_e\Big) \\
&= \mathrm{Tr}\Big(\mathsf{P}^{q/2}\big(L_G^{-1} L_s\big)\ L_G^{-1}\ B^T W B^T\ L_G^{-1}\ \mathsf{P}^{q/2}\big(L_s L_G^{-1}\big) L_e\Big) \\
&\stackrel{①}{=} (1 \pm \varepsilon_1/10) \cdot \mathrm{Tr}\Big(\mathsf{P}^{q/2}\big(\bar{L_G}^{-1} L_s\big)\ \bar{L_G}^{-1}\ B^T W B^T\ \bar{L_G}^{-1}\ \mathsf{P}^{q/2}\big(L_s \bar{L_G}^{-1}\big) L_e\Big) \\
&= (1 \pm \varepsilon_1/10) \cdot w_e \cdot \chi_e^T \mathsf{P}^{q/2}\big(\bar{L_G}^{-1} L_s\big)\ \bar{L_G}^{-1}\ B^T W B^T\ \bar{L_G}^{-1}\ \mathsf{P}^{q/2}\big(L_s \bar{L_G}^{-1}\big)\chi_e \\
&= (1 \pm \varepsilon_1/10) \cdot w_e \cdot \Big\| W^{1/2} B^T \bar{L_G}^{-1} \mathsf{P}^{q/2}\big(L_s \bar{L_G}^{-1}\big)\chi_e \Big\|_2^2 \ .
\end{aligned} \tag{8.40}$$

Above, ① follows because each $\bar{L_G}^{-1}$ is a $(1 \pm \frac{\varepsilon_1}{10dq})$ approximation to L_G^{-1}, while we have at most $(d-1)q + 2 \leq dq$ copies of L_G^{-1} in any sequence of the matrix multiplication on the left hand side of ①.

For this reason, we can preprocess by computing $T' \stackrel{\text{def}}{=} Q W^{1/2} B^T \bar{L_G}^{-1} \mathsf{P}^{q/2}\big(L_s \bar{L_G}^{-1}\big) \in \mathbb{R}^{m' \times n}$, where $Q \in \mathbb{R}^{m' \times m}$ is some Johnson-Lindenstrauss random matrix with $m' =$

$\Theta(\log n/\varepsilon_1^2)$ rows. This matrix T' satisfies that, with probability at least $1-O(n^{-\Omega(1)})$,

$$\forall e \in E, \qquad \left\| QW^{1/2}B^T \bar{L_G}^{-1} \mathsf{P}^{q/2}\left(L_s \bar{L_G}^{-1}\right)\chi_e \right\|_2^2 = (1 \pm \varepsilon_1/10)\|T'\chi_e\|_2^2 \ . \tag{8.41}$$

Combining (8.39), (8.40), and (8.41) together, we have

$$\forall e \in E, \qquad X \bullet \check{L}_e = (1 \pm \varepsilon_1/3) \cdot c^{-q} \cdot w_e \cdot \|T'\chi_e\|_2^2 \ .$$

Defining $T \stackrel{\text{def}}{=} \left(\frac{1}{1-\varepsilon_1/3} \cdot c^{-q} \cdot w_e\right)^{1/2} \cdot T'$, we get the desired inequality in Lemma 8.18.

Finally, we emphasize that the above computation of T requires $\tilde{O}(dq \cdot m' \cdot m) = \tilde{O}(cqm/\varepsilon_1^2)$ time. This is because, each row of T can be computed by left multiplying each row of Q with the matrix $W^{1/2}B^T \bar{L_G}^{-1} \mathsf{P}^{q/2}\left(L_s \bar{L_G}^{-1}\right)$.[13] The running time now follows from (i) we need to compute vector-matrix multiplication $O(dq)$ times, which is the power of the polynomial $\mathsf{P}^{q/2}(\cdot)$, and (ii) Theorem 8.20 implies that for inversion $v^T \bar{L_G}^{-1}$ can be computed in time $\tilde{O}(m \log(dq/\varepsilon_1))$ for any vector v. □

Lemma 8.19. *Suppose we are given positive $s_0, \ldots, s_{k-1}$ and a possibly negative real c satisfying that $\sum_{j=0}^{k-1} s_j \check{L}_{e_j} - cI \succeq \frac{1}{2}I$, where each $\check{L}_e$ is the normalized edge Laplacian and $k = O(m)$. Let q be any positive even integer. Then, we can compute a matrix $T \in \mathbb{R}^{m' \times m}$ in time $\tilde{O}(cqm/\varepsilon_1^2)$, where T has $m' = \Theta(\log n/\varepsilon_1^2)$ rows and satisfies that, with probability at least $1 - n^{-\Omega(1)}$,*

$$\forall e \in E, \qquad Y \bullet \check{L}_e \le \|T\chi_e\|_2^2 \le (1+\varepsilon_1) Y \bullet \check{L}_e \ , \qquad \text{where } Y \stackrel{\text{def}}{=} \left(\sum_{j=0}^{k-1} s_j \check{L}_{e_j} - cI\right)^{-q} \ .$$

Proof. There are two cases: $c > 0$ or $c \le 0$. We begin with the case when $c > 0$.

Denoting by $A = \frac{1}{c}\sum_{j=0}^{k-1} s_j \check{L}_{e_j}$, we have $A \succeq (1 + \frac{1}{2c})I$ by the assumption. Now we apply Lemma 8.22, and let $\mathsf{P}(A)$ be the matrix polynomial of degree $d = \Theta(c \log(q/\varepsilon_1))$ from Lemma 8.22. By the approximation guarantee, we have for every edge $e \in E$,

$$Y \bullet \check{L}_e = \left(\sum_{j=0}^{k-1} s_j \check{L}_{e_j} - cI\right)^{-q} \bullet \check{L}_e = \left(1 \pm \frac{\varepsilon_1}{10}\right) \cdot c^{-q} \cdot \mathsf{P}^q(A^{-1}) \bullet \check{L}_e \ . \tag{8.42}$$

Therefore, it suffices for us to compute $\mathsf{P}^q(A^{-1}) \bullet \check{L}_e$ for each possible edge e.

