Regularized Box-Simplex Games and Dynamic Decremental Bipartite Matching

Arun Jambulapati
Stanford University, CA, USA

Yujia Jin
Stanford University, CA, USA

Aaron Sidford
Stanford University, CA, USA

Kevin Tian
Stanford University, CA, USA

Abstract

Box-simplex games are a family of bilinear minimax objectives which encapsulate graph-structured problems such as maximum flow [41], optimal transport [29], and bipartite matching [5]. We develop efficient near-linear time, high-accuracy solvers for regularized variants of these games. Beyond the immediate applications of such solvers for computing Sinkhorn distances, a prominent tool in machine learning, we show that these solvers can be used to obtain improved running times for maintaining a (fractional) ϵ-approximate maximum matching in a dynamic decremental bipartite graph against an adaptive adversary. We give a generic framework which reduces this dynamic matching problem to solving regularized graph-structured optimization problems to high accuracy. Through our reduction framework, our regularized box-simplex game solver implies a new algorithm for dynamic decremental bipartite matching in total time $\widetilde{O}(m \cdot \epsilon^{-3})$, from an initial graph with m edges and n nodes. We further show how to use recent advances in flow optimization [11] to improve our runtime to $m^{1+o(1)} \cdot \epsilon^{-2}$, thereby demonstrating the versatility of our reduction-based approach. These results improve upon the previous best runtime of $\widetilde{O}(m \cdot \epsilon^{-4})$ [6] and illustrate the utility of using regularized optimization problem solvers for designing dynamic algorithms.

2012 ACM Subject Classification Theory of computation → Dynamic graph algorithms

Keywords and phrases bipartite matching, decremental matching, dynamic algorithms, continuous optimization, box-simplex games, primal-dual method

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.77

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2204.12721

Funding *Yujia Jin*: supported by a Stanford Graduate Fellowship and the Dantzig-Lieberman Operations Research Fellowship.
Aaron Sidford: supported in part by a Microsoft Research Faculty Fellowship, NSF CAREER Award CCF-1844855, NSF Grant CCF-1955039, a PayPal research award, and a Sloan Research Fellowship.
Kevin Tian: supported in part by a Google Ph.D. Fellowship, a Simons-Berkeley VMware Research Fellowship, a Microsoft Research Faculty Fellowship, NSF CAREER Award CCF-1844855, NSF Grant CCF-1955039, and a PayPal research award.

Acknowledgements We thank anonymous reviewers for their feedback, Amin Saberi and David Wajc for helpful conversations, Jason Altschuler for providing a reference for the unaccelerated convergence rate of Sinkhorn's algorithm (in the original submission, we claimed no such rate had been stated previously), and Monika Henzinger and Thatchaphol Saranurak for helpful information regarding prior work.

1 Introduction

Efficient approximate solvers for graph-structured convex programming problems have led to a variety of recent advances in combinatorial optimization. Motivated by problems related to maximum flow and optimal transportation, a recent line of work [40, 30, 41, 42, 29, 13] developed near-linear time, accelerated solvers for a particular family of convex programming objectives we refer to in this paper as *box-simplex games*:

$$\min_{x \in \Delta^m} \max_{y \in [-1,1]^n} y^\top \mathbf{A} x + c^\top x - b^\top y \text{ where } \Delta^m := \{x \in \mathbb{R}^m_{\geq 0} | \, \|x\|_1 = 1\} . \tag{1}$$

Box-simplex games, (1), are bilinear problems where a maximization player is constrained to the box (the ℓ_∞ ball) and a minimization player is constrained to the simplex (the nonnegative ℓ_1 shell). The problem provides a convenient encapsulation of linear programming problems with ℓ_1 or ℓ_∞ structure; (1) can be used to solve box-constrained ℓ_∞ regression problems [41, 42] and maximizing over the box player yields the following ℓ_1 regression problem

$$\min_{x \in \Delta^m} c^\top x + \|\mathbf{A}x - b\|_1 . \tag{2}$$

Furthermore, solvers for (1) and (2) are used in state-of-the-art algorithms for approximate maximum flow [41], optimal transport (OT) [29], (width-dependent) positive linear programming [7], and semi-streaming bipartite matching [5].

One of the main goals of our work is to develop efficient algorithms for solving *regularized* variants of the problems (1) and (2). An example of particular interest is the following

$$\min_{x \in \Delta^m | \mathbf{B}^\top x = d} c^\top x + \mu H(x), \text{ where } \mu \geq 0 \text{ and } H(x) := \sum_{i \in [m]} x_i \log x_i . \tag{3}$$

The case of (3) when $\mathbf{B} \in \mathbb{R}^{m \times n}$ is the (unsigned) edge-vertex incidence matrix of a complete bipartite graph, and d is a pair of discrete distributions supported on the sides of the bipartition, is known as the *Sinkhorn distance* objective. This is used in machine learning [14] as an efficiently-computable approximation to optimal transport distances: c corresponds to movement costs, and d encodes the prescribed marginals. This objective has favorable properties, e.g. differentiability [44], and there has been extensive work by both theorists and practitioners to solve (3) and analyze its properties (see e.g. [14, 4] and references therein). Choosing $\mathbf{A}$ and b to be sufficiently large multiples of $\mathbf{B}^\top$ and d, it can be shown that solutions to the following regularized variant of (2) yield approximate solutions to (3),

$$\min_{x \in \Delta^m} c^\top x + \|\mathbf{A}x - b\|_1 + \mu H(x) . \tag{4}$$

Beyond connections to Sinkhorn distances, there are additional reasons why it may be desirable to solve regularized box-simplex games. For example, regularization could speed up algorithms and allow high-precision solutions to be computed more efficiently. Further, obtaining a high-precision solution to a regularized version of the problem yields a more canonical and predictable approximate solution than an arbitrary low-precision approximation to the unregularized problem. Moreover, regularization potentially makes optimal solutions more stable to input changes. For box-simplex games stemming from bipartite matching we quantify this stability and show all of these properties allow regularized solvers to yield faster algorithms for a particular dynamic matching problem.

Altogether, the main contributions of this paper are the following.

1. We give improved running times for the problem of *dynamic decremental bipartite matching* (DDBM) with an *adaptive adversary*, a fundamental problem in dynamic graph algorithms. Our algorithm follows from a general black-box reduction we develop from DDBM to solving (variants of) regularized box-simplex games to high precision.
2. We give efficient solvers for (variants of) the regularized box-simplex problems (3), (4).
3. As a byproduct, we also show how to apply our new solvers (and additional tools from the literature) to obtain state-of-the-art methods for computing Sinkhorn distances.

Formally, the *DDBM* problem we consider is the following: given a bipartite graph undergoing edge deletions, maintain an *ϵ-approximate (maximum) matching*,[1] that is a matching which has size at least a $(1-\epsilon)$-fraction of the maximum (for a pre-specified) value of ϵ. Unless specified otherwise, we consider the *adaptive adversary model* where edge deletions can be specified adaptively to the matching returned. Further, we allow the matching output by the algorithm to be *fractional*, rather than integral.

We show how to reduce solving the DDBM problem to solving a sequence of regularized box-simplex games. This reduction yields a new approach to dynamic matching; this approach is inspired by prior work, e.g. [6], but conceptually distinct in that it decouples the solving of optimization subproblems from characterizing their solutions. For our specific DDBM problem, the only prior algorithm achieving an amortized polylogarithmic update time (for constant ϵ) is in the recent work of [6], which derives their dynamic algorithm as an application of the *congestion balancing* technique. Our reduction eschews this combinatorial tool and directly argues, via techniques from convex analysis, that solutions to appropriate regularized matching problems can be used dynamically as approximate matchings while requiring few recomputations. We emphasize our use of fast *high-accuracy solvers*[2] in the context of our reduction to obtain our improved runtimes, as our approach leverages structural characteristics of the exact solutions which we only show are inherited by approximate solutions when solved to sufficient accuracy.

Our work both serves as a proof-of-concept of the utility of regularized linear programming solvers as a subroutine in dynamic graph algorithms, and provides the tools necessary to solve said problems in various structured cases. This approach to dynamic algorithm design effectively separates a "stability analysis" of the solution to a suitable optimization problem from the computational burden of solving that problem to high accuracy: any improved solver would then have implications for faster dynamic algorithms as well. As a demonstration of this flexibility, we give three uses of our reduction framework (which proceed via different solvers) in obtaining our improved DDBM update time. We hope our work opens the door to exploring the use of the powerful continuous optimization toolkit, especially techniques originally designed for non-dynamic problems, for their dynamic counterparts.

Paper organization. We overview our contributions in Section 1.1, and related prior work in Section 1.2. We state preliminaries in Section 2. In Section 3.1, we describe our framework for reducing DDBM to a sequence of regularized optimization problems satisfying certain properties, and in Section 3.2 we give three different instantiations of the framework, obtaining a variety of DDBM solvers. Finally in Section 4 we provide our main algorithm for regularized box-simplex games. In the full version we provide additional discussions on a recent advancement for faster DDBM solvers, proofs for Section 3 and Section 4, and additional results for approximating Sinkhorn distances efficiently.

[1] This is sometimes also referred to as a $(1+\epsilon)$-multiplicatively approximate matching in the literature.
[2] Throughout, we typically use the term "high-accuracy" to refer to an algorithm whose runtime scales polylogarithmically in the inverse accuracy (as opposed to e.g. polynomially).

1.1 Our results

A framework for faster DDBM. We develop a new framework for solving the DDBM problem of computing an ϵ-approximate maximum matching in a dynamic graph undergoing edge deletions from an adaptive adversary. Our framework provides a reduction from this DDBM problem to solving various regularized formulations of box-simplex games.

To illustrate the reduction, suppose we have bipartite $G = (V, E)$ and, for simplicity, that we know M^*, the size of the (maximum cardinality) matching. As demonstrated in [5], solving the ℓ_1 regression problem $\min_{x \in M^* \Delta^m} -c^\top x + \|\mathbf{A}x - b\|_1$, to ϵM^* additive accuracy for appropriate choices of $\mathbf{A}$, b, and c yields an ϵ-approximate maximum cardinality matching. Intuitively, $\mathbf{A}$ and b penalize violations of the matching constraints, and c is a multiple of the all-ones vector capturing the objective of maximizing the matching size. However, ℓ_1 regression objectives do not necessarily have unique minimizers: as such the output of directly minimizing these objectives is difficult to characterize beyond (approximate) optimality. This induces difficulty in using solutions to such problems directly in dynamic graph algorithms.

