Rethinking Warm-Starts with Predictions: Learning Predictions Close to Sets of Optimal Solutions for Faster L-/L$^\natural$-Convex Function Minimization

Shinsaku Sakaue [1] Taihei Oki [1]

Abstract

An emerging line of work has shown that machine-learned predictions are useful to warm-start algorithms for discrete optimization problems, such as bipartite matching. Previous studies have shown time complexity bounds proportional to some distance between a prediction and an optimal solution, which we can approximately minimize by learning predictions from past optimal solutions. However, such guarantees may not be meaningful when multiple optimal solutions exist. Indeed, the dual problem of bipartite matching and, more generally, *L-/L$^\natural$-convex function minimization* have *arbitrarily many* optimal solutions, making such prediction-dependent bounds arbitrarily large. To resolve this theoretically critical issue, we present a new warm-start-with-prediction framework for L-/L$^\natural$-convex function minimization. Our framework offers time complexity bounds proportional to the distance between a prediction and the *set of all optimal solutions*. The main technical difficulty lies in learning predictions that are provably close to sets of all optimal solutions, for which we present an online-gradient-descent-based method. We thus give the first polynomial-time learnability of predictions that can provably warm-start algorithms regardless of multiple optimal solutions.

1. Introduction

Algorithms with predictions (Mitzenmacher & Vassilvitskii, 2021)—improving algorithms performance with predictions learned from data—is a rapidly growing research field. Seminal work by Dinitz et al. (2021) has initiated the study of using predictions to warm-start discrete optimization algorithms. A brief description of their result for weighted perfect bipartite matching problems is as follows. Consider finding a maximum-weight perfect matching in a bipartite graph (V, E) with an equal-sized bipartition $V = L \cup R$ and edge weights $w \in \mathbb{Z}^E$.[1] The dual of this problem is written as a linear program (LP) with variables $p = (s, t) \in \mathbb{R}^V$:

$$\begin{aligned} \underset{s \in \mathbb{R}^L, t \in \mathbb{R}^R}{\text{minimize}} \quad & \sum_{i \in L} s_i - \sum_{j \in R} t_j \\ \text{subject to} \quad & s_i - t_j \geq w_{ij} \quad \forall ij \in E. \end{aligned} \tag{1}$$

The authors showed that given a prediction $\hat{p} \in \mathbb{R}^V$ of *some* optimal dual solution p^*, we can efficiently convert $\hat{p}$ into an initial feasible solution p°, and the Hungarian method warm-started by p° runs in $\mathrm{O}(|E|\sqrt{|V|}\|p^* - \hat{p}\|_1)$ time, whereas the worst-case running time is $\mathrm{O}(|E||V|)$. Moreover, given about $\Omega(|V|^3)$ optimal solutions p^* drawn i.i.d. from a fixed distribution $\mathcal{D}$, we can learn $\hat{p}$ that approximately minimizes $\mathbb{E}_{p^* \sim \mathcal{D}}[\|p^* - \hat{p}\|_1]$ via empirical risk minimization. In a nutshell, learning predictions from past optimal solutions can provably accelerate the Hungarian method.

The above argument, however, has a subtle but critical pitfall: there *always* exist *arbitrarily many* optimal dual solutions. To see this, let p' be an optimal solution to (1). Then, adding any vector in the all-one direction to p' does not change the objective function value and the left-hand sides of the constraints; therefore, $p' + \zeta\mathbf{1}$ is also optimal for all $\zeta \in \mathbb{R}$. Hence, the $\mathrm{O}(|E|\sqrt{|V|}\|p^* - \hat{p}\|_1)$-time bound requires us to select one optimal solution p^*, and the bound can be arbitrarily large if selected p^* is far away from $\hat{p}$. One may think such a concern is unnecessary since the Hungarian method warm-started by $\hat{p}$ would return p^* close to $\hat{p}$. This idea, however, makes p^* selected depending on $\hat{p}$, and no existing results on the learnability of predictions $\hat{p}$ can deal with such dependence. We further detail this issue in Appendix A.

The cause of this troublesome situation is that an optimal solution p^* is not unique for a given bipartite matching instance. By contrast, *the set of all optimal solutions* is unique. Therefore, the distance between $\hat{p}$ and the set of all optimal solutions (or equivalently, the minimum distance between $\hat{p}$ and an optimal solution) is a well-defined measure to quantify the speed-up gained by using prediction $\hat{p}$. Moreover,

[1] The University of Tokyo, Japan. Correspondence to: Shinsaku Sakaue <sakaue@mist.i.u-tokyo.ac.jp>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1] While the minimum-weight setting was originally studied, we describe the maximum-weight setting as in (Sakaue & Oki, 2022).

Table 1: Improved time complexity bounds for the problems studied in (Sakaue & Oki, 2022) (see also Table 1 therein). For weighted perfect bipartite matching and discrete energy minimization, n and m are the sizes of vertex and edge sets, respectively. For weighted matroid intersection, r is the rank of matroids, n is the ground-set size, and τ is the running time of independence oracles.

Problem	Time complexity
Weighted perfect bipartite matching	$\mathrm{O}(m\sqrt{n} \cdot \bar{\mu}(\hat{p}; g))$
Weighted matroid intersection	$\mathrm{O}(\tau n r^{1.5} \cdot \bar{\mu}(\hat{p}; g))$
Discrete energy minimization	$\mathrm{O}(mn^2 \cdot \bar{\mu}(\hat{p}; g))$

this idea can strengthen distance-dependent time complexity bounds by taking the minimum among all optimal solutions.

1.1. Our Contribution

We present a new framework with time complexity bounds proportional to the distance between prediction $\hat{p}$ and the set of all optimal solutions. Building on a recent improvement (Sakaue & Oki, 2022) of (Dinitz et al., 2021), we develop our framework for *L-/L$^\natural$-convex function minimization*, a broad class of discrete optimization problems, such as the weighted perfect bipartite matching, weighted matroid intersection, and discrete energy minimization. The pitfall mentioned above also exists in L-/L$^\natural$-convex minimization (see Remark 2.3) and has remained open in the prior work.

We here give some informal definitions for convenience (see Section 2 for details). Let g be an L-/L$^\natural$-convex function to be minimized, which represents both an objective function and constraints, taking $+\infty$ if infeasible. We quantify the distance between prediction $\hat{p}$ and the set, $\mathrm{conv}(\mathrm{argmin}\, g)$, of all optimal solutions with the *$\ell_\infty^\pm$-norm*; let $\bar{\mu}(\hat{p}; g)$ denote this distance. Our high-level idea is to use $\bar{\mu}(\hat{p}; g)$ instead of any distance defined with some fixed optimal p^*. Although the idea is simple, it involves two unprecedented challenges: (i) to show that algorithms warm-started with $\hat{p}$ run in time proportional to $\bar{\mu}(\hat{p}; g)$ and (ii) to learn $\hat{p}$ that approximately minimizes $\mathbb{E}_g[\bar{\mu}(\hat{p}; g)]$. We describe how to achieve them.