[13] This can be implemented as follows. For any row vector of Q, denote it by $u^T \in \mathbb{R}^m$. We first sequentially compute

- $v^T \leftarrow u^T W^{1/2}$,
- $v^T \leftarrow v^T B^T$, and
- $v^T \leftarrow v^T \bar{L_G}^{-1}$.

Now, suppose $\mathsf{P}^{q/2}\left(L_s \bar{L_G}^{-1}\right) = \sum_{i=0}^{dq/2} c_i \left(L_s \bar{L_G}^{-1}\right)^i$ where each c_i is the coefficient of the i-th power term. We continue and compute

- $w^T \leftarrow \vec{0}$.
- For $i \leftarrow 0$ to $dq/2$,
  - $w^T \leftarrow w^T + v^T$.
  - $v^T \leftarrow v^T L_s$.
  - $v^T \leftarrow v^T \bar{L_G}^{-1}$.

In the end, the value of the row vector w^T is precisely the desired $u^T W^{1/2} B^T \bar{L_G}^{-1} \mathsf{P}^{q/2}\left(L_s \bar{L_G}^{-1}\right)$.

Denoting by $L_s \stackrel{\text{def}}{=} \sum_{j=0}^{k-1} \frac{s_j}{c} L_{e_j}$, we have $A^{-1} = L_G^{1/2} L_s^{-1} L_G^{1/2}$. Next, let $\bar{L_s}^{-1}$ and $\bar{L_G}^{-1}$ respectively be the approximation of L_s^{-1} and L_G^{-1} from Theorem 8.20 that satisfy

$$(1 - \frac{\varepsilon_1}{10dq}) L_s^{-1} \preceq \bar{L_s}^{-1} \preceq (1 + \frac{\varepsilon_1}{10dq}) L_s^{-1} \ , \text{ and}$$
$$(1 - \frac{\varepsilon_1}{10dq}) L_G^{-1} \preceq \bar{L_G}^{-1} \preceq (1 + \frac{\varepsilon_1}{10dq}) L_G^{-1} \ .$$

Accordingly, for every edge $e \in E$,

$$\begin{aligned}
\mathsf{P}^q(A^{-1}) \bullet \check{L}_e &= \mathrm{Tr}\Big(\mathsf{P}^q\big(L_G^{1/2} L_s^{-1} L_G^{1/2}\big)\ L_G^{-1/2} L_e L_G^{-1/2}\Big) \\
&= \mathrm{Tr}\Big(\mathsf{P}^q\big(L_s^{-1} L_G\big)\ L_G^{-1} L_e\Big) \\
&= \mathrm{Tr}\Big(\mathsf{P}^{q/2}\big(L_s^{-1} L_G\big)\ L_G^{-1}\ \mathsf{P}^{q/2}\big(L_G L_s^{-1}\big) L_e\Big) \\
&= \mathrm{Tr}\Big(\mathsf{P}^{q/2}\big(L_s^{-1} L_G\big)\ L_G^{-1}\ B^T W B^T\ L_G^{-1}\ \mathsf{P}^{q/2}\big(L_G L_s^{-1}\big) L_e\Big) \\
&\stackrel{①}{=} (1 \pm \varepsilon_1/10) \cdot \mathrm{Tr}\Big(\mathsf{P}^{q/2}\big(\bar{L_s}^{-1} L_G\big)\ \bar{L_G}^{-1}\ B^T W B^T\ \bar{L_G}^{-1}\ \mathsf{P}^{q/2}\big(L_G \bar{L_s}^{-1}\big) L_e\Big) \\
&= (1 \pm \varepsilon_1/10) \cdot w_e \cdot \chi_e^T \mathsf{P}^{q/2}\big(\bar{L_s}^{-1} L_G\big)\ \bar{L_G}^{-1}\ B^T W B^T\ \bar{L_G}^{-1}\ \mathsf{P}^{q/2}\big(L_G \bar{L_s}^{-1}\big) \chi_e \\
&= (1 \pm \varepsilon_1/10) \cdot w_e \cdot \Big\| W^{1/2} B^T \bar{L_G}^{-1} \mathsf{P}^{q/2}\big(L_G \bar{L_s}^{-1}\big) \chi_e \Big\|_2^2 \qquad (8.43)
\end{aligned}$$

Above, ① follows because each $\bar{L_s}^{-1}$ (resp. $\bar{L_G}^{-1}$) is a $(1 \pm \frac{\varepsilon_1}{10dq})$ approximation to L_s^{-1} (resp. L_G^{-1}), while we have at most $(d-1)q + 2 \le dq$ copies of L_s^{-1} and L_G^{-1} in any sequence of the matrix multiplication on the left hand side of ①.

For this reason, we can preprocess by computing $T' \stackrel{\text{def}}{=} Q W^{1/2} B^T \bar{L_G}^{-1} \mathsf{P}^{q/2}\big(L_G \bar{L_s}^{-1}\big) \in \mathbb{R}^{m' \times n}$, where $Q \in \mathbb{R}^{m' \times m}$ is some Johnson-Lindenstrauss random matrix with $m' = \Theta(\log n / \varepsilon_1^2)$ rows. This matrix T' satisfies that, with probability at least $1 - O(n^{-\Omega(1)})$,

$$\forall e \in E, \qquad \Big\| Q W^{1/2} B^T \bar{L_G}^{-1} \mathsf{P}^{q/2}\big(L_G \bar{L_s}^{-1}\big) \chi_e \Big\|_2^2 = (1 \pm \varepsilon_1/10) \|T' \chi_e\|_2^2 \ . \qquad (8.44)$$

Combining (8.42), (8.43), and (8.44), we have

$$\forall e \in E, \qquad Y \bullet \check{L}_e = (1 \pm \varepsilon_1/3) \cdot c^{-q} \cdot w_e \cdot \|T' \chi_e\|_2^2 \ .$$

Defining $T \stackrel{\text{def}}{=} \big(\frac{1}{1 - \varepsilon_1/3} \cdot c^{-q} \cdot w_e\big)^{1/2} \cdot T'$, we get the desired inequality in Lemma 8.19.