Our first key observation (building upon intuition from congestion balancing [6]) is that, beyond enabling faster runtime guarantees, regularization provides more robust solutions which are resilient to edge deletions in dynamic applications. We show that if

$$x^*_\epsilon := \min_{x \in M^* \cdot \Delta^m} -c^\top x + \|\mathbf{A}x - b\|_1 + \epsilon H(x) \tag{5}$$

is the solution to the *regularized* box-simplex formulation of bipartite matching, then x^*_ϵ enjoys favorable stability properties allowing us to argue about its size under deletions.

The stability of solutions to (5) is fairly intuitive: the entropy regularizer encourages the objective to spread the matching uniformly, when all else is held equal. For example, when G is a complete bipartite graph on $2n$ vertices, standard linear programming relaxations of matching do not favor either of (i) an integral perfect matching, and (ii) a fractional matching spreading mass evenly across many edges, over the other. However, using (i) as our approximate matching on a graph undergoing deletions is substantially more unstable; an adaptive adversary can remove edges corresponding to our matching, forcing $\Omega(n)$ recomputations. On the other hand, no deletions can cause this type of instability for strategy (ii): as each edge receives weight $\frac{1}{n}$ in the fractional matching, the only way to reduce the fractional matching size by ϵn is to remove $O(\epsilon n^2)$ edges: thus $O(\epsilon^{-1})$ recomputations intuitively suffice for maintaining an ϵ-approximate matching. This distinction underlies the use of *high-accuracy* solvers in our reduction; indeed, while they obtain large matching values in an original graph, approximate solutions may not carry the same types of dynamic matching value stability. We note similar intuition motivated the approach in [6].

To make this argument more rigorous, consider using x^*_ϵ as our approximate matching for a number of iterations corresponding to edge deletions, until its size restricted to the smaller graph has decreased by a factor of $1 - O(\epsilon)$. By using strong convexity of (5) in the ℓ_1 norm, we argue that whenever the objective value of x^*_ϵ has worsened, the maximum matching size itself must have gone down by a (potentially much smaller) amount. A tighter characterization of this strong convexity argument shows that we only need to recompute a solution to slight variants of (5) roughly $\widetilde{O}(\epsilon^{-2})$ times throughout the life of the algorithm. Combined with accelerated $\widetilde{O}(\frac{m}{\epsilon})$-time solvers for regularized box-simplex games (which are slight modifications of (5)), this strategy yields an overall runtime of $\widetilde{O}(\frac{m}{\epsilon^3})$, improving upon the recent state-of-the-art decremental result of [6].

We formalize these ideas in Section 3, where we demonstrate that a range of regularization strategies (see Definition 5) such as (5) are amenable to this reduction. Roughly, as long as our regularized objective is "at least as strongly convex" as the entropic regularizer, and closely approximates the matching value in the static setting, then it can be used in our DDBM algorithm. Combining this framework with solvers for regularized matching problems, we give three different results. The first two obtain amortized update times of roughly $\widetilde{O}(\epsilon^{-3})$, in Theorems 9 and 10 via box-simplex games and matrix scaling, respectively (though the latter holds only for dense graphs). We give an informal statement of the former here.

▶ **Theorem 1** (informal, see Theorem 9)**.** *Let $G = (V, E)$ be bipartite, $|V| = n$, $|E| = m$, and $\epsilon \geq \mathrm{poly}(m^{-1})$. There is a deterministic algorithm maintaining an ϵ-approximate matching in a dynamic bipartite graph with adversarial edge deletions running in time $O(m \log^5 m \cdot \epsilon^{-3})$.*

Notably, our algorithm (deterministically) returns a *fractional matching*. There is a black-box reduction from dynamic integral matching maintenance to dynamic fractional matching maintenance contained in [45], but this reduction is bottlenecked at an amortized $\widetilde{O}(\epsilon^{-4})$ runtime (see e.g. Appendix A.2, [6]). Improving this reduction is a key open problem.

High-accuracy solvers for regularized box-simplex games. To use our DDBM framework, we give a new algorithm for solving regularized box-simplex games of the form:

$$\min_{x \in \Delta^m} \max_{y \in [0,1]^n} f_{\mu,\epsilon}(x, y) := y^\top \mathbf{A}^\top x + c^\top x - b^\top y + \mu H(x) - \frac{\epsilon}{2} \left(y^2\right)^\top |\mathbf{A}|^\top x, \tag{6}$$

where ϵ and $\mu = \Omega(\epsilon)$ are regularization parameters and y^2, $|\mathbf{A}|$ are entrywise. The terms $H(x)$ and $(y^2)^\top |\mathbf{A}|^\top x$ in (6) are parts of a primal-dual regularizer proposed in [29] (and a variation of a similar regularizer of [41]) used in state-of-the-art algorithms for approximately solving (unregularized) box-simplex games. This choice of regularization enjoys favorable properties over the joint box-simplex domain, and sidesteps the infamous ℓ_∞-strong convexity barrier that has limited previous attempts at acceleration for this problem. Under relatively mild restrictions on problem parameters (see discussion at the start of Section 4), we develop a *high accuracy solver* for (6), stated informally here.

▶ **Theorem 2** (informal, see Theorem 25)**.** *Given an instance of (6), with $\mu = \Omega(\epsilon)$, $\|\mathbf{A}\|_\infty \leq 1$, and $\sigma \geq \mathrm{poly}(m^{-1})$ Algorithm 4 returns x with $\max_{y \in [0,1]^n} f_{\mu,\epsilon}(x, y) - f_{\mu,\epsilon}(x^\star, y^\star) \leq \sigma$ in time $\widetilde{O}(\mathrm{nnz}(\mathbf{A}) \cdot \frac{1}{\sqrt{\mu\epsilon}})$ where $(x^\star, y^\star)$ is the optimizer of (6).*

Our solver follows recent developments in solving unregularized box-simplex games. We analyze an approximate extragradient algorithm based on the mirror prox method of [37], and prove that iterates of the regularized problem (6) enjoy multiplicative stability properties previously shown for the iterates of mirror prox on the unregularized problem [13]. Leveraging these tools, we also show the regularizer-operator pair satisfies technical conditions known as *relative Lipschitzness* and *strong monotonicity*, thus enabling a similar convergence analysis as in [13]. This yields an efficient algorithm for solving (6).

Roughly, when the scale of the problem (defined in terms of the matrix operator norm $\|\mathbf{A}\|_\infty$ and appropriate norms of b and c) is bounded,[3] our algorithm for computing a high-precision optimizer to (6) runs in $\widetilde{O}(\frac{1}{\sqrt{\mu\epsilon}})$ iterations, each bottlenecked by a matrix-vector product through $\mathbf{A}$. When $\mu \approx \epsilon$, the optimizer of the regularized variant is an $O(\epsilon)$-approximate solution to the unregularized problem (1), and hence Theorem 25 recovers state-of-the-art runtimes (scaling as $\widetilde{O}(\epsilon^{-1})$) for box-simplex games up to logarithmic factors. We achieve our improved dependence on μ in Theorem 25 by trading off the scales of the

[3] Our runtimes straightforwardly extend to depend appropriately on these norms in a scale-invariant way.

primal and dual domains. This type of argument is well-known for *separable regularizers* [9], but a key technical novelty of our paper is demonstrating a similar analysis holds for non-separable regularizers compatible with box-simplex games e.g. the one from [29], which has not previously been done. To our knowledge, Theorem 25 is the first result for solving general regularized box-simplex games to high accuracy in nearly-linear time. We develop our box-simplex algorithm and prove Theorem 25 in Section 4.

Improved rates for the Sinkhorn distance objective. We apply our accelerated solver for (6) in computing approximations to the Sinkhorn distance objective (3), a fundamental algorithmic problem in the practice of machine learning, at a faster rate. It is well-known that solving the *regularized* Sinkhorn problem (3) with μ scaling much larger than the target accuracy ϵ enjoys favorable properties in practice [14] (compared to its unregularized counterpart, the standard OT distance). In [3], the authors show that Sinkhorn iteration studied in prior work solves (3) to additive accuracy ϵ at an unaccelerated rate of $\widetilde{O}(\frac{1}{\mu\epsilon})$. For completeness we provide a proof of this result (up to logarithmic factors) in Appendix C.3 in the full version of this paper.

As a straightforward application of the solver we develop for (6), we demonstrate that we can attain an accelerated rate of $\widetilde{O}(\frac{1}{\sqrt{\mu\epsilon}})$ for approximating (3) to additive accuracy ϵ via a first-order method. More specifically, the following result is based on reducing the "explicitly constrained" Sinkhorn objective (3) to a "soft constrained" regression variant of the form (4), where our box-simplex game solver is applicable. We now state our first result on improved rates for approximating Sinkhorn distance objectives.

▶ **Theorem 3** (informal, see Theorem 11 in full version)**.** *Let* $\mu \in [\Omega(\epsilon), O(\frac{\|c\|_\infty}{\log m})]$ *in* (3) *corresponding to a complete bipartite graph with* m *edges. There is an algorithm based on the regularized box-simplex game solver of Theorem 25 which obtains an* ϵ*-approximate minimizer to* (3) *in time* $\widetilde{O}(m \cdot \frac{\|c\|_\infty}{\sqrt{\mu\epsilon}})$.

By leveraging the particular structure of the Sinkhorn distance and its connection to a primitive in scientific computing and theoretical computer science known as *matrix scaling* [35, 12, 2], we give a further-improved solver for (3) in Theorem 4. This solver has a nearly-linear runtime scaling as $\widetilde{O}(\frac{1}{\mu})$, which is a high-precision solver for the original Sinkhorn objective. Our high-precision Sinkhorn solver applies powerful second-order optimization tools from [12] based on the *box-constrained Newton's method* for matrix scaling, yielding our second result on improved Sinkhorn distance approximation rates.

▶ **Theorem 4** (informal, see Theorem 12 in full version)**.** *Let* $\mu, \epsilon = O(\|c\|_\infty)$ *in* (3) *corresponding to a complete bipartite graph with* m *edges. There is an algorithm based on the matrix scaling solver of [12] which obtains an* ϵ*-approximate minimizer to* (3) *in time* $\widetilde{O}(m \cdot \frac{\|c\|_\infty}{\mu})$.