Section 3 shows that an L-/L$^\natural$-convex minimization method warm-started with $\hat{p}$ enjoys a time complexity bound proportional to $\bar{\mu}(\hat{p}; g)$. As with (Sakaue & Oki, 2022), we employ the steepest descent method for solving L-/L$^\natural$-convex minimization, and we additionally utilize a fact that it converges to an optimal solution closest to an initial feasible solution. Our analysis applies to all the problems studied in (Sakaue & Oki, 2022) and improves their time complexity bounds, which are proportional to the ℓ_∞-distance, $\|p^* - \hat{p}\|_\infty$, between $\hat{p}$ and some fixed optimal p^*. Table 1 summarizes our improved time complexity bounds, and Figure 1 illustrates how our idea improves their previous bounds.

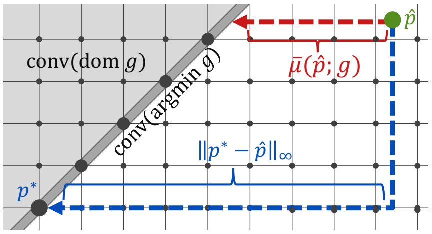


Figure 1: Comparison of our result with $\bar{\mu}(\hat{p}; g)$ and the previous result with $\|p^* - \hat{p}\|_\infty$ (Sakaue & Oki, 2022). Imagine L-convex minimization in $\mathbb{Z}^2$. Let $\hat{p} \in \mathbb{R}^2$ be a given prediction, g an L-convex function, and p^* an optimal solution selected for g. The gray area, $\mathrm{conv}(\mathrm{dom}\, g)$, is (the convex hull of) the feasible region, and the darker area, $\mathrm{conv}(\mathrm{argmin}\, g)$, is the set of all optimal solutions. As discussed in Section 1, $\mathrm{conv}(\mathrm{argmin}\, g)$ has freedom in the all-one direction; hence, $\|p^* - \hat{p}\|_\infty$ can be arbitrarily large if p^* is far from $\hat{p}$. By contrast, $\bar{\mu}(\hat{p}; g)$ uniquely represents the minimum $\ell_\infty^\pm$-distance between $\hat{p}$ and $p^* \in \mathrm{conv}(\mathrm{argmin}\, g)$ closest to $\hat{p}$.

Section 4 presents how to learn $\hat{p}$ that approximately minimizes $\mathbb{E}_g[\bar{\mu}(\hat{p}; g)]$. Similar to (Khodak et al., 2022; Sakaue & Oki, 2022), we prove a regret bound of the online gradient descent method (OGD) for learning $\hat{p}$ and obtain a sample complexity bound via online-to-batch conversion. Our contribution is to obtain those bounds for $\bar{\mu}(\hat{p}; g)$, not for any distance between $\hat{p}$ and fixed optimal p^*. The main difficulty lies in computing subgradients of $\bar{\mu}(\cdot; g)$ used in OGD, for which we use a connection between $\bar{\mu}(\cdot; g)$ and a shortest path problem and Danskin's theorem (see Section 4.3). Also, computing a subgradient requires an inequality system that represents the set, $\mathrm{conv}(\mathrm{argmin}\, g)$, of all optimal solutions, for which we give polynomial-time methods (see Section 5). We thus obtain the first polynomial-time learnability of predictions that can provably warm-start algorithms regardless of multiple optimal solutions. Furthermore, our regret bound is tight up to constant factors, as shown in Appendix D.

Remarks and Limitations. Since we focus on theoretically refining prediction-dependent time complexity bounds, the practical impact would be somewhat limited; still, Section 6 presents some promising empirical results. Also, we do not discuss worst-case bounds since we can bound the worst-case runtime by executing standard algorithms with worst-case guarantees in parallel, as in (Sakaue & Oki, 2022, Section 6). We emphasize that our motivation is to warm-start simple algorithms with predictions, while theoretically fast algorithms, which are often hard to implement and slow in practice, sometimes enjoy better time complexity bounds.

1.2. Related Work

A flurry of recent work has been devoted to going beyond the worst-case analysis of algorithms using predictions. While improving competitive ratios of online algorithms has occupied a central place (Purohit et al., 2018; Bamas et al., 2020; Lykouris & Vassilvitskii, 2021; Azar et al., 2022), the idea has gained increasing attention in various areas, including algorithmic game theory (Agrawal et al., 2022) and data structures (Boffa et al., 2022). A comprehensive list of papers in this field is provided by Lindermayr & Megow.

Besides (Sakaue & Oki, 2022), Chen et al. (2022) have improved the prediction-dependent time complexity bound of (Dinitz et al., 2021) and presented general results for warm-starting various graph algorithms and learning predictions. Polak & Zub (2022) have used predictions to warm-start a maximum-flow algorithm. Although those existing studies have provided time complexity bounds depending on some distance, $\|p^* - \hat{p}\|$, between an optimal solution p^* and a prediction $\hat{p}$, the non-uniqueness of p^*, despite its prevalence, has not been well discussed—p^* has been (implicitly) assumed to be unique.[2] Researchers have also used predictions to accelerate algorithms for support estimation (Eden et al., 2021), the shortest path problem (Feijen & Schäfer, 2023), generalized sorting (Lu et al., 2021), nearest neighbor search (Andoni & Beaglehole, 2022), and clustering (Ergun et al., 2022), while their time complexity analyses are different from those of warm-starts with predictions.

The L-/L$^\natural$-convexity is a fundamental notion in *discrete convex analysis* (Murota, 2003), a discrete analog of convex analysis. It enables us to see various discrete optimization algorithms as the steepest descent method. This viewpoint offers a geometric understanding of warm-starts with predictions; that is, a prediction closer to an optimum is naturally better since an algorithm iteratively approaches an optimum. Although (Sakaue & Oki, 2022) is also based on the steepest descent method, they did not utilize a notable property that it converges to an optimal solution closest to an initial point (see Proposition 2.4), which is a key to obtaining our result.

2. Preliminaries

Let $\lfloor\cdot\rfloor$, $\lceil\cdot\rceil$, and $\lfloor\cdot\rceil$ be the element-wise ceiling, floor, and rounding, respectively, where $\lfloor\cdot\rceil$ rounds down 0.5 fractional parts. For $n \in \mathbb{N}$, let $V = \{1, 2, \dots, n\}$ be a finite ground set of size n. Let $\mathbf{0}, \mathbf{1} \in \mathbb{R}^V$ denote the all-zero and all-one vectors, respectively. For $S \subseteq \mathbb{R}^V$, let $\mathrm{conv}(S) \subseteq \mathbb{R}^V$ be the convex hull of S and δ_S the indicator function of S, i.e., $\delta_S(p) = 0$ if $p \in S$ and $+\infty$ otherwise.

For a function $g : \mathbb{Z}^V \to \mathbb{R} \cup \{+\infty\}$, we define its *effective domain* by $\mathrm{dom}\, g = \left\{ p \in \mathbb{Z}^V \;\middle|\; g(p) < +\infty \right\}$, which represents the feasible region of a minimization problem of the form $\min_{p \in \mathbb{Z}^V} g(p)$. We say g is *proper* if $\mathrm{dom}\, g \neq \emptyset$.

2.1. L-/L$^\natural$-Convex Functions and Sets

We overview the properties of L-/L$^\natural$-convex functions and sets. We refer the reader to (Murota, 2003) for more details.