Finally, we emphasize that the computation of T requires $\tilde{O}(dq \cdot m' \cdot m) = \tilde{O}(dqm/\varepsilon_1^2)$ time. This is because, each row of T can be computed by left multiplying each row of Q with the matrix $W^{1/2} B^T \bar{L_G}^{-1} \mathsf{P}^{q/2}\big(L_G \bar{L_s}^{-1}\big)$.[14] The running time now follows from (i) we need to compute vector-matrix multiplication $O(dq)$ times, which is the power of the polynomial $\mathsf{P}^{q/2}(\cdot)$, and (ii) Theorem 8.20 implies the inversions $v^T \bar{L_G}^{-1}$ and $v^T \bar{L_s}^{-1}$ can both be computed in time $\tilde{O}(m \log(dq/\varepsilon_1))$, for any vector v.

[14] This can be implemented in a similar manner as discussed in Footnote 13.

In the second case, if $c \leq 0$, we can write

$$Y = \Big(\sum_{j=0}^{k-1} s_j \check{L}_{e_j} - cI\Big)^{-q} = \Big(L_G^{-1/2}(L_s - cL_G)L_G^{-1/2}\Big)^{-q} \ .$$

Therefore, denoting by $L'_s = L_s - cL_G$, which is another graph Laplacian matrix (with positive edge weights), we can write

$$\begin{aligned} Y \bullet \check{L}_e &= \mathrm{Tr}\Big(\big(L_G^{-1/2} L'_s L_G^{-1/2}\big)^{-q}\ L_G^{-1/2} L_e L_G^{-1/2}\Big) \\ &= \mathrm{Tr}\Big(\big(L_s'^{-1} L_G\big)^{-q/2}\ L_G^{-1}\ \big(L_G L_s'^{-1}\big)^{-q/2} L_e\Big) \\ &= w_e \cdot \chi_e^T \big(L_s'^{-1} L_G\big)^{-q/2}\ L_G^{-1} B^T W B L_G^{-1}\ \big(L_G L_s'^{-1}\big)^{-q/2} \chi_e \\ &= w_e \cdot \big\|W^{1/2} B L_G^{-1}\ \big(L_G L_s'^{-1}\big)^{-q/2} \chi_e\big\|_2^2 \ . \end{aligned}$$

It is now clear that similar to the previous case, we can approximately compute $L_s'^{-1}$ and L_G^{-1} using Theorem 8.20, and apply the Johnson-Lindenstrauss dimension reduction. We skip the detailed proofs here because it is only a repetition. □

8.H Efficient Implementation for Other Problems

As we have seen in Appendix 8.G, Lemma 8.18 and Lemma 8.19 are at the core of our efficient implementation for the graph sparsification problem. For each other possible sparsification problem, as long as these two lemmas can be properly revised, we can also obtain fast running times. Let us illustrate how to obtain such running times for two applications below.

Sparsifying sums of rank-1 matrices. To solve the problem in Theorem 8.2, it is not hard to verify that Lemma 8.18 can be revised as follows:

Suppose that we are given positive reals c and $s_0, \ldots, s_{k-1}$ satisfying $cI - \sum_{j=0}^{k-1} s_j \hat{L}_{e_j} \succeq \frac{1}{2} I$, where each $\hat{L}_{e_j} = v_{e_j} v_{e_j}^T$ is an explicit $n \times n$ rank-1 matrix and $k = O(m)$. Let q be any positive even integer. Then, we can compute a matrix $T \in \mathbb{R}^{m' \times n}$ in time $\tilde{O}(cqn^2/\varepsilon_1^2)$, where T has $m' = \Theta(\log n/\varepsilon_1^2)$ rows and satisfies that, with probability at least $1 - n^{-\Omega(1)}$,

$$\forall e \in E, \qquad X \bullet \hat{L}_e \leq \|T v_e\|_2^2 \leq (1+\varepsilon_1) X \bullet \hat{L}_e \ , \qquad \text{where } X \stackrel{\text{def}}{=} \Big(cI - \sum_{j=0}^{k-1} s_j \hat{L}_{e_j}\Big)^{-q} \ .$$

The key idea for proving the above variant of Lemma 8.18 is to note that the matrix inequality $cI - \sum_{j=0}^{k-1} s_j \hat{L}_{e_j} \succeq \frac{1}{2} I$ implies that the condition number for PSD matrix $M \stackrel{\text{def}}{=} cI - \sum_{j=0}^{k-1} s_j \hat{L}_{e_j}$ is at most $O(c)$. Therefore, one can use for instance steepest descent (or even conjugate gradient or Chebyshev method) to compute $M^{-1} v$ in time $O(cn^2)$ for every vector $v \in \mathbb{R}^n$. Next, one can apply the similar Johnson-Lindenstrauss dimension reduction as presented in the proof of Lemma 8.18.

A similar variant of Lemma 8.19 can be proved similarly.

In sum, each iteration of our Appendix 8.F is dominated by the computational time need to (1) compute the matrix $T \in \mathbb{R}^{m' \times n}$, which takes time $\tilde{O}(cqn^2/\varepsilon_1^2) = \tilde{O}(\sqrt{q}n^{2+1/q}/\varepsilon\varepsilon_1^2)$, and (2) compute Tv_e for all $e \in [m]$, which takes time $O(mn/\varepsilon_1^2)$. Taking into account that we have $T = n/\varepsilon^2$ such iterations, this is a total running time of

$$O\Big(\frac{\sqrt{q}n^{3+1/q}}{\varepsilon^2\varepsilon_1^2} + \frac{mn^2}{\varepsilon_1^2\varepsilon^2}\Big) \ .$$

Subgraph sparsification. Given a weighted undirected graph G that can be decomposed into edge-disjoint subgraphs, the goal of linear-sized subgraph sparsification is to construct a $(1+O(\varepsilon))$-spectral sparsifier G' to G, so that G' consists only of the reweighted versions of at most n/ε^2 given subgraphs.

In symbols, suppose that the edges of some weighted undirected graph G of n vertices and m' edges are decomposed into a disjoint union $E = \biguplus_{i=1}^{m} E_i$. We are interested in finding scalars $s_e \geq 0$ with $|\{e : s_e > 0\}| \leq O(n/\varepsilon^2)$ such that, letting $L \stackrel{\text{def}}{=} \sum_{e=1}^{m} s_e \cdot L_{G[E_e]}$, where L_{E_e} is the graph Laplacian matrix on the subgraph of G induced by E_e, we have $L_G \preceq L \preceq (1+\varepsilon)L_G$.