We present both Theorems 3 and 4 because they follow from somewhat incomparable solver frameworks. While the runtime of Theorem 3 is dominated by that of Theorem 4, it is a direct application of a more general solver (Theorem 25), which also applies to regularized regression or box-simplex objectives where the optimum does not have a characterization as a matrix scaling. Moreover, the algorithm of Theorem 4 is a second-order method which leverages recent advances in solving Laplacian systems, and hence may be less practical than its counterpart in Theorem 3. Finally, we note that due to subtle parameterization differences for our DDBM applications, the DDBM runtime attained by using our box-simplex solver within our reduction framework is more favorable on sparse graphs ($m \ll n^2$), compared to that obtained by the matrix scaling solver.

1.2 Prior work

Dynamic matching. Dynamic graph algorithms are an active area of research in the theoretical computer science, see e.g. [27, 15, 6, 31, 17, 21, 1, 26, 19, 22, 36, 20] and references therein. These algorithms have been developed under various dynamic graph models, including the additions and deletions on *vertices* or *edges*, and *oblivious adversary* model where the updates to the graph are fixed in advance (i.e. do not depend on randomness used by the algorithm), and the *adaptive adversary* model in which updates are allowed to respond to the algorithm, potentially adversarially. We focus on surveying deterministic dynamic matching algorithms with edge streams, which perform equally well under oblivious and adaptive updates; we remark the dynamic matching algorithms have also been studied under vertex addition and deletion model in [8]. For a more in-depth discussion and corresponding developments in other settings, see [45].

Many variants of the particular dynamic problem of maintaining matchings in bipartite graphs have been studied, such as the *fully dynamic* [25], *incremental* [24, 23], and *decremental* [6] cases. However, known conditional hardness results [28, 32] suggest that attaining a polylogarithmic update time for maintaining an exact fully dynamic matching may be unattainable, prompting the study of restricted variants which require maintaining an approximate matching. The works most relevant to our paper are those of [24], which provides a $\widetilde{O}(\epsilon^{-4})$ amortized update time algorithm for computing an ϵ-approximate matching for incremental bipartite matching, and [6], which achieves a similar $\widetilde{O}(\epsilon^{-4})$ update time for decremental bipartite matching. Our main DDBM results, stated in Theorems 9 and 10, improve upon [6] by roughly a factor of ϵ^{-1} in the decremental setting.

Box-simplex games. Box-simplex games, as well as ℓ_1 and ℓ_∞ regression, are equivalent to linear programs in full generality [33], have widespread utility, and hence have been studied extensively by the continuous optimization community. Here we focus on discussing *near-linear time* approximation algorithms, i.e. algorithms which run in time near-linear in the sparsity of the constraint matrix, potentially depending inverse polynomially on the desired accuracy. Interior point methods solve these problems with polylogarithmic dependence on accuracy, but are second-order and often encounter polynomial runtime overhead in the dimension (though there are exceptions, e.g. [43] and references therein).

A sequence of early works e.g. [37, 38, 39] on primal-dual optimization developed first-order methods for solving games of the form (1). These works either directly operated on the objective (1) as a minimax problem, or optimized a smooth approximation to the objective recast as a convex optimization problem. While these techniques obtained iteration complexities near-linear in the sparsity of the constraint matrix $\mathbf{A}$, they either incurred an (unaccelerated) ϵ^{-2} dependence on the accuracy ϵ, or achieved an ϵ^{-1} rate of convergence at the cost of additional dimension-dependent factors. This was due to the notorious "ℓ_∞ strong convexity barrier" (see Appendix A, [42]), which bottlenecked classical acceleration analyses over an ℓ_∞-constrained domain. [41] overcame this barrier by utilizing the primal-dual structure of (1) through a technique called "area convexity", obtaining a $\widetilde{O}(\epsilon^{-1})$-iteration algorithm. Since then, [13] demonstrated that fine-grained analyses of the classical algorithms of [37, 39] also obtain comparable rates for solving (1). Finally, we mention that area convexity has found applications in optimal transport and positive linear programming [29, 7].

Sinkhorn distances. Since [14] proposed Sinkhorn distances for machine learning applications, a flurry of work has aimed at developing algorithms with faster runtimes for (3). A line of work by [4, 16, 34] has analyzed the theoretical guarantees of the classical Sinkhorn matrix scaling algorithm for this problem, due to the characterization of its solution as a diagonal rescaling of a fixed matrix. These algorithms obtain rates scaling roughly as $\widetilde{O}(\|c\|_\infty^2 \epsilon^{-2})$ for solving (3) to additive accuracy ϵ. Perhaps surprisingly, to our knowledge no guarantees for solving (3) which improve as the regularization parameter μ grows are currently stated in the literature, a shortcoming addressed by this work. Finally, we remark that Sinkhorn iteration has also received extensive treatment from the theoretical computer science community, e.g. [35], due to connections with algebraic complexity; see [18] for a recent overview of these connections.

2 Preliminaries

General notation. We denote $[n] := \{1, 2, \ldots, n\}$ and let $\mathbf{0}$ and $\mathbf{1}$ denote the all-0 and all-1 vectors. Given $v \in \mathbb{R}^d$, v_i or $[v]_i$ denotes the i^{th} entry of v, and for any subset $E \subseteq [d]$ we use v_E or $[v]_E$ to denote the vector in $\mathbb{R}^d$ zeroing out v on entries outside of E. We use $([v]_i)_+ = \max([v]_i, 0)$ to denote the operation truncating negative entries. We use $v \circ w$ to denote elementwise multiplication between any $v, w \in \mathbb{R}^d$. Given matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we use $\mathbf{A}_{ij}$ to denote its $(i,j)^{\text{th}}$ entry, and denote its i^{th} row and j^{th} column by $\mathbf{A}_{i:}$ and $\mathbf{A}_{:j}$ respectively; its nonzero entry count is $\text{nnz}(\mathbf{A})$. We use $\mathbf{diag}\,(v)$ to denote the diagonal matrix where $[\mathbf{diag}\,(v)]_{ii} = v_i$, for each i. Given two quantities M and M', for any $c > 1$ we say M is a c-approximation to M' if it satisfies $\frac{1}{c}M' \le M \le cM'$. For $\epsilon \ll 1$, we say M is an ϵ(-multiplicative)-approximation of M' if $(1-\epsilon)M' \le M \le (1+\epsilon)M'$. Throughout the paper, we use $|\mathbf{A}|$ to denote taking the elementwise absolute value of a matrix $\mathbf{A}$, and v^2 to denote the elementwise squaring of a vector v when clear from context.

Norms. $\|\cdot\|_p$ denotes the ℓ_p norm of a vector or corresponding operator norm of a matrix. In particular, $\|\mathbf{A}\|_\infty = \max_i \|\mathbf{A}_{i:}\|_1$. We use $\|\cdot\|$ interchangeably with $\|\cdot\|_2$. We use Δ^m to denote an m-dimensional simplex, i.e. $x \in \Delta^m \iff x \in \mathbb{R}^d_{\ge 0}, \|x\|_1 = 1$.

Graphs. A graph $G = (V, E)$ has vertices V and edges E; we abbreviate $n := |V|$ and $m := |E|$ whenever the graph is clear from context. For bipartite graphs, $V = L \cup R$ denotes the bipartition. We let $\mathbf{B} \in \{0,1\}^{E \times V}$ be the (unsigned edge-vertex) incidence matrix with $\mathbf{B}_{ev} = 1$ if v is an endpoint of e and $\mathbf{B}_{ev} = 0$ otherwise.

Bregman divergence. Given any convex distance generating function (DGF) $q(x)$, we use $V^q_{x'}(x) = q(x) - q(x') - \langle \nabla q(x'), x - x' \rangle \ge 0$ as its induced Bregman divergence. When the DGF is clear from context, we abbreviate $V := V^q$. By definition, V satisfies

$$\langle -\nabla V_{x'}(x), x - u \rangle = V_{x'}(u) - V_x(u) - V_{x'}(x) \text{ for any } x, x', u. \tag{7}$$

Computational model. We use the standard word RAM model, where one can perform each basic arithmetic operations on $O(\log n)$-bit words in constant time.

3 Dynamic decremental bipartite matching

Here we provide a reduction from maintaining an approximately maximum matching in a decremental bipartite graph to solving regularized matching problems to sufficiently high precision. In Section 3.1 we give this framework and then, in Section 3.2, we provide various instantiations of our framework based on different solvers, to demonstrate its versatility.

3.1 DDBM framework

Here we provide our general framework for solving DDBM, which assumes that for bipartite $G = (V, E)$ and approximate matching value M there is a canonical regularized matching problem with properties given in Definition 5; we later provide multiple such examples. Throughout this section, $\mathrm{MCM}(E)$ denotes the size of the maximum cardinality matching on edge set E; the vertex set V is fixed throughout, so we omit it in definitions.

▶ **Definition 5** (Canonical regularized objective). *Let $G = (V, E_0)$ be a bipartite graph and $M \geq 0$ be an 8-approximation of $\mathrm{MCM}(E_0)$. For all $E \subseteq E_0$ with $\mathrm{MCM}(E) \geq \frac{M}{8}$, let $f_{M,E} : \mathbb{R}^E_{\geq 0} \to \mathbb{R}$,*

$$\nu^E := \min_{x \in \mathbb{R}^E_{\geq 0}} f_{M,E}(x) \text{ , and } x^E := \mathrm{argmin}_{x \in \mathbb{R}^E_{\geq 0}} f_{M,E}(x). \tag{8}$$

*We call the set of $\{f_{M,E}\}_{\mathrm{MCM}(E) \geq \frac{M}{8}}$ a family of (ϵ, β)-*canonical regularized objectives *(CROs) for $G(E_0)$ and M if the following four properties hold.*

1. *For all $E \subseteq E_0$ with $\mathrm{MCM}(E) \geq \frac{M}{8}$, $-\nu^E$ is an $\frac{\epsilon}{8}$-approximation of $\mathrm{MCM}(E)$.*
2. *For all $E \subseteq E_0$ with $\mathrm{MCM}(E) \geq \frac{M}{8}$, $f_{M,E}$ is equivalent to f_{M,E_0} with the extra constraint that $x_{E_0 \setminus E}$ is fixed to 0 entrywise.*
3. *For any $E' \subseteq E \subseteq E_0$ with $\mathrm{MCM}(E) \geq \frac{M}{8}$ and $\mathrm{MCM}(E') \geq \frac{M}{8}$,*

$$f_{M,E'}(x^{E'}) - f_{M,E}(x^E) \geq \beta V^H_{x^E}(x^{E'}) \text{ where } H(x) := \sum_e x_e \log x_e \tag{9}$$

4. *For any $x \in \mathbb{R}^E_{\geq 0}$ such that $8Mx$ is a feasible matching on (V, E),*

$$8M \|x_E\|_1 - \frac{\epsilon}{128} M \leq -f_{M,E}(x) \leq 8M \|x_E\|_1 + \frac{\epsilon}{128} M. \tag{10}$$

We further define the following notion of a canonical solver for a given CRO, which solves the CRO to sufficiently high accuracy, and rounds the approximate solution to feasibility.