Let $g : \mathbb{Z}^V \to \mathbb{R} \cup \{+\infty\}$ be a proper function. We say g is *L-convex* if $g(p) + g(q) \geq g(p \vee q) + g(p \wedge q)$ for all $p, q \in \mathbb{Z}^V$, where $\vee$ ($\wedge$) is the element-wise maximum (minimum), and there exists $r \in \mathbb{R}$ such that $g(p + \mathbf{1}) = g(p) + r$ for all $p \in \mathbb{Z}^V$. We say g is *L$^\natural$-convex* if $\tilde{g}(p_0, p) := g(p - p_0\mathbf{1})$ is L-convex on $\mathbb{Z} \times \mathbb{Z}^V$; this is equivalent to $g(p) + g(q) \geq g\left(\left\lceil \frac{p+q}{2} \right\rceil\right) + g\left(\left\lfloor \frac{p+q}{2} \right\rfloor\right)$ for all $p, q \in \mathbb{Z}^V$, analogous to the standard convexity of functions on $\mathbb{R}^V$. Since L-convexity on $\mathbb{Z} \times \mathbb{Z}^V$ and L$^\natural$-convexity on $\mathbb{Z}^V$ are equivalent, we may use whichever is convenient. If g_1 and g_2 are L-/L$^\natural$-convex on $\mathbb{Z}^V$ and $g_1 + g_1$ is proper, $g_1 + g_1$ is L-/L$^\natural$-convex.

The L-/L$^\natural$-convex function minimization, $\min_{p \in \mathbb{Z}^V} g(p)$, is known to contain a wide variety of problems, e.g., the dual of weighted bipartite matching, dual of weighted matroid intersection, and discrete energy minimization, which generalizes minimum-cost flow (see (Murota, 2003, Chapter 9)). In what follows, we make the following basic assumption.

Assumption 2.1. L-/L$^\natural$-convex functions g always have at least one minimizer, i.e., $\mathrm{argmin}\, g \neq \emptyset$.

A non-empty set $S \subseteq \mathbb{Z}^V$ is *L-/L$^\natural$-convex* if its indicator function $\delta_S : \mathbb{Z}^V \to \{0, +\infty\}$ is L-/L$^\natural$-convex. Conversely, if $g : \mathbb{Z}^V \to \mathbb{R} \cup \{+\infty\}$ is L-/L$^\natural$-convex, $\mathrm{dom}\, g \subseteq \mathbb{Z}^V$ is an L-/L$^\natural$-convex set; furthermore, the set of all minimizers, $\mathrm{argmin}\, g \subseteq \mathbb{Z}^V$, is also L-/L$^\natural$-convex (Murota, 2003, Theorem 7.17). We use this fact in Section 4.3. L-/L$^\natural$-convex sets enjoy useful inequality-system representations as follows.

Proposition 2.2 (Murota (2003, Sections 5.3 and 5.5))**.** *For a non-empty set $S \subseteq \mathbb{Z}^V$, the following two are equivalent: (i) S is an L$^\natural$-convex set and (ii) S is written as*

$$\left\{ p \in \mathbb{Z}^V \;\middle|\; \begin{array}{l} \alpha_i \leq p_i \leq \beta_i \textit{ for } i \in V, \\ p_j - p_i \leq \gamma_{ij} \textit{ for distinct } i, j \in V \end{array} \right\} \tag{2}$$

with some $\alpha_i \in \mathbb{Z} \cup \{-\infty\}$, $\beta_i \in \mathbb{Z} \cup \{+\infty\}$, and $\gamma_{ij} \in \mathbb{Z} \cup \{+\infty\}$. Also, S is L-convex if and only if S is written as in (2) *without box constraints, i.e., $\alpha_i = -\infty$ and $\beta_i = +\infty$. The convex hull, $\mathrm{conv}(S) \subseteq \mathbb{R}^V$, of an L-/L$^\natural$-convex set S is also characterized as above replacing $\mathbb{Z}^V$ in* (2) *with $\mathbb{R}^V$.*

Example 1. The dual LP (1) of weighted bipartite matching with variables $(s, t) \in \mathbb{R}^V$ has constraints $s_i - t_j \geq w_{ij}$ for $ij \in E$. These are of the form (2) and represent the convex hull of the L-convex feasible region, or $\mathrm{conv}(\mathrm{dom}\, g)$. Furthermore, given a maximum weight matching $M^* \subseteq E$, a dual feasible (s, t) is optimal if and only if $s_i - t_j = w_{ij}$ for $ij \in M^*$ (see Proposition 5.1). Thus, we can represent the L-convex set of optimal dual solutions, or $\mathrm{conv}(\mathrm{argmin}\, g)$, with additional inequalities $s_i - t_j \leq w_{ij}$ for $ij \in M^*$.

Remark 2.3. The fact that $\mathrm{argmin}\, g$ is written as in (2) immediately implies the non-uniqueness of optimal solutions.

[2] In (Polak & Zub, 2022), their bound is said to hold for every optimal solution, but its non-uniqueness is not correctly handled when learning predictions. See Appendix A.3 for details.

Algorithm 1 Steepest descent method

```
1: $p \leftarrow p^\circ$    $\triangleright$ $p^\circ \in \mathrm{dom}\, g$ is an initial feasible solution.
2: while not converged :
3:     $d \leftarrow \mathrm{argmin}\{\, g(p + d') \mid d' \in \mathcal{N} \,\}$
4:     if $g'(p; d) = 0$ :
5:         return $p$
6:     $\lambda \leftarrow \sup\{\, \lambda' \in \mathbb{Z}_{>0} \mid g'(p; \lambda' d) = \lambda' g'(p; d) \,\}$
7:     $p \leftarrow p + \lambda d$
```

Specifically, if g is L-convex, shifting $p^* \in \mathrm{argmin}\, g$ in the all-one direction never goes out of $\mathrm{argmin}\, g$, as discussed in Section 1, and similar reasoning applies to the L$^\natural$-convex case if such a shifting does not go out of box constraints.

2.2. Steepest Descent for L-/L$^\natural$-Convex Minimization

We can solve L-/L$^\natural$-convex minimization, $\min_{p \in \mathbb{Z}^V} g(p)$, by using the steepest descent method in Algorithm 1, which iterates to proceed along a locally steepest descent direction. The set, $\mathcal{N}$, of local directions is defined as $\mathcal{N} := \{0, +1\}^V$ if g is L-convex and $\mathcal{N} := \{0, -1\}^V \cup \{0, +1\}^V$ if g is L$^\natural$-convex. Let $g'(p; d) := g(p + d) - g(p)$ denote the *slope* of g at $p \in \mathrm{dom}\, g$ in the direction of $d \in \mathbb{Z}^V$.

We detail how Algorithm 1 works. Starting from an initial point $p^\circ \in \mathrm{dom}\, g$, it iteratively performs the following steps: find a steepest direction d by solving a local optimization problem (Step 3), compute a step length $\lambda \geq 1$ (Step 6),[3] and update a current solution p by adding λd to p (Step 7). If slope $g'(p; d)$ in some steepest direction d is zero (Step 4), p is ensured to be optimal (due to the L-/L$^\natural$-convexity of g). In short, Algorithm 1 minimizes an L-/L$^\natural$-convex function g by iteratively solving local optimization problems in Step 3.