For this sparsification problem, for each $e \in [m]$, we define $\hat{L}_e = \frac{L_G^{-1/2} L_{G[E_e]} L_G^{-1/2}}{w_e}$ to be the normalized subgraph Laplacian scaled by w_e. Here, w_e is the scaling parameter which ensures that $\mathrm{Tr}\hat{L}_e$ is between $1-\varepsilon_1$ and 1. (It suffices to compute $L_G^{-1} \bullet L_{G[E_e]}$ up to a multiplicative $1+\varepsilon_1$ error, and then assign $w_e \approx L_G^{-1} \bullet L_{G[E_e]}$.)

For this particular problem, we do not even need to revise Lemma 8.18 or Lemma 8.19. Recall that we only need to compute 'matrix inversions' of the form

$$\Big(c^X \cdot I - \sum_{j=0}^{k-1} \frac{\alpha \hat{L}_{e_j}}{\mathsf{Dot}(\hat{L}_{e_j}, X_j)^{1/q}}\Big)^{-q} \bullet \hat{L}_e \ ,$$

while each $\hat{L}_{e_j}$ is now —instead of a single (scaled) edge Laplacian matrix— the summation of a few (scaled) edge Laplacian matrices. This remains to be the same problem Lemma 8.18 is trying to implement. The total running time for this subgraph sparsification is therefore

$$\tilde{O}\Big(\frac{\sqrt{q}n^{1+1/q}m'}{\varepsilon_1^2\varepsilon^3}\Big) \ .$$

Bibliography

[1] Jacob Abernethy and Alexander Rakhlin. Beating the adaptive bandit with high probability. In *Proceedings of the 22nd Conference on Learning Theory - COLT' 09*, pages 280–289, 2009. 214

[2] Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin. Interior-Point Methods for Full-Information and Bandit Online Learning. *IEEE Transactions on Information Theory*, 58(7):4164–4175, July 2012. An earlier version of this paper has appeared in COLT'08. 211, 212

[3] Dilip Abreu and Hitoshi Matsushima. Virtual implementation in iteratively undominated strategies: Complete information. *Econometrica*, 60(5):993–1008, 1992. 44

[4] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral Sparsification and Regret Minimization Beyond Multiplicative Updates. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, STOC '15, 2015. 163, 186, 191, 207

[5] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *ArXiv e-prints*, abs/1407.1537, July 2014. 89, 116, 120, 126, 143, 147, 153, 155

[6] Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly-Linear Time Positive LP Solver with Faster Convergence Rate. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, STOC '15, 2015. 89, 97, 118, 143, 185, 187, 198

[7] Zeyuan Allen-Zhu and Lorenzo Orecchia. Using optimization to break the epsilon barrier: A faster and simpler width-independent algorithm for solving positive linear programs in parallel. In *Proceedings of the 26th ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, 2015. 89, 97, 106, 115, 143, 145, 146, 147, 149, 150, 151, 157, 175, 185, 187, 188, 189, 190, 194, 195, 196, 198, 199, 202, 204, 205, 211, 212, 214

[8] David G. Anderson, Ming Gu, and Christopher Melgaard. An Efficient Algorithm for Unweighted Spectral Graph Sparsification. *ArXiv e-prints*, abs/1410.4273, October 2014. 212, 224

[9] Sanjeev Arora, Elad Hazan, and Satyen Kale. Fast Algorithms for Approximate Semidefinite Programming using the Multiplicative Weights Update Method. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 339–348. IEEE, 2005. 90, 108, 187, 210, 212

[10] Sanjeev Arora, Elad Hazan, and Satyen Kale. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing*, 8:121–164, 2012. 90, 94, 107, 108, 116, 117, 143, 144, 145, 149, 162, 163, 187, 209, 210, 211, 212

[11] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC '07*, page 227, New York, New York, USA, 2007. ACM Press. 186, 187, 188, 208, 210, 211, 212

[12] Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An $O(\log n/\log\log n)$-approximation Algorithm for the Asymmetric Traveling Salesman Problem. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms - SODA '10*, pages 379–389, 2010. 210

[13] Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Minimax policies for combinatorial prediction games. *Proceedings of COLT 2011*, 2011. 212

[14] Robert J. Aumann. Utility theory without the completeness axiom. *Econometrica*, 30(3):445–462, July 1962. 17, 44

[15] Lawrence M. Ausubel and Paul Milgrom. The lovely but lonely vickrey auction. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, page Ch. 1. MIT Press, 2006. 18

[16] Baruch Awerbuch, Yossi Azar, and Rohit Khandekar. Fast load balancing via bounded best response. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 314–322, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics. 134

[17] Baruch Awerbuch and Rohit Khandekar. Stateless distributed gradient descent for positive linear programs. *Proceedings of the fourtieth annual ACM symposium on Theory of computing - STOC 08*, page 691, 2008. 117, 118, 119, 120, 121, 122, 132, 133, 134, 135, 143, 145, 146, 187, 190

[18] Baruch Awerbuch and Rohit Khandekar. Stateless near optimal flow control with poly-logarithmic convergence. In *LATIN 2008: Theoretical Informatics*, pages 580–592. Springer, 2008. 134

[19] Baruch Awerbuch and Rohit Khandekar. Greedy distributed optimization of multi-commodity flows. *Distributed Computing*, 21(5):317–329, 2009. 134

[20] Baruch Awerbuch, Rohit Khandekar, and Satish Rao. Distributed algorithms for multicommodity flow problems via approximate steepest descent framework. *ACM Transactions on Algorithms*, 9(1):1–14, December 2012. 116, 144

[21] Moshe Babaioff, Ron Lavi, and Elan Pavlov. Single-value combinatorial auctions and implementation in undominated strategies. In *SODA '06: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1054–1063, New York, NY, USA, 2006. ACM. 17, 44

[22] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *The Journal of Machine Learning Research*, 6:1705–1749, 2005. 108, 222

[23] Nikhil Bansal, Uriel Feige, Robert Krauthgamer, Konstantin Makarychev, Viswanath Nagarajan, Joseph (Seffi) Naor, and Roy Schwartz. Min-max Graph Partitioning and Small Set Expansion. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 17–26. IEEE, October 2011. 210