▶ **Definition 6** (Canonical solver). *For (ϵ, β)-CROs $\{f_{M,E}\}_{\mathrm{MCM}(E) \geq \frac{M}{8}}$, we call $\mathcal{A}$ an $(\epsilon, \mathcal{T})$-*canonical solver *if it has subroutines* Solve *and* Round *running in $O(\mathcal{T})$ time, satisfying:*

1. Solve *finds an approximate solution $\hat{x}^E$ of $f_{M,E}$ satisfying*

$$\left(1 + \frac{\epsilon}{8}\right) \nu^E \leq f_{M,E}(\hat{x}^E) \leq \left(1 - \frac{\epsilon}{8}\right) \nu^E. \tag{11a}$$

$$\left\|\hat{x}^E - x^E\right\|_1 \leq \frac{\epsilon}{1100}. \tag{11b}$$

2. Round *takes $\hat{x}^E$ and returns $\tilde{x}^E$ where $8M\tilde{x}^E$ is a feasible matching for $G(E)$, and:*

$$\left(1 + \frac{\epsilon}{8}\right) \nu^E \leq f_{M,E}(\tilde{x}^E) \leq \left(1 - \frac{\epsilon}{8}\right) \nu^E. \tag{12a}$$

$$\tilde{x}^E \leq \hat{x}^E \text{ monotonically.} \tag{12b}$$

Our DDBM framework, Algorithm 1, uses CRO solvers satisfying the approximation guarantees of Definition 6 to dynamically maintain an approximately maximum matching. We state its correctness and runtime in Proposition 7, and defer a proof to Appendix A.1 in the full version.

In the following, we let E_0 be the original graph's edge set, and $E_1, E_2, \ldots, E_K$ be the sequence of edge sets recomputed in Line 8, before termination for E_{K+1} on Line 4.

Algorithm 1 $\mathsf{DecMatching}(\epsilon, G = (V, E))$.

Input: $\epsilon \in (0, \frac{1}{8})$, graph $G = (V, E)$
Parameters: Family of CROs $\{f_{M,E}\}_{E \text{ is MP}}$, $(\epsilon, \mathcal{T})$-canonical solver $(\mathsf{Solve}, \mathsf{Round})$
1 Compute M with $\frac{1}{2}\mathrm{MCM}(E) \le M \le \mathrm{MCM}(E)$, via the greedy algorithm
2 $\hat{x}^E \leftarrow \mathsf{Solve}(f_{M,E})$
3 $\tilde{x}^E \leftarrow \mathsf{Round}(\hat{x}^E)$, $M_{\text{est}} \leftarrow M$
4 **while** $M_{\text{est}} > \frac{1}{4}M$ **do**
5 $\quad E_{\text{del}} \leftarrow \emptyset$
6 $\quad$ **while** *edge e is deleted and* $\|\tilde{x}^E_{E_{\text{del}}}\|_1 \le \frac{\epsilon}{8}\left\|\tilde{x}^E\right\|_1$ **do**
$\qquad$ ▷ recompute whenever the deleted approximate matching size reaches a factor $\Theta(\epsilon)$
7 $\qquad E_{\text{del}} \leftarrow E_{\text{del}} \cup \{e\}$
8 $\quad E \leftarrow E \setminus E_{\text{del}}$, $E_{\text{del}} \leftarrow \emptyset$
9 $\quad \hat{x}^E \leftarrow \mathsf{Solve}(f_{M,E})$ ▷ find high-accuracy minimizer of $F_{M,E}$ satisfying (11a) and (11b)
10 $\quad \tilde{x}^E \leftarrow \mathsf{Round}(\hat{x}^E)$ ▷ round to feasible matching satisfying (12a) and (12b)
11 $\quad$ Compute M_{est} with $\frac{1}{2}\mathrm{MCM}(E) \le M_{\text{est}} \le \mathrm{MCM}(E)$, via the greedy algorithm

▶ **Proposition 7.** *Let $\epsilon \in (0, 1)$ and $M \ge 0$. Given a family of (ϵ, β)-CROs $\{f_{M,E}\}$ for $G = (V, E_0)$, and an $(\epsilon, \mathcal{T})$-canonical solver for the family, Algorithm 1 satisfies the following.*

1. *When $M_{\text{est}} > \frac{1}{4}M$ on Line 4, where M_{est} estimates $\mathrm{MCM}(E_k)$: at any point in the loop of Lines 6 to 7, $8M\tilde{x}^{E_k}_{E_k \setminus E_{\text{del}}}$ is an ϵ-approximate matching of $G(V, E_k \setminus E_{\text{del}})$.*
2. *When $M_{\text{est}} \le \frac{1}{4}M$ on Line 4, where M_{est} estimates $\mathrm{MCM}(E)$: $\mathrm{MCM}(E) \le \frac{1}{2}\mathrm{MCM}(E_0)$.*

The runtime of the algorithm is $O(m + (\mathcal{T} + m) \cdot \frac{M}{\beta\epsilon})$.

Proof sketch. We summarize proofs of the two properties, and our overall runtime bound.

Matching approximation properties. By the greedy matching guarantee in Line 4, it holds that for any E_k (the edge set recomputed in the k^{th} iteration of Line 8 before termination), its true matching size $\mathrm{MCM}(E_k)$ must be no smaller than $\frac{M}{4}$. Consequently, we can use the CRO family to approximate the true matching size up to $O(\epsilon)$ multiplicative factors, and by the guarantee (12a), this implies $8M\tilde{x}^{E_k}_{E_k \setminus E_{\text{del}}}$ is an $O(\epsilon)$ approximation of the true matching size. Also, our algorithm's termination condition and the guarantee on M_{est} immediately imply $\mathrm{MCM}(E_{K+1}) \le \frac{1}{2}\mathrm{MCM}(E_0)$.

Iteration bound. We use a potential argument. Given $E_{k+1} \subset E_k$, corresponding to consecutive edge sets requiring recomputation, we use the following inequalities:

$$f_{M,E_{k+1}}\left(x^{E_{k+1}}\right) - f_{M,E_k}\left(x^{E_k}\right) \overset{(i)}{\ge} \beta V^H_{x^{E_k}}\left(x^{E_{k+1}}\right) \tag{13}$$
$$\overset{(ii)}{\ge} \beta \sum_{i \in E_{\text{del}}} \left([x^{E_{k+1}}]_i \log[x^{E_{k+1}}]_i - [x^{E_k}]_i \log[x^{E_k}]_i - \left(1 + \log[x^{E_k}]_i\right) \cdot \left([x^{E_{k+1}}]_i - [x^{E_k}]_i\right)\right)$$
$$\overset{(iii)}{=} \beta \sum_{i \in E_{\text{del}}} [x^{E_k}]_i \overset{(iv)}{\ge} \beta\left(\left\|\hat{x}^{E_k}_{E_{\text{del}}}\right\|_1 - \left\|\hat{x}^{E_k} - x^{E_k}\right\|_1\right) \overset{(v)}{\ge} \beta\left(\left\|\tilde{x}^{E_k}_{E_{\text{del}}}\right\|_1 - \left\|\hat{x}^{E_k} - x^{E_k}\right\|_1\right),$$

where (i) uses the third property in (9), (ii) uses convexity of the scalar function $c \log c$, (iii) uses that $x^{E_{k+1}}_{E_{\text{del}}}$ is 0 entrywise, (iv) uses the triangle inequality, and (v) uses the monotonicity property (12b). Moreover, between recomputations we have that the ℓ_1-norm of deleted edges satisfies $\left\|\tilde{x}^{E_k}_{E_{\text{del}}}\right\|_1 = \Omega(\epsilon)$, and our solver guarantees $\left\|\hat{x}^{E_k} - x^{E_k}\right\|_1 = O(\epsilon)$. Since the overall function decrease before termination is $O(M)$ given the stopping criterion in Line 4, the algorithm terminates after $O(\frac{M}{\beta\epsilon})$ recomputations. ◀

Using this generic DDBM framework, we obtain improved decremental matching algorithms by defining families of CROs $f_{M,E(G)}$ with associated $(\epsilon, \mathcal{T})$-canonical solvers satisfying Definition 6. In Appendix A.2, we give a regularized primal-dual construction of $f_{M,E}$, and adapt the solver of Section 4 to develop a canonical solver for the family (specifically, as the subroutine Solve). Similarly, in Appendix A.3, we show how to construct an appropriate family of $f_{M,E}$ using Sinkhorn distances, and apply the matrix scaling method presented in Appendix C.2 (based on work of [12]) to appropriately instantiate Solve.

While both algorithms, as stated, only maintain an approximate fractional matchings, this fractional matching can be rounded at any point to an explicit integral matching via e.g. the cycle-canceling procedure of Proposition 3 in [5] in time $O(m \log m)$, or dynamically (albeit at amoritized cost $\widetilde{O}(\epsilon^{-4})$ using [45]). Moreover, our algorithm based on the regularized box-simplex solver (Theorem 9) is deterministic, and both work against an adaptive adversary. Repeatedly applying Proposition 7, we obtain the following overall claim.