A remarkable property of Algorithm 1 is that the number of iterations is bounded by the distance between an initial point p° and an optimal solution $p^* \in \mathrm{argmin}\, g$ closest to p°. We introduce some definitions to describe this property more precisely. For any $p \in \mathbb{R}^V$, we define the $\ell_\infty^\pm$*-norm* as

$$\|p\|_\infty^\pm = \max_{i \in V} \max\{0, +p_i\} + \max_{i \in V} \max\{0, -p_i\},$$

which satisfies the axioms of norms. For any L-/L$^\natural$-convex $g : \mathbb{Z}^V \to \mathbb{R} \cup \{+\infty\}$, we define $\mu(\cdot; g) : \mathbb{Z}^V \to \mathbb{Z}_{\geq 0}$ as a function that returns the $\ell_\infty^\pm$-distance between input $p \in \mathbb{Z}^V$ and an optimal solution $p^* \in \mathrm{argmin}\, g$ closest to p, i.e.,

$$\mu(p; g) := \min\bigl\{\, \|p^* - p\|_\infty^\pm \bigm| p^* \in \mathrm{argmin}\, g \,\bigr\}.$$

Then, Algorithm 1 converges to an optimal solution closest to p° in $\mu(p^\circ; g) + 1$ iterations, as in the next proposition.

Proposition 2.4 ((Murota & Shioura, 2014, Theorem 1.2) and (Fujishige et al., 2015, Theorem 6.2)). *Algorithm 1 returns an optimal solution $p^* \in \mathrm{argmin}\, g$ such that $\|p^* - p^\circ\|_\infty^\pm = \mu(p^\circ; g)$ in at most $\mu(p^\circ; g) + 1$ iterations.*

Example 2. We again consider the dual LP (1) of weighted perfect bipartite matching. Since w_{ij} are integers, we can restrict the domain to $\mathbb{Z}^V$ and reduce the LP to the minimization of an L-convex function g, which is a sum of a linear objective function and the indicator function of the L-convex feasible region. As in (Sakaue & Oki, 2022, Section 3.1), we can reduce local optimization in Step 3 to a maximum cardinality matching problem. If we solve it with the $\mathrm{O}(m\sqrt{n})$-time Hopcroft–Karp algorithm, Algorithm 1 runs in $\mathrm{O}(m\sqrt{n}\cdot\mu(p^\circ; g))$ time, which can be faster than the $\mathrm{O}(mn)$-time Hungarian method if $\mu(p^\circ; g)$ is small. Indeed, Algorithm 1 with a fixed feasible p° closely resembles the Hungarian method (see (Schrijver, 2003, Section 18.5b)).

For later use, we also define $\bar{\mu}(\cdot; g) : \mathbb{R}^V \to \mathbb{R}_{\geq 0}$ as

$$\bar{\mu}(p; g) := \min\bigl\{\, \|p^* - p\|_\infty^\pm \bigm| p^* \in \mathrm{conv}(\mathrm{argmin}\, g) \,\bigr\},$$

which is a continuous extension of $\mu(\cdot; g)$ and is helpful in benefiting from real-valued predictions. Note that using $\bar{\mu}$ instead of μ only strengthens time complexity bounds since $\bar{\mu}(p; g) \leq \mu(p; g)$ for all $p \in \mathbb{Z}^V$. Indeed, $\bar{\mu}(p; g) = \mu(p; g)$ holds for all $p \in \mathbb{Z}^V$, which we can prove by confirming the existence of integral $p^* \in \mathrm{conv}(\mathrm{argmin}\, g)$ that attains the minimum $\ell_\infty^\pm$-distance. See Appendix B for the proof.

Lemma 2.5. *Let $g : \mathbb{Z}^V \to \mathbb{R} \cup \{+\infty\}$ be an L-/L$^\natural$-convex function. For every $p \in \mathbb{Z}^V$, it holds that $\mu(p; g) = \bar{\mu}(p; g)$.*

3. Time Complexity Bound

We give an improved prediction-dependent time complexity bound for L-/L$^\natural$-convex minimization. As with (Dinitz et al., 2021; Sakaue & Oki, 2022), we decompose our framework into three phases: (i) converting a prediction $\hat{p} \in \mathbb{R}^V$ into an initial feasible solution $p^\circ \in \mathbb{Z}^V$, (ii) solving a problem with an algorithm warm-started by p°, and (iii) learning predictions $\hat{p}$. The following theorem gives formal guarantees to phases (i) and (ii), and Section 4 studies phase (iii).

Theorem 3.1. *Let $g : \mathbb{Z}^V \to \mathbb{R} \cup \{+\infty\}$ be an L-/L$^\natural$-convex function and $\hat{p} \in \mathbb{R}^V$ a possibly infeasible prediction.*

(i) *If we can compute an $\ell_\infty^\pm$-projection $\hat{q}$ of the prediction $\hat{p}$ onto $\mathrm{conv}(\mathrm{dom}\, g)$, defined by*

$$\hat{q} \in \mathrm{argmin}\bigl\{\, \|q - \hat{p}\|_\infty^\pm \bigm| q \in \mathrm{conv}(\mathrm{dom}\, g) \,\bigr\}, \tag{3}$$

in T_{prj} time, we can obtain an initial feasible solution $p^\circ = \lfloor \hat{q} \rceil \in \mathrm{dom}\, g$ in $\mathrm{O}(T_{\mathrm{prj}} + |V|)$ time.

(ii) *Algorithm 1 starting from $p^\circ = \lfloor \hat{q} \rceil$ returns an optimal solution to $\min_{p \in \mathbb{Z}^V} g(p)$ in $2\bar{\mu}(\hat{p}; g) + 2$ iterations. Thus, if we can solve local optimization in Step 3 in T_{loc} time, Algorithm 1 runs in $\mathrm{O}(T_{\mathrm{loc}} \cdot \bar{\mu}(\hat{p}; g))$ time.*

[3] Step 6 computes a so-called *long-step* λ (Fujishige et al., 2015; Shioura, 2017). If computing λ is costly, we can instead set $\lambda \leftarrow 1$ without affecting Proposition 2.4 and the subsequent analysis.

Proof. The claim of (i) is identical to that of (Sakaue & Oki, 2022, Theorem 1), and so is its proof. We below prove the claim of (ii) by modifying their original proof.

Proposition 2.4 states that Algorithm 1 finds an optimal solution in $\mu(p^\circ; g) + 1$ iterations. Hence, the second claim holds if $\mu(p^\circ; g) \le 2\bar{\mu}(\hat{p}; g) + 1$, which is proved as follows.