[24] Yair Bartal, John W. Byers, and Danny Raz. Global optimization using local information with applications to flow control. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 303–312. IEEE Comput. Soc, 1997. 97, 116, 117, 118, 119, 143, 144, 145, 146, 187, 190

[25] Yair Bartal, John W. Byers, and Danny Raz. Fast, Distributed Approximation Algorithms for Positive Linear Programming with Applications to Flow Control. *SIAM Journal on Computing*, 33(6):1261–1279, January 2004. 97, 116, 117, 118, 133, 143, 144, 145, 146, 187, 190

[26] Joshua Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-Ramanujan Sparsifiers. *SIAM Review*, 56(2):315–334, May 2014. 207, 208, 218, 224

[27] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, January 2013. 90, 91, 92, 94, 112, 120, 128, 147, 210

[28] András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, pages 47–55, New York, New York, USA, 1996. ACM Press. 207

[29] András A. Benczúr and David R. Karger. Randomized Approximation Schemes for Cuts and Flows in Capacitated Graphs. Technical report, July 2002. 207

[30] Truman F. Bewley. Knightian decision theory. Part I. *Decisions in Economics and Finance*, 25(2):79–110, 2002. Earlier version appeared as discussion paper no. 807 of the Cowles Foundation at Yale University, November 1986. 15, 16, 44

[31] Rajendra Bhatia. *Matrix Analysis*, volume 169 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 1997. 213

[32] D. Bienstock and G. Iyengar. Faster approximation algorithms for packing and covering problems. Technical report, Columbia University, September 2004. Preliminary version published in STOC '04. 117, 143, 145, 187

[33] Aaron L. Bodoh-Creed. Ambiguous beliefs and mechanism design. *Games and Economic Behavior*, 75(2):518–537, 2012. 17, 44

[34] Subir Bose, Emre Ozdenoren, and Andreas Pape. Optimal auctions with ambiguity. *Theoretical Economics*, 1(4):411–438, December 2006. 17, 44

[35] Subir Bose and Ludovic Renou. Mechanism design with ambiguous communication devices. *Econometrica*, 82(5):1853–1872, 2014. 17

[36] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and trends in machine learning*, 5(1):1–122, 2012. 212

[37] Dave Buchfuhrer, Shaddin Dughmi, Hu Fu, Robert Kleinberg, Elchanan Mossel, Christos Papadimitriou, Michael Schapira, Yaron Singer, and Chris Umans. Inapproximability for vcg-based combinatorial auctions. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 518–536, 2010. 39

[38] John Byers and Gabriel Nasser. Utility-based decision-making in wireless sensor networks. In *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*, pages 143–144. IEEE, 2000. 116, 144

[39] Marcel K. de Carli Silva, Nicholas J. A. Harvey, and Cristiane M. Sato. Sparse Sums of Positive Semidefinite Matrices. Technical report, July 2011. 207, 208, 212

[40] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, 2006. 168, 210, 226

[41] Moses Charikar, Tom Leighton, Shi Li, and Ankur Moitra. Vertex sparsifiers and abstract rounding algorithms. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 265–274, 2010. 207

[42] Alessandro Chiesa, Silvio Micali, and Zeyuan Allen Zhu. Mechanism design with approximate valuations. In *Proceedings of the 3rd Innovations in Theoretical Computer Science conference*, ITCS '12, 2012. 45

[43] Alessandro Chiesa, Silvio Micali, and Zeyuan Allen Zhu. Knightian Self Uncertainty in the VCG Mechanism for Unrestricted Combinatorial Auctions. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, 2014. 39

[44] Alessandro Chiesa, Silvio Micali, and Zeyuan Allen Zhu. Knightian analysis of the Vickrey mechanism. *Econometrica*, 2015. To appear. 15

[45] C. G. Chorus, T. A. Arentze, and H. J. P. Timmermans. Spatial choice: a matter of utility or regret? *Environment and Planning B: Planning and Design*, 36(3):538–551, 2009. 45, 82

[46] Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd annual ACM symposium on Theory of computing - STOC '11*, page 273, New York, New York, USA, October 2011. ACM Press. 90, 210

[47] Fabián A. Chudak and Vânia Eleutério. Improved Approximation Schemes for Linear Programming Relaxations of Combinatorial Optimization Problems. In *Proceedings of the 11th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 81–96, 2005. 143, 145, 187

[48] Vincent Conitzer and Tuomas Sandholm. Complexity of (iterated) dominance. In *Proceedings of the 6th ACM conference on Electronic commerce*, EC '05, pages 88–97, New York, NY, USA, 2005. ACM. 86

[49] Eric Danan. Randomization vs. selection: How to choose in the absence of preference? *Management Science*, 56:503–518, March 2010. 17, 44

[50] Partha S. Dasgupta, Peter J. Hammond, and Eric S. Maskin. The implementation of social choice rules: Some general results on incentive compatibility. *Review of Economic Studies*, 46(2):185–216, April 1979. 26

[51] Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms - SODA '11*, pages 235–254, 2011. 210

[52] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *The Journal of Machine Learning Research*, 13(1):165–202, 2012. 94

[53] Alfredo Di Tillio, Nenad Kos, and Matthias Messner. The design of ambiguous mechanisms. Technical report, 2012. 17

[54] Ran Duan and Seth Pettie. Linear-Time Approximation for Maximum Weight Matching. *Journal of the ACM*, 61(1):1–23, January 2014. 146

[55] Juan Dubra, Fabio Maccheroni, and Efe A. Ok. Expected utility theory without the completeness axiom. *Journal of Economic Theory*, 115(1):118–133, March 2004. 17, 44

[56] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite Objective Mirror Descent. In *Proceedings of the 23rd Annual Conference on Learning Theory - COLT '10*, number 1, 2010. 92

[57] Shaddin Dughmi and Jan Vondrák. Limitations of randomized mechanisms for combinatorial auctions. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 502–511. IEEE, 2011. 39

[58] Richard Engelbrecht-Wiggans. The effect of regret on optimal bidding in auctions. *Management Science*, 35(6):685–692, 1989. 45, 85

[59] Richard Engelbrecht-Wiggans and Elena Katok. Regret in auctions: Theory and evidence. *Economic Theory*, 33(1):81–101, 2007. 45, 82