▶ **Corollary 8.** *Let $G = (V, E(G))$ be bipartite, and suppose for any subgraph $(V, E_0 \subseteq E(G))$, we are given a family of (ϵ, β)-CROs and an $(\epsilon, \mathcal{T})$-canonical solver for the family. There is a deterministic algorithm maintaining a fractional ϵ-approximate matching in a dynamic bipartite graph with adversarial edge deletions, running in time $O\left(m \log^3 n + (\mathcal{T} + m) \cdot \frac{M}{\beta\epsilon} \cdot \log n\right)$.*

Proof. It suffices to repeatedly apply Proposition 7 until we can safely conclude $\mathrm{MCM}(E) = 0$, which by the second property can only happen $O(\log n)$ times. ◀

3.2 DDBM solvers

In this section, we demonstrate the versatility of the DDBM framework in Section 3.1 by instantiating it with different classes of CRO families, and applying different canonical solvers on these families. By using regularized box-simplex game solvers developed in this paper (see Section 4), we give an $\widetilde{O}(m\epsilon^{-3})$ time algorithm for maintaining a ϵ-multiplicatively approximate fractional maximum matching in a m-edge bipartite graph undergoing a sequence of edge deletions, improving upon the previous best running time of $\widetilde{O}(m\epsilon^{-4})$ [6]. We also use our framework to obtain different decremental matching algorithms with runtime $\widetilde{O}(n^2\epsilon^{-3})$ and $O(m^{1+o(1)}\epsilon^{-2})$, building on recent algorithmic developments in the literature on matrix scaling. The former method uses box-constrained Newton's method solvers for matrix scaling problems in [12] (these ideas are also used in Appendix C.2), and the latter uses a recent almost-linear time high-accuracy Sinkhorn-objective solver in [11], a byproduct of their breakthrough maximum flow solver. We defer readers to corresponding sections in Appendix A for omitted proofs.

Given a bipartite graph initialized at $G = (V, E_0)$ with unsigned incidence matrix $\mathbf{B} \in \{0,1\}^{E \times V}$; we denote $n := |V|$ and $m := |E_0|$. The first family of CROs one can consider is the regularized box-simplex game objective in form:

$$\min_{(x,\xi)\in\Delta^{E+1}} \max_{y\in[0,1]^V} f_{M,E}(x,\xi,y) := -\mathbf{1}_E^\top(8Mx) - y^\top\left(8M\mathbf{B}^\top x - \mathbf{1}\right) + \gamma^{\mathsf{x}} H(x,\xi) + \gamma^{\mathsf{y}}\left(y^2\right)^\top \mathbf{B}^\top x,$$
$$\text{where } \gamma^{\mathsf{x}} = \widetilde{\Theta}\left(\epsilon M\right),\ \gamma^{\mathsf{y}} = \Theta\left(\epsilon M\right), \text{ and}$$
$$f_{M,E}(x) := \min_{\xi|(x,\xi)\in\Delta^{E+1}} \max_{y\in[0,1]^V} f_{M,E}(x,\xi,y). \quad (14)$$

We prove this is a family of $(\epsilon, \gamma^{\mathsf{x}})$-CROs (see Lemma 8 in full version). The canonical solver for this family uses RemoveOverflow (Algorithm 4, [5]) as Round and uses the regularized box-simplex games developed later in this paper (see Section 4) as Solve, which finds an ϵ-approximate solution of (14) in time $\widetilde{O}(\frac{m}{\epsilon})$. Combining all these components with the DDBM framework in Corollary 8 leads to the following DDBM solver based on regularized box-simplex games.

▶ **Theorem 9.** *Let $G = (V, E)$ be bipartite and let $\epsilon \in [\Omega(m^{-3}), 1)$. There is a deterministic algorithm for the DDBM problem which maintains an ϵ-approximate matching, based on solving regularized box-simplex games, running in time $O\left(m\epsilon^{-3}\log^5 n\right)$.*

Our second CRO family is the following regularized Sinkhorn distance objective:

$$\min_{\binom{x}{x^{\mathsf{dum}}}\in\mathbb{R}^{\widetilde{E}}_{\geq 0}\ \mid\ 2|R|\widetilde{\mathbf{B}}^\top\binom{x}{x^{\mathsf{dum}}}=d} f^{\mathsf{sink}}_{M,E}(x^{\mathsf{tot}}) := 2|R|\mathbf{1}_E^\top x + \gamma H\left(x, x^{\mathsf{dum}}\right) \text{ where } \gamma = \widetilde{\Theta}\left(\epsilon M\right), \\ f^{\mathsf{sink}}_{M,E}(x) := \min_{x^{\mathsf{dum}}\in\mathbb{R}^{E\setminus E_0}_{\geq 0}} f^{\mathsf{sink}}_{M,E}(x, x^{\mathsf{dum}}), \tag{15}$$

where we extend graph $G = (V, E)$ to a balanced bipartite graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ by introducing dummy vertices and edges. The extended graph allows us to write the inequality constraint $\mathbf{B}^\top x = \mathbf{1}_V$ equivalently as the linear constraint $2|R|\widetilde{\mathbf{B}}^\top\binom{x}{x^{\mathsf{dum}}} = d$ for some defined $d \in \mathbb{R}^{\widetilde{V}}$ as some properly-extended vector of $\mathbf{1}_V$. This allows us to apply known matrix scaling solver to such approximating Sinkhorn distance objective in literature.

We prove this is a family of (ϵ, γ)-CROs (see Lemma 10 in full version). The canonical solver for this family uses truncation to E as Round and uses the matrix scaling solver from [12] based on box-constrained Newton's method as Solve, which finds an ϵ-approximate solution of (15) in time $\widetilde{O}(n^2/\epsilon)$. Combining all these components with the DDBM framework in Corollary 8 leads to the following DDBM solver based on approximating Sinkhorn distances.

▶ **Theorem 10.** *Let $G = (V, E)$ be bipartite and $\epsilon \in [\Omega(m^{-3}), 1)$. There is a randomized algorithm for the DDBM problem which maintains an ϵ-approximate matching with probability $1 - n^{-\Omega(1)}$, based on matrix scaling solver of [12], running in time $\widetilde{O}\left(n^2\epsilon^{-3}\right)$.*

Alternatively, for the same (ϵ, γ)-CRO family as in (15), one can use the same Round procedure and the recent high-accuracy almost-linear time graph flow problems solver of [11] for Solve as a canonical solver. Since this new solver can find high-accuracy solutions of entropic-regularized problems of the form (15) within a runtime of $(|E_0| + O(|V|))^{1+o(1)} = m^{1+o(1)}$, this gives a third DDBM solver, which yields and improved an dependence on ϵ^{-1}.

▶ **Theorem 11.** *Let $G = (V, E)$ be bipartite and $\epsilon \in [\Omega(m^{-3}), 1)$. There is a randomized algorithm for the DDBM problem which maintains an ϵ-approximate matching with probability $1 - n^{-\Omega(1)}$, based on the Sinkhorn objective solver of [11], running in time $m^{1+o(1)}\epsilon^{-2}$.*

4 Regularized box-simplex games

In this section, we develop a high-accuracy solver for regularized box-simplex games:

$$\min_{x\in\Delta^m}\max_{y\in[0,1]^n} f_{\mu,\epsilon}(x, y) := y^\top\mathbf{A}^\top x + c^\top x - b^\top y + \mu H(x) - \frac{\epsilon}{2}\left(y^2\right)^\top|\mathbf{A}|^\top x, \\ \text{where } \ H(x) := \sum_{i\in[m]} x_i \log x_i \text{ is the standard entropic regularizer,} \tag{16}$$

where we recall absolute values and squaring act entrywise.

For ease of presentation, we make the following assumptions for some $\delta > 0$.

1. Upper bounds on entries: $\|\mathbf{A}\|_\infty \leq 1$, $\|b\|_\infty \leq B_{\max}$, $\|c\|_\infty \leq C_{\max}$. For simplicity, we assume $B_{\max} \geq C_{\max} \geq 1$; else, $C_{\max} \leftarrow \max(1, C_{\max})$ and $B_{\max} \leftarrow \max(C_{\max}, B_{\max})$.
2. Lower bounds on matrix column entries: $\max_i |\mathbf{A}_{ij}| \geq \delta$ for every $j \in [n]$.

We defer the detailed arguments of why these assumptions are without loss of generality to Appendix B in the full version. Our algorithm acts on the induced (monotone) gradient operator of the regularized box-simplex objective (16), namely $(\nabla_x f_{\mu,\epsilon}(x,y), -\nabla_y f_{\mu,\epsilon}(x,y))$, defined as

$$g_{\mu,\epsilon}(x,y) := \left(\mathbf{A}y + c + \mu(\mathbf{1} + \log(x)) - \frac{\epsilon}{2}\left|\mathbf{A}\right|(y^2), -\mathbf{A}^\top x + b + \epsilon \cdot \mathbf{diag}\,(y)\left|\mathbf{A}\right|^\top x\right). \quad (17)$$

Further, it uses the following joint (non-separable) regularizer of

$$r_{\mu,\epsilon}(x,y) := \rho \sum_{i\in[m]} x_i \log x_i + \frac{1}{\rho} x^\top \left|\mathbf{A}\right| \left(y^2\right) \quad \text{where} \quad \rho = \sqrt{\frac{2\mu}{\epsilon}}, \quad (18)$$

variants of which have been used in [41, 29, 5, 13]. When clear from context, we drop subscripts and refer to these as operator g and regularizer r. Our method is the first high-accuracy near-linear time solver for the regularized problem (16), yielding an $O(\sigma)$-approximate solution with a runtime scaling polylogarithmically in problem parameters and σ. We utilize a variant of the mirror prox [37] method for strongly monotone objectives, which appeared in [9, 13] for regularized saddle point problems with separable regularizers.

In Section 4.1, we present high-level ideas of our algorithm, which uses a mirror prox outer loop (Algorithm 2) and an alternating minimization inner loop (Algorithm 3) to implement outer loop steps; we also provide convergence guarantees. In Section 4.2, we state useful properties of the regularizer (18), and discuss a technical detail ensuring iterate stability in our method. In Section 4.3, we provide our full algorithm for regularized box-simplex games, Algorithm 4 and give guarantees in Theorem 25. Omitted proofs are in Appendix B.

4.1 Algorithmic framework

In this section, we give the algorithmic framework we use to develop our high-precision solver, which combines an outer loop inspired by mirror prox [37] with a custom inner loop for implementing each iteration. We first define an approximate solution for a proximal oracle.