For the given prediction $\hat{p}$, take any $p^* \in \mathrm{conv}(\mathrm{argmin}\, g)$ that attains $\|p^* - \hat{p}\|_\infty^\pm = \bar{\mu}(\hat{p}; g)$. Note that we have

$$\mu(p^\circ; g) = \bar{\mu}(p^\circ; g) \le \|p^* - p^\circ\|_\infty^\pm,$$

where the equality is due to Lemma 2.5 with $p^\circ = \lfloor \hat{q} \rceil \in \mathbb{Z}^V$ and the inequality comes from the definition of $\bar{\mu}(\cdot; g)$ and $p^* \in \mathrm{conv}(\mathrm{argmin}\, g)$. From $p^\circ = \lfloor \hat{q} \rceil$ and the fact that rounding changes each entry up to $\pm 1/2$, we have

$$\|p^* - p^\circ\|_\infty^\pm \le \|p^* - \hat{q}\|_\infty^\pm + 1.$$

Furthermore, the triangle inequality implies

$$\|p^* - \hat{q}\|_\infty^\pm \le \|p^* - \hat{p}\|_\infty^\pm + \|\hat{q} - \hat{p}\|_\infty^\pm.$$

Here, we have $\|p^* - \hat{p}\|_\infty^\pm = \bar{\mu}(\hat{p}; g)$ due to the choice of p^*. Also, $\|\hat{q} - \hat{p}\|_\infty^\pm \le \|p^* - \hat{p}\|_\infty^\pm = \bar{\mu}(\hat{p}; g)$ holds since $\hat{q}$ is defined as in (3) and $p^* \in \mathrm{conv}(\mathrm{argmin}\, g) \subseteq \mathrm{conv}(\mathrm{dom}\, g)$. Thus, we obtain $\mu(p^\circ; g) \le 2\bar{\mu}(\hat{p}; g) + 1$. □

Theorem 3.1 states that, given a prediction $\hat{p} \in \mathbb{R}^V$, we can solve $\min_{p \in \mathbb{Z}^V} g(p)$ in $\mathrm{O}(T_{\mathrm{prj}} + |V| + T_{\mathrm{loc}} \cdot \bar{\mu}(\hat{p}; g))$ time. Furthermore, it holds that $T_{\mathrm{prj}} + |V| \le T_{\mathrm{loc}}$ in most cases, including all the problems listed in Table 1 (see (Sakaue & Oki, 2022, Section 3)). In such cases, our time complexity bound reduces to $\mathrm{O}(T_{\mathrm{loc}} \cdot \bar{\mu}(\hat{p}; g))$, and we can obtain the results in Table 1 by substituting the running time of local optimization solvers into T_{loc}. For example, in the bipartite-matching case, we can solve local optimization (maximum cardinality matching) with the Hopcroft–Karp algorithm in $T_{\mathrm{loc}} = \mathrm{O}(m\sqrt{n})$ time, thus obtaining the $\mathrm{O}(m\sqrt{n} \cdot \bar{\mu}(\hat{p}; g))$-time bound in Table 1. For T_{loc} of the other problems, see (Sakaue & Oki, 2022, Sections 3.2 and 3.3). Note that our bounds in Table 1 are at least as good as those of (Sakaue & Oki, 2022) up to constant factors since we have $\bar{\mu}(\hat{p}; g) \le \|p^* - \hat{p}\|_\infty^\pm \le 2\|p^* - \hat{p}\|_\infty$ for any $p^* \in \mathrm{conv}(\mathrm{argmin}\, g)$.

4. Learning Predictions

We now discuss how to learn predictions $\hat{p} \in \mathbb{R}^V$. Following (Khodak et al., 2022; Sakaue & Oki, 2022), we mainly study the online learning setting, where L-/L$^\natural$-convex functions g_t for $t = 1, \dots, T$ are chosen adversarially. We apply the online gradient descent method (OGD) to online minimization of $\bar{\mu}(\cdot; g_t)$ and prove its regret bound. We then obtain a sample complexity bound via online-to-batch conversion. To begin with, we briefly overview basics of OGD.

4.1. Basics of Online Gradient Descent

Let $f_t : \mathbb{R}^V \to \mathbb{R}$ be the tth loss for each $t = 1, \dots, T$. We consider the following standard OGD: starting from $\hat{p}_1 = \mathbf{0}$, in each tth round, play $\hat{p}_t$, observe f_t, compute $z_t \in \partial f_t(\hat{p}_t)$, and set $\hat{p}_{t+1} \leftarrow \Pi_C(\hat{p}_t - \eta z_t)$, where $\eta > 0$ is a learning rate and Π_C is the ℓ_2-projection onto $[-C, +C]^V$. This OGD enjoys the following regret bound.

Proposition 4.1 (Orabona (2020, Section 2.2))**.** *Let $C > 0$ and $f_1, \dots, f_T$ be an arbitrary sequence of convex functions from $\mathbb{R}^V$ to $\mathbb{R}$. If OGD uses subgradients $z_t \in \partial f_t(\hat{p}_t)$ such that $\|z_t\|_2 \le L$ for $t = 1, \dots, T$ and a learning rate of $\eta = \frac{C}{L}\sqrt{\frac{n}{T}}$, it returns $\hat{p}_1, \dots, \hat{p}_T$ satisfying*

$$\sum_{t=1}^{T} f_t(\hat{p}_t) \le \min_{\hat{p}^* \in [-C,+C]^V} \sum_{t=1}^{T} f_t(\hat{p}^*) + CL\sqrt{nT}.$$

4.2. Main Results

Our basic idea is to regard $f_t(\hat{p}) = \bar{\mu}(\hat{p}; g_t)$ as the tth loss for $t = 1, \dots, T$ and use the above OGD. We first confirm the convexity of the loss functions.

Lemma 4.2. *$f_t(\hat{p}) = \bar{\mu}(\hat{p}; g_t)$ is convex in $\hat{p} \in \mathbb{R}^V$.*

Proof. Let $S = \mathrm{conv}(\mathrm{argmin}\, g_t)$. We can rewrite $f_t(\hat{p}) = \bar{\mu}(\hat{p}; g_t) = \min\{\, \|p^* - \hat{p}\|_\infty^\pm \mid p^* \in S \,\}$ as

$$f_t(\hat{p}) = \inf\left\{\, \|p^* - \hat{p}\|_\infty^\pm + \delta_S(p^*) \;\middle|\; p^* \in \mathbb{R}^V \,\right\},$$

where $\delta_S : \mathbb{R}^V \to \{0, +\infty\}$ is the indicator function of a convex set S. Also, $\|\cdot\|_\infty^\pm$ is convex by the triangle inequality. Thus, $f_t(\hat{p})$ is the infimal convolution of convex functions, hence convex (Rockafellar, 1970, Theorem 5.4). □

Our goal is to show that OGD applied to $\bar{\mu}(\hat{p}; g_t)$ enjoys a regret upper bound and runs in polynomial time as follows.

Theorem 4.3. *Let $C > 0$. For an arbitrary sequence of L-/L$^\natural$-convex functions, $g_1, \dots, g_T$, from $\mathbb{Z}^V$ to $\mathbb{R} \cup \{+\infty\}$, OGD computes predictions $\hat{p}_1, \dots, \hat{p}_T$ that satisfy*

$$\sum_{t=1}^{T} \bar{\mu}(\hat{p}_t; g_t) \le \min_{\hat{p}^* \in [-C,+C]^V} \sum_{t=1}^{T} \bar{\mu}(\hat{p}^*; g_t) + C\sqrt{2nT}.$$

In each round t, if an inequality system of $\mathrm{conv}(\mathrm{argmin}\, g_t)$ can be obtained in T_{ineq} time (as in Assumption 4.6) and $p_t^* \in \mathrm{argmin}\, g_t$ is given, OGD takes $T_{\mathrm{ineq}} + \mathrm{O}(n^2)$ time.

Note that the regret bound in terms of $\bar{\mu}(\hat{p}; g_t)$ is the main difference from the previous studies, which consider simpler functions of the form $\|p_t^* - \hat{p}\|$ with some fixed optimal p_t^*. Our regret bound is as small as that of (Sakaue & Oki, 2022) even though we consider more involved functions, $\bar{\mu}(\cdot; g_t)$, and is indeed asymptotically tight as shown in Appendix D.