[60] Uriel Feige and Moshe Tennenholtz. Mechanism design with uncertain inputs: (to err is human, to forgive divine). In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, STOC '11, pages 549–558, New York, NY, USA, 2011. ACM. 44

[61] Olivier Fercoq and Peter Richtárik. Accelerated, Parallel and Proximal Coordinate Descent. *ArXiv e-prints*, abs/1312.5799:25, December 2013. 107, 147, 153, 167

[62] Emel Filiz and Erkut Y Ozbay. Auctions with anticipated regret: Theory and experiment. *The American Economic Review*, 97(4):1407–1418, 2007. 45, 82

[63] Lisa K. Fleischer. Approximating Fractional Multicommodity Flow Independent of the Number of Commodities. *SIAM Journal on Discrete Mathematics*, 13(4):505–520, January 2000. 116, 144

[64] Vincy Fon and Yoshihiko Otani. Classical welfare theorems with non-transitive and non-complete preferences. *Journal of Economic Theory*, 20(3):409–418, June 1979. 17, 44

[65] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995. 90, 107, 108, 212

[66] Drew Fudenberg and Jean Tirole. *Game theory. 1991.* MIT Press, 1991. 22, 47

[67] D. Gale and A. Mas-Colell. An equilibrium existence theorem for a general model without ordered preferences. *Journal of Mathematical Economics*, 2(1):9–15, March 1975. 17, 44

[68] Naveen Garg and Jochen Könemann. Faster and Simpler Algorithms for Multicommodity Flow and Other Fractional Packing Problems. *SIAM Journal on Computing*, 37(2):630–652, January 2007. 116, 119, 144, 210

[69] Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica: journal of the Econometric Society*, pages 587–601, 1973. 26

[70] Itzhak Gilboa and David Schmeidler. Maxmin expected utility with non-unique prior. *Journal of Mathematical Economics*, 18(2):141–153, April 1989. 16, 17, 24, 44

[71] Joseph Y. Halpern and Rafael Pass. Iterated regret minimization: A new solution concept. *Games and Economic Behavior*, 74(1):184–207, January 2012. A preliminary version appeared in IJCAI'09. 45, 85, 86

[72] Elad Hazan. The convex optimization approach to regret minimization. In *Optimization for machine learning*, chapter 10, pages 287–304. MIT press, 2012. 209, 210

[73] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, August 2007. 105

[74] Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. In *Proceedings of the 25th Annual Conference on Learning Theory - COLT '12*, pages 38.1—-38.13, 2012. 212, 214

[75] David A Hensher, William H Greene, and Caspar G Chorus. Random regret minimization or random utility maximization: an exploratory analysis in the context of automobile fuel choice. *Journal of Advanced Transportation*, 2011. 45, 82

[76] Nathanael Hyafil and Craig Boutilier. Regret Minimizing Equilibria and Mechanisms for Games with Strict Type Uncertainty. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 268–277, July 2004. 45, 83, 86

[77] Garud Iyengar, David J. Phillips, and Cliff Stein. Feasible and accurate algorithms for covering semidefinite programs. In *SWAT*, pages 150–162, 2010. 186, 187

[78] Garud Iyengar, David J. Phillips, and Clifford Stein. Approximating semidefinite packing programs. *SIAM Journal on Optimization*, 21(1):231–268, 2011. 186, 187

[79] Matthew O. Jackson. Implementation in undominated strategies: A look at bounded mechanisms. *Review of Economic Studies*, 59(4):757–75, October 1992. 17, 22, 44, 47, 83, 85

[80] Matthew O. Jackson, Thomas Palfrey, and Sanjay Srivastava. Undominated Nash implementation in bounded mechanisms. *Games and Economic Behavior*, 6(3):474–501, 1994. 17, 44

[81] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. QIP = PSPACE. *Journal of the ACM (JACM)*, 58(6):30, 2011. 186, 187, 211

[82] Rahul Jain, Sarvagya Upadhyay, and John Watrous. Two-message quantum interactive proofs are in pspace. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 534–543. IEEE, 2009. 186, 187

[83] Rahul Jain and John Watrous. Parallel approximation of non-interactive zero-sum quantum games. In *Computational Complexity, 2009. CCC'09. 24th Annual IEEE Conference on*, pages 243–253. IEEE, 2009. 186, 187

[84] Rahul Jain and Penghui Yao. A Parallel Approximation Algorithm for Positive Semidefinite Programming. *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 463–471, October 2011. 185, 187, 188

[85] Rahul Jain and Penghui Yao. A parallel approximation algorithm for mixed packing and covering semidefinite programs. *ArXiv e-prints*, abs/1201.6090, January 2012. 185, 187, 188, 190, 195

[86] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984. 220

[87] Anatoli Juditsky. Convex optimization ii: Algorithms. Lecture notes, November 2013. 89, 95, 97

[88] Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An Almost-Linear-Time Algorithm for Approximate Max Flow in Undirected Graphs, and its Multicommodity Generalizations. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms - SODA '14*, number 1 in STOC '14, April 2014. 90, 91, 92, 107, 207

[89] Philip Klein and Hsueh-I Lu. Efficient approximation algorithms for semidefinite programs arising from max cut and coloring. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 338–347. ACM, 1996. 186

[90] Philip Klein and Neal Young. On the number of iterations for dantzig-wolfe optimization and packing-covering approximation algorithms. In Gérard Cornuéjols, Rainer E. Burkard, and Gerhard J. Woeginger, editors, *Integer Programming and Combinatorial Optimization*, volume 1610 of *Lecture Notes in Computer Science*, pages 320–327. Springer Berlin Heidelberg, 1999. 146

[91] Frank H. Knight. *Risk, Uncertainty and Profit*. Houghton Mifflin, 1921. 15, 44

[92] Christos Koufogiannakis and Neal E. Young. A Nearly Linear-Time PTAS for Explicit Fractional Packing and Covering Linear Programs. *Algorithmica*, pages 494–506, March 2013. Previously appeared in FOCS '07. 118, 143, 145, 146, 187

[93] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, January 2011. 95

[94] Yin Tat Lee, Satish Rao, and Nikhil Srivastava. A new approach to computing maximum flows using electrical flows. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing - STOC '13*, page 755, New York, New York, USA, 2013. ACM Press. 90, 95