▶ **Definition 12** (Approximate proximal oracle solution). *Given a convex function f over domain $\mathcal{Z}$ and $\sigma \geq 0$, we say $z' \in \mathcal{Z}$ is a* σ-approximate solution *for a proximal oracle if z' satisfies $\langle \nabla f(z'), z' - z\rangle \leq \sigma$. We denote this approximation property by $z' \leftarrow_\sigma \operatorname{argmin}_{z\in\mathcal{Z}} f$.*

We employ such approximate solutions as the proximal oracle within our "outer loop" method. Our outer loop is a variant of mirror prox (Algorithm 2) which builds upon both the mirror prox type method in [41] for solving unregularized box-simplex games and the high-accuracy mirror prox solver developed in [9, 13] for bilinear saddle-point problems on geometries admitting separable regularizers. We first give a high-level overview of the analysis, which requires bounds on two properties. First, suppose g is ν-strongly monotone with respect to regularizer r, i.e.

$$\text{for any } w, z \in \mathcal{Z},\ \langle g(w) - g(z), w - z\rangle \geq \nu \langle \nabla r(w) - \nabla r(z), w - z\rangle. \quad (19)$$

Further, suppose it is α-*relatively Lipschitz* with respect to r and Algorithm 4 (see Definition 1 of [13]), i.e. for any consecutive iterates z_{k-1}, $z_{k-1/2}$, z_k of our algorithm,[4]

$$\left\langle g(z_{k-1/2}) - g(z_{k-1})), z_{k-1/2} - z_k\right\rangle \leq \alpha\Big(V_{z_{k-1/2}}(z_k) + V_{z_{k-1}}(z_{k-1/2})\Big). \quad (20)$$

[4] This property (i.e. relative Lipschitzness restricted to iterates of the algorithm) was referred to as "local relative Lipschitzness" in [13], but we drop the term "local" for simplicity.

With these assumptions, we show that the strongly monotone mirror prox step makes progress by decreasing the divergence to optimal solution $V_{z_k}(z^\star)$ by a factor of $\frac{\alpha}{\alpha+\nu}$: this implies $\widetilde{O}(\frac{\alpha}{\nu})$ iterations suffice for finding a high-accuracy solution. We provide the formal convergence guarantee of MirrorProx in Proposition 13, which also accommodates the error of each approximate proximal step used in the algorithm. This convergence guarantee is generic and does not rely on the concrete structure of g, r in our box-simplex problem.

Algorithm 2 MirrorProx().

Input: $\frac{\sigma}{2}$-approximate proximal oracle, operator and regularizer pair (g, r) such that g is ν-strongly monotone and α-relatively Lipschitz with respect to r

Parameters: Number of iterations K

$z_0 \leftarrow \text{argmin}_{z\in\mathcal{Z}} r(z)$

for $k = 1, \ldots, K$ **do**

  $z_{k-1/2} \leftarrow_{\sigma/2} \text{argmin}_{z\in\mathcal{Z}} \left\{\langle g\left(z_{k-1}\right), z\rangle + \alpha V_{z_{k-1}}(z)\right\}$

  $z_k \leftarrow_{\sigma/2} \text{argmin}_{z\in\mathcal{Z}} \left\{\left\langle g\left(z_{k-1/2}\right), z\right\rangle + \alpha V_{z_{k-1}}(z) + \nu V_{z_{k-1/2}}(z)\right\}$

return z_K

▶ **Proposition 13** (Convergence of Algorithm 2). *Given regularizer r with range at most Θ, suppose g is ν-strongly-monotone with respect to r (see (19)), and is α-relatively-Lipschitz with respect to r (see (20)). Let z_K be the output of Algorithm 2. Then, $V^r_{z_K}(z^\star) \le \left(\frac{\alpha}{\nu+\alpha}\right)^K \Theta + \frac{\sigma}{\nu}$.*

Given the somewhat complicated nature of our joint regularizer, we cannot solve the proximal problems required by Algorithm 2 in closed form. Instead, we implement each proximal step to the desired accuracy by using an alternating minimization scheme, similarly to the implementation of approximate proximal steps in [41, 29].

To analyze our algorithm, we use a generic progress guarantee for alternating minimization from [29] to solve each subproblem, stated below.

Algorithm 3 AltMin($\gamma^{\mathsf{x}}, \gamma^{\mathsf{y}}, \mathbf{A}, \theta, T, x_{\text{init}}, y_{\text{init}}$).

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}_{\ge 0}$, $\gamma^{\mathsf{x}} \in \mathbb{R}^m$, $\gamma^{\mathsf{y}} \in \mathbb{R}^n$, $T \in \mathbb{N}$, $\theta > 0$, $x_{\text{init}} \in \Delta^m$, $y_{\text{init}} \in [0,1]^n$

Output: Approximate minimizer to $\langle(\gamma^{\mathsf{x}}, \gamma^{\mathsf{y}}), z\rangle + \theta r(z)$ for $r(z)$ in (18)

$x^{(0)} \leftarrow x_{\text{init}}$, $y^{(0)} \leftarrow y_{\text{init}}$;

for $0 \le t \le T$ **do**

  $x^{(t+1)} \leftarrow \text{argmin}_{x\in\Delta^m} \left\{\langle\gamma^{\mathsf{x}}, x\rangle + \theta r(x, y^{(t)})\right\}$;

  $y^{(t+1)} \leftarrow \text{argmin}_{y\in[0,1]^n} \left\{\langle\gamma^{\mathsf{y}}, y\rangle + \theta r(x^{(t+1)}, y)\right\}$;

return $(x^{(T+1)}, y^{(T)})$

▶ **Lemma 14** (Alternating minimization progress, Lemma 5 and Lemma 7, [29]). *Let $r : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ be jointly convex, $\theta > 0$, and γ^x and γ^y be linear operators on $\mathcal{X}, \mathcal{Y}$. Define*

$$x_{\mathsf{OPT}}, y_{\mathsf{OPT}} = \text{argmin}_{x\in\mathcal{X}}\text{argmin}_{y\in\mathcal{Y}} f(x, y) := \langle\gamma^x, x\rangle + \langle\gamma^y, y\rangle + \theta r(x, y). \tag{21}$$

Suppose $f(x, y)$ is twice-differentiable and satisfies: for all $x' \ge \frac{1}{2}x$ entrywise, $x', x \in \mathcal{X}$ and $y', y \in \mathcal{Y}$, $\nabla^2 f(x', y') \succeq \frac{1}{\kappa} f(x, y)$. Then the iterates of Algorithm 3 satisfy

$$f(x^{(t+2)}, y^{(t+1)}) - f(x_{\mathsf{OPT}}, y_{\mathsf{OPT}}) \le \left(1 - \frac{1}{2\kappa}\right)\left(f(x^{(t+1)}, y^{(t)}) - f(x_{\mathsf{OPT}}, y_{\mathsf{OPT}})\right).$$

Combining this lemma with the structure of our regularizer (18), we obtain the following guarantees, showing Algorithm 3 finds a $\frac{\sigma}{2}$-approximate solution to the proximal oracle.

▶ **Corollary 15** (Convergence of Algorithm 3). *Let $\delta, \sigma \in (0,1)$, $\rho \geq 1$. Suppose we are given $\gamma \in \mathcal{Z}_* = \mathcal{X}_* \times \mathcal{Y}_*$ with $\max(\|\gamma^{\mathsf{x}}\|_\infty, \|\gamma^{\mathsf{y}}\|_1) \leq B$, and define the proximal subproblem solution*

$$x_{\mathsf{OPT}}, y_{\mathsf{OPT}} = \operatorname{argmin}_{x\in\Delta^m} \operatorname{argmin}_{y\in[0,1]^n} f(x,y) := \langle \gamma^x, x\rangle + \langle \gamma^y, y\rangle + \theta r(x,y) \text{ for some } \theta > 0.$$

If the Hessian condition in Lemma 14 holds with a constant $\kappa > 0$, and all simplex iterates x of Algorithm 3 satisfy $x \geq \delta$ elementwise, then the algorithm finds a $\frac{\sigma}{2}$-approximate solution to the proximal oracle within $T = O\left(\log\left(\frac{\rho(B+mn\theta)^2}{\delta\sigma\theta}\right)\right)$ iterations.

4.2 Helper lemmas

Before providing our full method and analysis, here we list a few helper lemmas, which we rely on heavily in our later development. The first characterizes a useful property of $r_{\mu,\epsilon}$, showing that its Hessian is locally well-approximated by a diagonal matrix, which induces appropriate local norms for the blocks $x \in \mathcal{X}$, $y \in \mathcal{Y}$. We use this to prove the "strong monotonicity" (Lemma 20) and "relative Lipschitzness" (Lemma 22) bounds required in Section 4.3.

▶ **Lemma 16** (Bounds on regularizer). *Suppose $\mathbf{A} \in \mathbb{R}^{m\times n}$ has $\|\mathbf{A}\|_\infty \leq 1$. For any $z = (x,y) \in \Delta^m \times [0,1]^n$, $r = r_{\mu,\epsilon}$ defined as in (18), and $\bar{x} \in \mathbb{R}^m_{>0}$, $\langle x, \mathbf{A}y\rangle \leq \|x\|_{\mathbf{diag}\left(\frac{1}{\bar{x}}\right)} \|y\|_{\mathbf{diag}\left(|\mathbf{A}|^\top \bar{x}\right)}$. Further, if $\rho \geq 3$, the matrix*

$$\mathbf{D}(x) := \begin{pmatrix} \frac{\rho}{2}\mathbf{diag}\left(\frac{1}{x}\right) & \mathbf{0} \\ \mathbf{0} & \frac{1}{\rho}\mathbf{diag}\left(|\mathbf{A}|^\top x\right) \end{pmatrix} \tag{22}$$

satisfies the following relationship with the Hessian matrix of $r(z)$:

$$\mathbf{D}(x) \preceq \nabla^2 r(z) = \begin{pmatrix} \rho \cdot \mathbf{diag}\left(\frac{1}{x}\right) & \frac{2}{\rho}\mathbf{A}\mathbf{diag}\left(y\right) \\ \frac{2}{\rho}\mathbf{diag}\left(y\right)\mathbf{A}^\top & \frac{2}{\rho}\mathbf{diag}\left(|\mathbf{A}|^\top x\right) \end{pmatrix} \preceq 4\mathbf{D}(x). \tag{23}$$

We also introduce the following notion of a padding oracle (cf. Definition 2 of [10]), which helps us control the multiplicative stability of iterates when running our algorithm.

▶ **Definition 17.** *Given $\delta > 0$, and any $\bar{z} = (\bar{x}, y) \in \Delta^m \times [0,1]^n$, a padding oracle $\mathcal{O}_\delta$ returns $z = (x,y)$ by setting $\hat{x}_i = \max(\bar{x}_i, \delta)$ coordinate-wise and letting $x = \frac{\hat{x}}{\|\hat{x}\|_1}$.*

This padding oracle has two merits which we exploit. First, the error incurred due to padding is small proportional to the padding size δ, which finds usage in proving the correctness of our main algorithm, Algorithm 4 (see Proposition 24).