To prove Theorem 4.3, it suffices to show how to compute a subgradient z_t of $f_t(\hat{p}_t) = \bar{\mu}(\hat{p}_t; g_t)$ such that $\|z_t\|_2 \le L = \sqrt{2}$ in $T_{\mathrm{ineq}} + \mathrm{O}(n^2)$ time, which we present in Section 4.3.

We show in Section 5.2 that T_{ineq} is polynomial even when we only have black-box access to g_t. Moreover, Section 5.1 shows that T_{ineq} can be much smaller for the specific problems listed in Table 1. The assumption that $p_t^* \in \operatorname{argmin} g_t$ is available usually holds since we learn $\hat{p}_t$ after solving the tth instance, $\min_{p \in \mathbb{Z}^V} g_t(p)$. (If not, we may solve the tth instance with standard polynomial algorithms; then OGD runs in polynomial time.) In the bipartite-matching case, under those assumptions, OGD will turn out to take only $\mathrm{O}(m + n \log n)$ time per round (see Section 5.1), which is even faster than a single local optimization step in Algorithm 1, or the $\mathrm{O}(m\sqrt{n})$-time Hopcroft–Karp algorithm. Therefore, although our learning method is generally slower than the previous ones, it is usually not a serious drawback.

Given Theorem 4.3, we can obtain a sample complexity bound via online-to-batch conversion. The proof is almost identical to those of (Khodak et al., 2022; Sakaue & Oki, 2022) and thus deferred to Appendix C.

Corollary 4.4. *Let $\mathcal{D}$ be an (unknown) distribution over L-/L$^\natural$-convex functions $g : \mathbb{Z}^V \to \mathbb{R} \cup \{+\infty\}$, $\delta \in (0, 1]$, and $\varepsilon > 0$. Given $T = \Omega\left(\left(\frac{C}{\varepsilon}\right)^2 \left(n + \log \frac{1}{\delta}\right)\right)$ i.i.d. draws of $g_1, \dots, g_T \sim \mathcal{D}$, we can obtain $\hat{p} \in \mathbb{R}^V$ that satisfies*

$$\mathbb{E}_{g \sim \mathcal{D}}[\bar{\mu}(\hat{p}; g)] \le \min_{\hat{p}^* \in [-C, +C]^V} \mathbb{E}_{g \sim \mathcal{D}}[\bar{\mu}(\hat{p}^*; g)] + \varepsilon$$

with probability at least $1 - \delta$. Under the assumptions of Theorem 4.3, we can compute $\hat{p}$ in $\mathrm{O}(T \cdot (T_{\mathrm{ineq}} + n^2))$ time.

Remark 4.5. We can usually bound C with instance parameters. For example, if edge weights of bipartite-matching instances are always in $[-W, +W]$, $C = nW$ is large enough to ensure that an optimal prediction $\hat{p}^*$ is in $[-C, +C]^V$. See (Sakaue & Oki, 2022, Section 4) for more information.

4.3. Computation of Subgradients

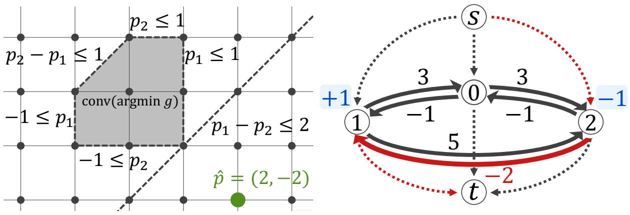
We below omit t and let, e.g., $g = g_t$ and $\hat{p} = \hat{p}_t$ for brevity since this section focuses only on the tth round.

First, we detail the assumption in Theorem 4.3. Recall that $\operatorname{argmin} g$ is an L-/L$^\natural$-convex set due to (Murota, 2003, Theorem 7.17). Therefore, $\operatorname{conv}(\operatorname{argmin} g)$ has an inequality-system representation as in Proposition 2.2. In this section, we assume one such inequality system to be available.

Assumption 4.6. We can obtain an inequality-system representation of $\operatorname{conv}(\operatorname{argmin} g) \subseteq \mathbb{R}^V$ of the form

$$\left\{ p \in \mathbb{R}^V \;\middle|\; \begin{array}{l} \alpha_i \le p_i \le \beta_i \text{ for } i \in V, \\ p_j - p_i \le \gamma_{ij} \text{ for distinct } i, j \in V \end{array} \right\} \tag{4}$$

in T_{ineq} time, where $-\alpha_i, \beta_i, \gamma_{ij} \in \mathbb{Z} \cup \{+\infty\}$.



(a) $\operatorname{conv}(\operatorname{argmin} g)$

(b) $(\tilde{V}, \tilde{E})$ and a shortest path

Figure 2: If $\operatorname{conv}(\operatorname{argmin} g)$ is given by inequalities in (a) and $\hat{p} = (2, -2)$, we can compute $\bar{\mu}(\hat{p}; g)$ by solving the shortest path problem in $(\tilde{V}, \tilde{E})$ as in (b), where weights of dashed edges are zero and the others have weights $\tilde{w}_{ij}(\hat{p})$ shown nearby edges. A shortest path $P^* = \{s2, 21, 1t\}$ is shown in red, and the negative of its total weight is equal to $\bar{\mu}(\hat{p}; g) = 2$. We can obtain a subgradient, $-\nabla\phi(\hat{p}; P^*) = (+1, -1)$, as shown in blue in (b). If we replace a redundant inequality constraint, $p_1 - p_2 \le 2$, in (a) with $p_1 - p_2 \le +\infty$, the edge from 2 to 1 is removed in (b); still, the other shortest path, $\{s2, 20, 01, 1t\}$, yields the same subgradient.

Remark 4.7. Although inequality-system representations of $\operatorname{conv}(\operatorname{argmin} g)$ are not unique, whichever of the form (4) works in the following discussion. If an inequality system at hand lacks inequalities for some $i, j \in V$, we suppose those with $\alpha_i = -\infty$, $\beta_i = +\infty$, and $\gamma_{ij} = +\infty$ to be given; we always apply this treatment to all α_i and β_i if g is L-convex since $\operatorname{argmin} g$ has no box constraints (see Proposition 2.2).

We then observe that computing the value of $\bar{\mu}(\hat{p}; g)$ for any given $\hat{p} \in \mathbb{R}^V$ can be reduced to a shortest path problem in a directed graph with possibly negative weights. Since the reduction is presented in (Sakaue & Oki, 2022, Appendix D), we here only give a brief description for later convenience.

Let $E = \{ ij \mid i, j \in V; i \ne j \}$ and $V_0 = \{0\} \cup V$. We use $\tilde{V} = V_0 \cup \{s, t\}$ as a vertex set, where s is the origin and t is the destination. We define a set $\tilde{E}$ of directed edges as

$$E \cup \{\{0\} \times V\} \cup \{V \times \{0\}\} \cup \{\{s\} \times V_0\} \cup \{V_0 \times \{t\}\}.$$

Given any $\hat{p} \in \mathbb{R}^V$, we define weights of edges $ij \in \tilde{E}$ as

$$\tilde{w}_{ij}(\hat{p}) = \begin{cases} \gamma_{ij} - \hat{p}_j + \hat{p}_i & \text{if } ij \in E, \\ -\alpha_i + \hat{p}_i & \text{if } i \in V \text{ and } j = 0, \\ \beta_j - \hat{p}_j & \text{if } i = 0 \text{ and } j \in V, \\ 0 & \text{if } i = s \text{ or } j = t, \end{cases} \tag{5}$$

where α_i, β_j, γ_{ij} are those representing $\operatorname{conv}(\operatorname{argmin} g)$ as in (4). We take $ij \in \tilde{E}$ to be removed if $\tilde{w}_{ij} = +\infty$.