[95] Kevin Leyton-Brown and Yoav Shoham. Essentials of game theory: A concise multidisciplinary introduction. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2(1):1–88, 2008. 22, 47

[96] Elliott H. Lieb. Convex trace functions and the wigner-yanase-dyson conjecture. *Advances in Mathematics*, 11(3):267–288, 1973. 213

[97] Elliott H. Lieb and Walter E. Thirring. Inequalities for the Moments of the Eigenvalues of the Schrödinger Hamiltonian and Their Relation to Sobolev Inequalities. *Studies in Mathematical Physics*, pages 269–303, 1976. 191

[98] Peter B Linhart and Roy Radner. Minimax-regret strategies for bargaining over several variables. *Journal of Economic Theory*, 48(1):152–178, 1989. 45, 85

[99] Giuseppe Lopomo, Luca Rigotti, and Chris Shannon. Uncertainty in mechanism design. Technical report, 2009. 17, 37, 44

[100] Giuseppe Lopomo, Luca Rigotti, and Chris Shannon. Knightian uncertainty and moral hazard. *Journal of Economic Theory*, 146(3):1148 – 1172, 2011. Incompleteness and Uncertainty in Economics. 17

[101] Michael Luby and Noam Nisan. A parallel approximation algorithm for positive linear programming. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing - STOC '93*, pages 448–457, New York, New York, USA, 1993. ACM Press. 97, 115, 116, 117, 118, 119, 133, 143, 144, 145, 146, 186, 187, 188

[102] Fabio Maccheroni, Massimo Marinacci, and Aldo Rustichini. Ambiguity aversion, robustness, and the variational representation of preferences. *Econometrica*, 74(6):1447–1498, 2006. 17

[103] Aleksander Madry. Faster approximation schemes for fractional multicommodity flow problems via dynamic graph algorithms. In *Proceedings of the 42nd ACM symposium on Theory of computing - STOC '10*, page 121, New York, New York, USA, 2010. ACM Press. 116, 144

[104] Aleksander Madry. Navigating Central Path with Electrical Flows: From Flows to Matchings, and Back. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 253–262. IEEE, October 2013. 90

[105] Andrew Mas-Colell. An equilibrium existence theorem without complete or transitive preferences. *Journal of Mathematical Economics*, 1(3):237–246, December 1974. 17, 44

[106] H. Brendan McMahan. A Unified View of Regularized Dual Averaging and Mirror Descent with Implicit Updates. *arXiv preprint arXiv:1009.3240*, September 2011. Previously appeared in AISTATS 2011 as a conference paper entitled "Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization". 93

[107] H. Brendan McMahan and Matthew Streeter. Adaptive Bound Optimization for Online Convex Optimization. In *Proceedings of the 23rd Annual Conference on Learning Theory - COLT '10*, February 2010. 93

[108] Paul Milgrom. Auctions and bidding: A primer. *Journal of Economic Perspectives*, 3(3):3–22, Summer 1989. 44

[109] John W. Milnor. Games against nature. In Robert M Thrall, Clyde Hamilton Coombs, and Robert L Davis, editors, *Decision processes*. John Wiley & Sons, Inc., 1954. 45, 82

[110] Roger B. Myerson. Incentive compatibility and the bargaining problem. *Econometrica: journal of the Econometric Society*, pages 61–73, 1979. 26

[111] Assaf Naor. Sparse quadratic forms and their geometric applications [after Batson, Spielman and Srivastava]. *Astérisque*, 2012. 209

[112] Leandro Nascimento. Remarks on the consumer problem under incomplete preferences. *Theory and Decision*, 70(1):95–110, January 2011. 17, 44

[113] Arkadi Nemirovski. Prox-Method with Rate of Convergence $O(1/t)$ for Variational Inequalities with Lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems. *SIAM Journal on Optimization*, 15(1):229–251, January 2004. 117, 143, 145, 149, 187

[114] Arkadi Nemirovsky and David Yudin. *Problem complexity and method efficiency in optimization.* Nauka Publishers, Moscow (in Russian), 1978. John Wiley, New York (in English) 1983. 92, 93

[115] Yu Nesterov. Rounding of convex sets and efficient gradient methods for linear programming problems. *Optimisation Methods and Software*, 23(1):109–128, 2008. 143, 145, 187

[116] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Doklady AN SSSR (translated as Soviet Mathematics Doklady)*, volume 269, pages 543–547, 1983. 89, 95, 102, 107, 147

[117] Yurii Nesterov. *Introductory Lectures on Convex Programming Volume: A Basic course*, volume I. Kluwer Academic Publishers, 2004. 89, 91, 92, 95, 97, 98, 102, 104, 105, 107, 111, 120, 147

[118] Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, December 2005. 89, 92, 95, 96, 102, 103, 104, 106, 107, 116, 117, 120, 143, 144, 145, 147, 149, 153, 187

[119] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, June 2007. 92, 93

[120] Yurii Nesterov. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008. 95

[121] Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013. 95

[122] Yurii Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, May 2014. 95

[123] Brendan O'Donoghue and Emmanuel Candès. Adaptive Restart for Accelerated Gradient Schemes. *Foundations of Computational Mathematics*, July 2013. 95, 105, 106

[124] Efe A. Ok. Utility representation of an incomplete preference relation. *Journal of Economic Theory*, 104:429–449, 2002. 17, 44

[125] Lorenzo Orecchia. *Fast Approximation Algorithms for Graph Partitioning using Spectral and Semidefinite-Programming Techniques*. PhD thesis, EECS Department, University of California, Berkeley, May 2011. 186, 208

[126] Lorenzo Orecchia, Sushant Sachdeva, and Nisheeth K. Vishnoi. Approximating the exponential, the lanczos method and an $\tilde{O}(m)$-time spectral algorithm for balanced separator. In *STOC '12*. ACM Press, November 2012. 90, 108, 186, 210, 211, 212

[127] Richard Peng. Private communication, April 2015. 187, 190

[128] Richard Peng and Daniel A. Spielman. An efficient parallel solver for SDD linear systems. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 333–342, 2014. 207

[129] Richard Peng and Kanat Tangwongsan. Faster and simpler width-independent parallel algorithms for positive semidefinite programming. In *Proceedinbgs of the 24th ACM symposium on Parallelism in algorithms and architectures - SPAA '12*, page 101, New York, New York, USA, January 2012. ACM Press. 185, 186, 187, 188, 190, 193, 195