▶ **Lemma 18** (Error of padding, cf. Lemma 6, [10]). *For $\delta > 0$ and $\bar{z} = (\bar{x}, y) \in \Delta^m \times [0,1]^n$ let $z = (x,y) \in \Delta^m \times [0,1]^n$ where $x = \mathcal{O}_\delta(\bar{x})$ (Definition 17), then for r in (18), and any $w \in \mathcal{Z} = \Delta^m \times [0,1]^n$, $V_z^r(w) - V_{\bar{z}}^r(w) \leq \left(\rho + \frac{8}{\rho}\right) m\delta$.*

Second, padding ensures that the iterates of our algorithm satisfy $x = \Omega(\delta)$ entrywise, i.e. no entries of our simplex iterates x are too small. This helps ensure the stability of iterates throughout one call of Algorithm 3, formally through the next lemma.

▶ **Lemma 19** (Iterate stability in Algorithm 3). *Suppose $\epsilon \leq 1$, $\rho \geq 6$, and $\alpha \geq \frac{36}{\rho}(\mu \log \frac{4}{\delta} + 3C_{\max})$. Let (x_k, y_k) denote blocks of z_k, the k^{th} iterate of Algorithm 2. In any iteration k of Algorithm 2, calling Algorithm 3 to implement Line 3, if $x_{k-1} \geq \frac{\delta}{2}$ entrywise, $x^{(t+1)} \in x_{k-1} \cdot \left[\exp\left(-\frac{1}{9}\right), \exp\left(\frac{1}{9}\right)\right]$, for all $t \in [T]$. Calling Algorithm 3 to implement Line 4, if $x_{k-1/2} \geq \frac{\delta}{4}$ entrywise, $x^{(t+1)} \in x_{k-1}^{\frac{\alpha}{\alpha+\nu}} \circ x_{k-1/2}^{\frac{\nu}{\alpha+\nu}} \cdot \left[\exp\left(-\frac{1}{9}\right), \exp\left(\frac{1}{9}\right)\right]$ for all $t \in [T]$.*

4.3 Regularized box-simplex solver and its guarantees

We give our full high-accuracy regularized box-simplex game solver as Algorithm 4, which combines Algorithm 2, Algorithm 3, and a padding step to ensure stability. For space considerations, we defer the complete statement of algorithm to the full version of the paper.

Algorithm 4 RegularizedBS$(\mathbf{A}, b, c, \epsilon, \mu, \sigma)$.

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^m$, $b \in \mathbb{R}^n$, accuracy $\sigma \in (m^{-10}, 1)$, $72\epsilon \le \mu \le 1$

Output: Approximate solution pair (x, y) to (16)

1 **Global:** $\delta \leftarrow \frac{\epsilon\sigma^2}{m^2}$, $\rho \leftarrow \sqrt{\frac{2\mu}{\epsilon}}$, $\nu \leftarrow \frac{1}{2}\sqrt{\frac{\mu\epsilon}{2}}$, $\alpha \leftarrow 18C_{\max} + 32\sqrt{\frac{\mu\epsilon}{2}}\log\frac{4}{\delta}$

2 **Global:** $T \leftarrow O\left(\log\frac{mnB_{\max}\alpha\rho}{\delta\sigma}\right)$, $K \leftarrow O\left(\frac{\alpha}{\nu}\log\left(\frac{\nu\log m}{\sigma}\right)\right)$ for appropriate constants

3 $(x_0, y_0) \leftarrow (\frac{1}{m}\cdot\mathbf{1}_m, \mathbf{0}_n)$

4 **for** $k = 1$ *to* K **do**

5 $\quad(\gamma^{\mathsf{x}}, \gamma^{\mathsf{y}}) \leftarrow \mathsf{GradBS}(x_{k-1}, y_{k-1}, x_{k-1}, y_{k-1}, 0)$

6 $\quad(x_{k-\frac{1}{2}}, y_{k-\frac{1}{2}}) \leftarrow \mathsf{AltminBS}(\gamma^{\mathsf{x}}, \gamma^{\mathsf{y}}, \alpha, x_{k-1}, y_{k-1})$

7 $\quad(\gamma^{\mathsf{x}}, \gamma^{\mathsf{y}}) \leftarrow \mathsf{GradBS}(x_{k-\frac{1}{2}}, y_{k-\frac{1}{2}}, x_{k-1}, y_{k-1}, \nu)$

8 $\quad(x^{(T+1)}, y^{(T)}) \leftarrow \mathsf{AltminBS}(\gamma^{\mathsf{x}}, \gamma^{\mathsf{y}}, \alpha+\nu, x_{k-\frac{1}{2}}, y_{k-\frac{1}{2}})$

9 $\quad x_k \leftarrow \frac{1}{\left\|\max\left(x^{(T+1)}, \delta\right)\right\|_1}\cdot\max\left(x^{(T+1)}, \delta\right)$, $y_k \leftarrow y^{(T)}$ ▷ Implement padding $\mathcal{O}_\delta(x^{(T+1)})$

10 **function** $\mathsf{GradBS}(x, y, x_0, y_0, \Theta)$

11 $\quad g^{\mathsf{x}} \leftarrow \mathbf{A}y + c + \mu(\mathbf{1} + \log(x)) - \frac{\epsilon}{2}|\mathbf{A}|(y^2)$, $g^{\mathsf{y}} \leftarrow -\mathbf{A}^\top x + b + \epsilon\,\mathbf{diag}(y)\,|\mathbf{A}|^\top x$

12 $\quad g^{\mathsf{x}}_r \leftarrow -\alpha\rho(1 + \log x_0) - \frac{\alpha}{\rho}|\mathbf{A}|y_0^2 - \Theta\rho(1 + \log x) - \frac{\Theta}{\rho}|\mathbf{A}|y^2$

13 $\quad g^{\mathsf{y}}_r \leftarrow -\frac{2\alpha}{\rho}\mathbf{diag}(y_0)\,|\mathbf{A}|^\top x_0 - \frac{2\Theta}{\rho}\mathbf{diag}(y)\,|\mathbf{A}|^\top x)$

14 $\quad$**return** $(g^{\mathsf{x}} + g^{\mathsf{x}}_r, g^{\mathsf{y}} + g^{\mathsf{y}}_r)$

15 **function** $\mathsf{AltminBS}(\gamma^{\mathsf{x}}, \gamma^{\mathsf{y}}, \theta, x^{(0)}, y^{(0)})$ ▷ Implement approximate proximal oracle via AltMin

16 $\quad$**for** $0 \le t \le T$ **do**

17 $\quad\quad x^{(t+1)} \leftarrow \frac{1}{\left\|\exp\left(-\frac{1}{\theta\rho}\gamma^{\mathsf{x}} - \frac{1}{\rho^2}|\mathbf{A}|\left(y^{(t)}\right)^2\right)\right\|_1}\cdot\exp\left(-\frac{1}{\theta\rho}\gamma^{\mathsf{x}} - \frac{1}{\rho^2}|\mathbf{A}|\left(y^{(t)}\right)^2\right)$

18 $\quad\quad y^{(t+1)} \leftarrow \mathrm{med}\left(0, 1, -\frac{\rho}{2\theta}\cdot\frac{\gamma^{\mathsf{y}}}{|\mathbf{A}|^\top x^{(t+1)}}\right)$

19 $\quad$**return** $(x^{(T+1)}, y^{(T)})$

In order to analyze the convergence of Algorithm 4, we begin by observing that the operator in (17) satisfies strong monotonicity with respect to our regularizer (18).

▶ **Lemma 20** (Strong monotonicity). *Let $\mu \ge \frac{\epsilon}{2}$ and $\rho := \sqrt{\frac{2\mu}{\epsilon}}$. The gradient operator $g_{\mu,\epsilon}$ (17) is $\nu := \frac{1}{2}\sqrt{\frac{\mu\epsilon}{2}}$-strongly monotone (see (19)) with respect to $r_{\mu,\epsilon}$ defined in (18).*

Next, we show iterate stability through each loop of alternating minimization (i.e. from Line 5 to Line 6, and Line 7 to Line 8 respectively), via Lemma 19.

▶ **Corollary 21** (Iterate stability in Algorithm 4). *Assume the same parameter bounds as Lemma 19, and that $\delta \in (0, m^{-1})$. In the k^{th} outer loop of Algorithm 4, $x_{k-1} \ge \frac{\delta}{2}$ entrywise. Further, for all iterates $x^{(t+1)}$ computed in Line 5 to Line 6 and x_{OPT} as defined in (21) with $\theta = \alpha$, $\frac{1}{2}x_{k-1} \le x^{(t+1)}, x_{\mathsf{OPT}} \le 2x_{k-1}$, and $x^{(t+1)}, x_{\mathsf{OPT}} \ge \frac{\delta}{4}$, entrywise. Similarly, for all iterates $x^{(t+1)}$ computed in Line 7 to Line 8 and x_{OPT} as defined in (21) with $\theta = \alpha + \nu$, $\frac{1}{2}x_{k-1/2} \le x^{(t+1)}, x_{\mathsf{OPT}} \le 2x_{k-1/2}$ and $x^{(t+1)}, x_{\mathsf{OPT}} \ge \frac{\delta}{4}$, entrywise.*

Under iterate stability, our next step is to prove that our operator $g_{\mu,\epsilon}$ is relatively Lipschitz with respect to our regularizer $r_{\mu,\epsilon}$ (as defined in (20)).

▶ **Lemma 22** (Relative Lipschitzness). *Assume the same parameter bounds as in Lemma 19. In the k^{th} outer loop of Algorithm 4, let $\bar{z}_k \leftarrow (x^{(T+1)}, y^{(T)})$ from Line 8 be z_k before the padding operation. Then, $x_{k-1/2}, \bar{x}_k \in [\frac{1}{2}x_{k-1}, 2x_{k-1}]$ elementwise and*

$$\left\langle g(z_{k-1/2}) - g(z_{k-1}), z_{k-1/2} - \bar{z}_k \right\rangle \le \alpha \left(V_{z_{k-1}}(z_{k-1/2}) + V_{z_{k-1/2}}(\bar{z}_k) \right) \quad \text{for } \alpha = 4 + 32\sqrt{\frac{\mu\epsilon}{2}}.$$

Next, we give a convergence guarantee on the inner loops (from Line 5 to Line 6, and Line 7 to Line 8) in Algorithm 4, as an immediate consequence of Corollary 15.