Note that the negative weights, $-\tilde{w}_{ij}(\hat{p})$, for $ij \in V_0 \times V_0$ indicate how much $\hat{p}$ violates the corresponding inequalities in (4) representing $\operatorname{conv}(\operatorname{argmin} g)$. From this fact, we can show that the negative of the total weight of a shortest

s–t path in $(\tilde{V}, \tilde{E})$ is equal to $\bar{\mu}(\hat{p}; g)$, or how far $\hat{p}$ is from $\mathrm{conv}(\mathrm{argmin}\, g)$ in terms of the $\ell_\infty^\pm$-norm (see (Sakaue & Oki, 2022, Appendix D)). Figure 2 illustrates an example of $\mathrm{conv}(\mathrm{argmin}\, g)$ and the shortest path problem for computing $\bar{\mu}(\hat{p}; g)$. Note that $(\tilde{V}, \tilde{E})$ has no negative cycles; otherwise, the shortest-path weight is $-\infty$, hence $\bar{\mu}(\hat{p}; g) = +\infty$, contradicting $\mathrm{argmin}\, g \neq \emptyset$ (Assumption 2.1). Also, the shortest-path weight is always non-positive since there always exist zero-weight s–t paths $\{si, it\}$ for $i \in V_0$.

We then rewrite $\bar{\mu}(\hat{p}; g)$ keeping the reduction to the shortest path problem in mind. Let $\mathcal{P} \subseteq 2^{\tilde{E}}$ be the set of all simple s–t paths. For each $P \in \mathcal{P}$, define $\phi(\cdot; P) : \mathbb{R}^V \to \mathbb{R}$ by

$$\phi(\hat{p}; P) \coloneqq \sum_{ij \in P} \tilde{w}_{ij}(\hat{p}), \tag{6}$$

which equals the total weight of an s–t path P. Since $\bar{\mu}(\hat{p}; g)$ is the negative of the total weight of a shortest path, we have

$$\bar{\mu}(\hat{p}; g) = \max\{ -\phi(\hat{p}; P) \mid P \in \mathcal{P} \}. \tag{7}$$

Since each $-\phi(\hat{p}; P)$ is linear in $\hat{p}$ by (5) and (6), and $\mathcal{P}$ is finite (hence compact), Danskin's theorem (Danskin, 1966) (see, also (Bertsekas, 2016, Proposition B.22)) implies

$$\partial \bar{\mu}(\hat{p}; g) = \mathrm{conv}\{ -\nabla \phi(\hat{p}; P^*) \mid P^* \in \mathcal{P}(\hat{p}) \},$$

where $\mathcal{P}(\hat{p}) \coloneqq \mathrm{argmax}\{ -\phi(\hat{p}; P) \mid P \in \mathcal{P} \}$ is the set of all the shortest s–t paths when $\hat{p} \in \mathbb{R}^V$ is given. Therefore, we can compute a subgradient of $\bar{\mu}(\cdot; g)$ at $\hat{p}$ by finding a shortest s–t path $P^* \in \mathcal{P}(\hat{p})$ and calculating

$$-\nabla \phi(\hat{p}; P^*) = -\sum_{ij \in P^*} \nabla \tilde{w}_{ij}(\hat{p}).$$

We then take a closer look at the subgradient $-\nabla \phi(\hat{p}; P^*)$. From (5), each $-\nabla \tilde{w}_{ij}(\hat{p})$ has at most one -1 and one $+1$. These non-zeros are canceled out by taking the summation along the shortest path P^*, except for at most two non-zeros, -1 and $+1$, corresponding to the two vertices adjacent to s and t in P^*, respectively; if s and/or t are adjacent to $0 \in V_0$, the corresponding non-zeros also vanish. See Figure 2b for an illustration of how $-\nabla \phi(\hat{p}; P^*)$ is calculated.

Formally, if the first and last edges in a shortest path $P^* \in \mathcal{P}(\hat{p})$ are si and jt, respectively, with $i \neq j$, a subgradient $-\nabla \phi(\hat{p}; P^*) \in \partial \bar{\mu}(\hat{p}; g)$ can be written as

$$\big(0 \dots 0 \quad \overset{i\text{th}}{-\mathbb{1}_{i \neq 0}} \quad 0 \dots 0 \quad \overset{j\text{th}}{\mathbb{1}_{j \neq 0}} \quad 0 \dots 0\big), \tag{8}$$

where $\mathbb{1}_{k \neq 0} = 1$ if $k \neq 0$ and 0 otherwise; if $i = j$, the subgradient is zero. To conclude, we obtain the next lemma.

Lemma 4.8. *If an inequality system of* $\mathrm{conv}(\mathrm{argmin}\, g)$ *and* $p^* \in \mathrm{argmin}\, g$ *are available, we can compute a subgradient* $z \in \partial \bar{\mu}(\hat{p}; g)$ *with* $\|z\|_2 \leq \sqrt{2}$ *in* $\mathrm{O}(n^2)$ *time.*

Proof. Given a shortest s–t path $P^* \in \mathcal{P}(\hat{p})$, we can compute $z \in \partial \bar{\mu}(\hat{p}; g)$ as in (8), which satisfies $\|z\|_2 \leq \sqrt{2}$. To obtain P^*, we first transform the possibly negative edge weights (5) into non-negative ones that preserve the shortest-path set $\mathcal{P}(\hat{p})$ via a *potential* (see Appendix E for details). We can do this transformation in $\mathrm{O}(|\tilde{E}|)$ time by using an optimal solution $p^* \in \mathrm{argmin}\, g$, which is assumed to be given in Theorem 4.3. Therefore, by finding a shortest s–t path P^* with Dijkstra's algorithm in $\mathrm{O}(|\tilde{E}| + |\tilde{V}| \log |\tilde{V}|) \lesssim \mathrm{O}(n^2)$ time, we can compute a subgradient in $\mathrm{O}(n^2)$ time. □

We can easily obtain an intuition of the subgradient (8) when a shortest s–t path, $P^* \in \mathcal{P}(\hat{p})$, is of the form $\{si, ij, jt\}$. Such a shortest path implies that a current prediction $\hat{p}$ violates inequality constraint $p_j - p_i \leq \gamma_{ij}$ most largely among those in (4) that represent $\mathrm{conv}(\mathrm{argmin}\, g)$. Updating $\hat{p}$ along the negative direction of the subgradient (8) reduces the magnitude of the violation, $\hat{p}_j - \hat{p}_i - \gamma_{ij} > 0$, by increasing $\hat{p}_i$ and decreasing $\hat{p}_j$. Thus, the subgradient descent moves a current prediction closer to $\mathrm{conv}(\mathrm{argmin}\, g)$.

At a high level, our strategy is to write the distance to the set of optimal solutions as a maximum of linear (or convex) functions, as in (7), and use Danskin's theorem to obtain a subgradient. Then, we can use OGD to learn predictions close to sets of optimal solutions. We expect that this simple idea is also useful in other settings, e.g., (Chen et al., 2022).