[130] Richard Peng and Kanat Tangwongsan. Faster and simpler width-independent parallel algorithms for positive semidefinite programming. *ArXiv e-prints*, abs/1201.5135v2, August 2014. `http://arxiv.org/abs/1201.5135v2`. 187, 190

[131] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast Approximation Algorithms for Fractional Packing and Covering Problems. *Mathematics of Operations Research*, 20(2):257–301, May 1995. 90, 108, 116, 117, 135, 143, 144, 145, 187, 212

[132] Ryan Porter, Amir Ronen, Yoav Shoham, and Moshe Tennenholtz. Fault tolerant mechanism design. *Artificial Intelligence*, 172:1783–1799, October 2008. 44

[133] Alexander Rakhlin. Lecture notes on online learning. *Draft*, 2009. Available at `http://www-stat.wharton.upenn.edu/~rakhlin/courses/stat991/papers/lecture_notes.pdf`. 214

[134] Ludovic Renou and Karl H. Schlag. Minimax regret and strategic uncertainty. *Journal of Economic Theory*, 145(1):264–286, January 2010. 45, 83, 86

[135] Luca Rigotti and Chris Shannon. Uncertainty and risk in financial markets. *Econometrica*, 73(1):203–243, 01 2005. 17, 44

[136] R. Tyrrell Rockafellar. *Convex Analysis (Princeton Landmarks in Mathematics and Physics)*. Princeton University Press, 1996. 108, 222

[137] Tuomas Sandholm. Issues in computational vickrey auctions. *International Journal of Electronic Commerce*, 4:107–129, March 2000. 44

[138] Leonard J Savage. The theory of statistical decision. *Journal of the American Statistical association*, 46(253):55–67, 1951. 45, 82

[139] David Schmeidler. Subjective probability and expected utility without additivity. *Econometrica*, 57(3):571–87, May 1989. 17, 44

[140] Reinhard Selten. Blame avoidance as motivating force in the first price sealed bid private value auction. In *Economics Essays in Honor of Werner Hildenbrand*, pages 333–344. Springer, 1989. 45, 85

[141] Wayne Shafer and Hugo Sonnenschein. Equilibrium in abstract economies without ordered preferences. *Journal of Mathematical Economics*, 2(3):345–348, December 1975. 17, 44

[142] Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, Hebrew University, 2007. 210

[143] Shai Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011. 99, 109, 214

[144] Shai Shalev-Shwartz and Yoram Singer. Logarithmic regret algorithms for strongly convex repeated games. Technical report, 2007. 105

[145] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for l1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011. 147

[146] Shai Shalev-Shwartz and Tong Zhang. Accelerated Mini-Batch Stochastic Dual Coordinate Ascent. In *NIPS*, pages 1–17, May 2013. 95

[147] Shai Shalev-Shwartz and Tong Zhang. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. *arXiv preprint arXiv:1309.2375*, pages 1–38, September 2013. 95

[148] Ohad Shamir and Tong Zhang. Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes. In *Proceedings of the 30th International Conference on Machine Learning - ICML '13*, volume 28, 2013. 94

[149] Jonah Sherman. Breaking the multicommodity flow barrier for $o(\sqrt{\log n})$-approximations to sparsest cut. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 363–372, 2009. 210

[150] Jonah Sherman. Nearly Maximum Flows in Nearly Linear Time. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 263–269. IEEE, October 2013. 90

[151] Daniel A. Spielman and Nikhil Srivastava. Graph Sparsification by Effective Resistances. *SIAM Journal on Computing*, 40(6):1913–1926, January 2011. 207, 208, 217, 220, 224, 244

[152] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing - STOC '04*, page 81, New York, New York, USA, 2004. ACM Press. 208, 220, 248

[153] Daniel A. Spielman and Shang-Hua Teng. Spectral Sparsification of Graphs. *SIAM Journal on Computing*, 40(4):981–1025, January 2011. 208

[154] David Steurer. Fast SDP algorithms for constraint satisfaction problems. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms - SODA '10*, pages 684—-697, 2010. 210

[155] Joerg Stoye. Axioms for minimax regret choice correspondences. *Journal of Economic Theory*, 146(6):2226–2251, 2011. 45, 82

[156] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014. 95, 105

[157] David R. M. Thompson and Kevin Leyton-Brown. Valuation uncertainty and imperfect introspection in second-price auctions. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1*, AAAI '07, pages 148–153, 2007. 44

[158] Luca Trevisan. Parallel Approximation Algorithms by Positive Linear Programming. *Algorithmica*, 21(1):72–88, May 1998. 116, 144

[159] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961. 18

[160] Nisheeth K. Vishnoi. personal communication, 2014. 207, 208

[161] Abraham Wald. Statistical decision functions. *The Annals of Mathematical Statistics*, 20(2):pp. 165–205, 1949. 45, 82

[162] R. M. Wilcox. Exponential operators and parameter differentiation in quantum physics. *Journal of Mathematical Physics*, 8(4):962–982, 1967. 199

[163] Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. *The Journal of Machine Learning Research*, 11:2543–2596, 2010. 92

[164] Penghui Yao. Private communication, April 2015. 187, 190

[165] Neal E. Young. Sequential and parallel algorithms for mixed packing and covering. In *42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'01)*, pages 538–546. IEEE Comput. Soc, 2001. 97, 117, 118, 119, 122, 133, 143, 145, 146, 187, 190

[166] Neal E. Young. Nearly linear-time approximation schemes for mixed packing/covering and facility-location linear programs. *ArXiv e-prints*, abs/1407.3015, July 2014. 118, 143, 145, 146, 158, 187, 190

[167] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003. 214

[168] Anastasios Zouzias. A matrix hyperbolic cosine algorithm and applications. In *Proceedings of the 39th International Colloquium Conference on Automata, Languages, and Programming - Volume Part I*, ICALP'12, pages 846–858, Berlin, Heidelberg, 2012. Springer-Verlag. 207, 208

[169] Edo Zurel and Noam Nisan. An efficient approximate allocation algorithm for combinatorial auctions. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 125–136. ACM, 2001. 116, 144