▶ **Corollary 23** (Inner loop convergence in Algorithm 4). *Assume the same parameter bounds as in Lemma 19. For γ defined in Line 5, suppose for an appropriate constant $T = \Omega\left(\log \frac{mnB_{\max}\alpha\rho}{\delta\sigma}\right)$. Then, for all k iterate $z_{k-1/2} = (x_{k-1/2}, y_{k-1/2})$ of Line 6 satisfies*

$$\left\langle \nabla g(z_{k-1}) + \alpha \nabla V_{z_{k-1}}(z_{k-1/2}), z_{k-1/2} - w \right\rangle \le \frac{\nu\sigma}{4}, \text{ for all } w \in \mathcal{Z}.$$

Similarly, for γ defined in Line 7, iterate $\bar{z}_k = (x_{(T+1)}, y_{(T)})$ of Line 8 satisfies

$$\left\langle \nabla g(z_{k-1}) + \alpha \nabla V_{z_{k-1}}(\bar{z}_k) + \nu \nabla V_{z_{k-1/2}}(\bar{z}_k), \bar{z}_k - w \right\rangle \le \frac{\nu\sigma}{4}, \text{ for all } w \in \mathcal{Z}.$$

We now analyze the progress made by each outer loop of Algorithm 4. The proof is very similar to that of Proposition 13; the only difference is controlling the extra error incurred in the padding step of Line 9, which we bound via Lemma 18.

▶ **Proposition 24** (Convergence of Algorithm 4). *Assume the same parameter bounds as in Lemma 19, and that $\delta \le \frac{\sigma}{4\rho\alpha m}$. Algorithm 4 returns z_K satisfying $V^r_{z_K}(z^\star) \le \frac{3\sigma}{\nu}$, letting (for an appropriate constant) $K = \Omega(\frac{\alpha}{\nu}\log(\frac{\nu \log m}{\sigma}))$.*

We are now ready to prove the main theorem of this section, which gives a complete convergence guarantee of Algorithm 4 by combining our previous claims.

▶ **Theorem 25** (Regularized box-simplex solver). *Given regularized box-simplex game (16) with $72\epsilon \le \mu \le 1$ and optimizer $(x^\star, y^\star)$, and letting $\sigma \in (m^{-10}, 1)$,* RegularizedBS *(Algorithm 4) returns x^K satisfying $\left\|x^K - x^\star\right\|_1 \le \frac{\sigma}{C_{\max}\log^2 m}$ and $\max_{y\in[0,1]^n} f_{\mu,\epsilon}(x^K, y) - f_{\mu,\epsilon}(x^\star, y^\star) \le \sigma$. The total runtime of the algorithm is $O(\text{nnz}(\mathbf{A}) \cdot (\frac{C_{\max}}{\sqrt{\mu\epsilon}} + \log(\frac{m}{\sigma\epsilon})) \cdot \log(\frac{C_{\max}\log m}{\sigma}) \log(\frac{mnB_{\max}}{\sigma}))$.*

As a corollary, we obtain an approximate solver for regularized box-simplex games in the following form (which in particular does not include a quadratic regularizer):

$$\min_{x\in\Delta^m} \max_{y\in[0,1]^n} f_\mu(x, y) = y^\top \mathbf{A}^\top x + c^\top x - b^\top y + \mu H(x), \text{ where } H(x) := \sum_{i\in[m]} x_i \log x_i. \quad (24)$$

▶ **Corollary 26** (Half-regularized approximate solver). *Given regularized box-simplex game (24) with regularization parameters $72\epsilon \le \mu \le 1$ and optimizer $(x^\star, y^\star)$, and letting $\epsilon \in (m^{-10}, 1)$, Algorithm 4 with $\sigma \leftarrow \frac{\epsilon}{2}$ returns x^K satisfying $\max_{y\in[0,1]^n} f_\mu(x^K, y) - f_\mu(x^\star, y^\star) \le \epsilon$. The total runtime of the algorithm is $O\left(\text{nnz}(\mathbf{A}) \cdot (\frac{C_{\max}}{\sqrt{\mu\epsilon}} + \log\left(\frac{m}{\epsilon}\right)) \cdot \log\left(\frac{C_{\max}\log m}{\epsilon}\right) \log\left(\frac{mnB_{\max}}{\epsilon}\right)\right)$.*

References

1 Ittai Abraham, David Durfee, Ioannis Koutis, Sebastian Krinninger, and Richard Peng. On fully dynamic graph sparsifiers. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 335–344. IEEE Computer Society, 2016.

2 Zeyuan Allen-Zhu, Yuanzhi Li, Rafael Mendes de Oliveira, and Avi Wigderson. Much faster algorithms for matrix scaling. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 890–901, 2017.

3 Jason Altschuler, Francis Bach, Alessandro Rudi, and Jonathan Niles-Weed. Massively scalable sinkhorn distances via the nyström method. *Advances in neural information processing systems*, 32, 2019.

4 Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1964–1974, 2017.

5 Sepehr Assadi, Arun Jambulapati, Yujia Jin, Aaron Sidford, and Kevin Tian. Semi-streaming bipartite matching in fewer passes and optimal space. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Alexandria, VA, USA, January 9 - 12, 2022*, 2022.

6 Aaron Bernstein, Maximilian Probst Gutenberg, and Thatchaphol Saranurak. Deterministic decremental reachability, scc, and shortest paths via directed expanders and congestion balancing. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1123–1134, 2020.

7 Digvijay Boob, Saurabh Sawlani, and Di Wang. Faster width-dependent algorithm for mixed packing and covering lps. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15253–15262, 2019.

8 Bartlomiej Bosek, Dariusz Leniowski, Piotr Sankowski, and Anna Zych. Online bipartite matching in offline time. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 384–393. IEEE, 2014.

9 Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix games. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11377–11388, 2019.

10 Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Coordinate methods for matrix games. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 283–293, 2020. doi:10.1109/FOCS46700.2020.00035.

11 Li Chen, Rasmus Kyng, Yang P Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. *arXiv preprint*, 2022. arXiv:2203.00671.

12 Michael B Cohen, Aleksander Madry, Dimitris Tsipras, and Adrian Vladu. Matrix scaling and balancing via box constrained newton's method and interior point methods. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 902–913. IEEE, 2017.

13 Michael B. Cohen, Aaron Sidford, and Kevin Tian. Relative lipschitzness in extragradient methods and a direct recipe for acceleration. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, pages 62:1–62:18, 2021.

14 Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2292–2300, 2013.

15 David Durfee, Yu Gao, Gramoz Goranci, and Richard Peng. Fully dynamic spectral vertex sparsifiers and applications. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 914–925. ACM, 2019.

16 Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 1366–1375, 2018.

17 Yu Gao, Yang P. Liu, and Richard Peng. Fully dynamic electrical flows: Sparse maxflow faster than goldberg-rao. *CoRR*, abs/2101.07233, 2021. `arXiv:2101.07233`.

18 Ankit Garg, Leonid Gurvits, Rafael Mendes de Oliveira, and Avi Wigderson. Operator scaling: Theory and applications. *Found. Comput. Math.*, 20(2):223–290, 2020.

19 Gramoz Goranci. Dynamic graph algorithms and graph sparsification: New techniques and connections. *CoRR*, abs/1909.06413, 2019. `arXiv:1909.06413`.

20 Gramoz Goranci, Monika Henzinger, and Pan Peng. The power of vertex sparsifiers in dynamic graph algorithms. In Kirk Pruhs and Christian Sohler, editors, *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, volume 87 of *LIPIcs*, pages 45:1–45:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

21 Gramoz Goranci, Monika Henzinger, and Pan Peng. Dynamic Effective Resistances and Approximate Schur Complement on Separable Graphs. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

22 Gramoz Goranci, Harald Räcke, Thatchaphol Saranurak, and Zihan Tan. The expander hierarchy and its applications to dynamic graph algorithms. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2212–2228. SIAM, 2021.

23 Fabrizio Grandoni, Stefano Leonardi, Piotr Sankowski, Chris Schwiegelshohn, and Shay Solomon. $(1+\epsilon)$-approximate incremental matching in constant deterministic amortized time. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1886–1898, 2019.

24 Manoj Gupta. Maintaining approximate maximum matching in an incremental bipartite graph in polylogarithmic update time. In *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India*, pages 227–239, 2014.

25 Manoj Gupta and Richard Peng. Fully dynamic $(1+\epsilon)$-approximate matchings. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 548–557, 2013.

26 Maximilian Probst Gutenberg and Christian Wulff-Nilsen. Fully-dynamic all-pairs shortest paths: Improved worst-case time and space bounds. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2562–2574. SIAM, 2020.

27 Kathrin Hanauer, Monika Henzinger, and Christian Schulz. Recent advances in fully dynamic graph algorithms. *CoRR*, abs/2102.11169, 2021. `arXiv:2102.11169`.

28 Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 21–30, 2015.

29 Arun Jambulapati, Aaron Sidford, and Kevin Tian. A direct $\tilde{O}(1/\epsilon)$ iteration parallel algorithm for optimal transport. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11355–11366, 2019.

30 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 Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 217–226, 2014.

31 Peter Kiss. Improving update times of dynamic matching algorithms from amortized to worst case. *CoRR*, abs/2108.10461, 2021. `arXiv:2108.10461`.

32 Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3sum conjecture. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1272–1287, 2016.

33 Yin Tat Lee and Aaron Sidford. Efficient inverse maintenance and faster algorithms for linear programming. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 230–249, 2015.

34 Tianyi Lin, Nhat Ho, and Michael I. Jordan. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3982–3991, 2019.

35 Nathan Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 644–652, 1998.

36 Danupon Nanongkai, Thatchaphol Saranurak, and Christian Wulff-Nilsen. Dynamic minimum spanning forest with subpolynomial worst-case update time. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 950–961. IEEE Computer Society, 2017.

37 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, 2004.

38 Yurii Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

39 Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Math. Program.*, 109(2-3):319–344, 2007.

40 Jonah Sherman. Nearly maximum flows in nearly linear time. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 263–269, 2013.

41 Jonah Sherman. Area-convexity, ℓ_∞ regularization, and undirected multicommodity flow. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 452–460. ACM, 2017.

42 Aaron Sidford and Kevin Tian. Coordinate methods for accelerating ℓ_∞ regression and faster approximate maximum flow. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science*, pages 922–933. IEEE, 2018.

43 Jan van den Brand, Yin Tat Lee, Yang P. Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 859–869, 2021.

44 François-Xavier Vialard. An elementary introduction to entropic regularization and proximal methods for numerical optimal transport, 2019.

45 David Wajc. Rounding dynamic matchings against an adaptive adversary. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 194–207, 2020.