5. Obtaining Inequality Systems of Minimizers

We show how to get an inequality system of $\mathrm{conv}(\mathrm{argmin}\, g)$ as in Assumption 4.6. Section 5.1 provides efficient methods that utilize problem-specific structures, and Section 5.2 presents a general polynomial-time method that only uses black-box access to g, implying T_{ineq} is at most polynomial.

5.1. Efficient Problem-Specific Methods

We can efficiently construct a desired inequality system if the primal-dual structure of the problem, $\min_{p \in \mathbb{Z}^V} g(p)$, is available. We detail this method for bipartite matching.

We consider the weighted perfect bipartite matching problem introduced in Section 1. Let (V, E) be a bipartite graph with equal-sized bipartition $V = L \cup R$, weights $w \in \mathbb{Z}^E$, $n = |V|$, and $m = |E|$. Recall that we can write the dual LP as in (1) with constraints $s_i - t_j \geq w_{ij}$ for $ij \in E$. The following complementarity theorem gives a useful characterization of the set of dual optimal solutions.

Proposition 5.1 (Murota (1995, Proposition 2.3))**.** *Let* $M \subseteq E$ *be a matching in* (V, E) *and* $p = (s, t) \in \mathbb{Z}^{L \cup R}$ *a dual feasible solution to* (1)*. Then,* M *and* p *are optimal if and only if* $s_i - t_j = w_{ij}$ *for all* $ij \in M$*.*

This proposition implies that given an arbitrary maximum weight matching $M^* \subseteq E$, we can represent the set of dual optimal solutions, $\operatorname{argmin} g$, by an inequality system as

$$\left\{ p = (s,t) \in \mathbb{Z}^{L \cup R} \;\middle|\; \begin{array}{l} s_i - t_j \geq w_{ij} \text{ for } ij \in E, \\ s_i - t_j \leq w_{ij} \text{ for } ij \in M^* \end{array} \right\},$$

and replacing $\mathbb{Z}^{L \cup R}$ with $\mathbb{R}^{L \cup R}$ yields an inequality-system representation of $\operatorname{conv}(\operatorname{argmin} g)$ (see Proposition 2.2). Note that a maximum weight matching M^* is usually available for free since the tth instance is already solved when learning $\hat{p}_t$. Once M^* is given, we can construct the above inequality system in $\mathrm{O}(m)$ time, hence $T_{\text{ineq}} = \mathrm{O}(m)$.

Having seen T_{ineq} is small enough, the dominant part in the per-round time complexity of OGD is Dijkstra's algorithm for computing a subgradient, which runs in $\mathrm{O}(m + n \log n)$ time since the above inequality system leads to graph $(\tilde{V}, \tilde{E})$ with $|\tilde{E}| = \mathrm{O}(m)$. Hence, OGD's per-round running time is shorter than that of solving local optimization in Algorithm 1 once with the $\mathrm{O}(m\sqrt{n})$-time Hopcroft–Karp algorithm.

The core idea of the above method is to utilize the "if and only if" condition of the complementarity theorem. In other words, once we find an arbitrary primal (dual) optimal solution, we can capture the set of all dual (primal) optimal solutions via the complementarity condition. This idea has been well studied in *combinatorial relaxation* (Murota, 1995) and is applicable to matroid intersection and discrete energy minimization. Below, we only present the results due to the space limitation; see Appendices F.1 and F.2, respectively.

Theorem 5.2. *Consider the dual problem,* $\min_{p \in Z^V} g(p)$*, of weighted matroid intersection defined on a ground set* V *of size* n*. If a maximum weight common base is available (or the problem is already solved), we can obtain an inequality system of the form* (4) *representing* $\operatorname{conv}(\operatorname{argmin} g)$ *in* $T_{\text{ineq}} = \mathrm{O}(\tau n r)$ *time, where* r *is the rank of the matroids and* τ *is the running time of independence oracles.*

Theorem 5.3. *Consider discrete energy minimization,* $\min_{p \in Z^V} g(p)$*, defined on a graph with* n *vertices and* m *edges. If* $\operatorname{dom} g \subseteq [0, W]^V$ *for some* $W > 0$ *(which is true in most computer-vision applications), we can obtain an inequality system of the form* (4) *representing* $\operatorname{conv}(\operatorname{argmin} g)$ *in* $T_{\text{ineq}} = \mathrm{O}(mn \log(n^2/m) \log(nW))$ *time.*

5.2. General Polynomial-Time Method

We then discuss L-/L$^\natural$-convex minimization, $\min_{p \in \mathbb{Z}^V} g(p)$, where we only have black-box access to g values. Unlike the above cases, this setting does not enjoy useful primal-dual structures. Still, we can construct a desired inequality system in polynomial time. We here assume $\operatorname{argmin} g \cap [-C, +C]^V \neq \emptyset$, where $C > 0$ is the constant used in OGD, to deal with possibly unbounded $\operatorname{argmin} g$. This condition is reasonable since the best prediction, $\hat{p}^*$, in Theorem 4.3 is selected from $[-C, +C]^V$. We also assume g to have a finite minimum value. Under these assumptions, the following theorem holds (see Appendix F.3 for the complete proof).

Theorem 5.4. *For general L-/L$^\natural$-convex function minimization,* $\min_{p \in Z^V} g(p)$*, such that* $\operatorname{argmin} g \cap [-C, +C]^V \neq \emptyset$ *and* $\min g > -\infty$*, we can obtain an inequality system of a subset of* $\operatorname{conv}(\operatorname{argmin} g)$ *that is sufficient for the subgradient computation in* $T_{\text{ineq}} = \mathrm{O}(n^2 \log^2 C \cdot (\mathrm{EO} \cdot n^3 \log^2 n + n^4 \log^{\mathrm{O}(1)} n))$ *time, where* EO *is the time for evaluating* g*.*

Proof sketch. Note that $\operatorname{argmin} g$ can be written with $\mathrm{O}(n^2)$ inequalities due to Proposition 2.2. We seek appropriate values of all the $\mathrm{O}(n^2)$ constants, $\alpha_i, \beta_i, \gamma_{ij} \in \mathbb{Z}$, via binary search, each of which takes $\mathrm{O}(\log C)$ iterations by the assumption of $\operatorname{argmin} g \cap [-C, +C]^V \neq \emptyset$. In each iteration, we check whether a given inequality, e.g., $p_j - p_i \leq \gamma_{ij}$, is satisfied by all relevant minimizers of g or not. Based on the steepest descent scaling algorithm (Murota, 2003, Section 10.3.2), we can check this by solving submodular function minimization (defined as with local optimization in Step 3 of Algorithm 1) $\mathrm{O}(\log C)$ times. If we solve it with an $\mathrm{O}(\mathrm{EO} \cdot n^3 \log^2 n + n^4 \log^{\mathrm{O}(1)} n)$-time algorithm of (Lee et al., 2015), we obtain the desired time complexity. □

6. Experiments

We conducted experiments on random weighted bipartite matching instances to compare our learning method with the previous methods (Dinitz et al., 2021; Sakaue & Oki, 2022). The source code is available at https://github.com/ssakaue/alps-l-convex-optset-code.

We generated random instances with $n = 10$ and 100 as follows. Let (L, R) be a bipartition of a vertex set V such that $L = \{1, 2, \dots, \frac{n}{2}\}$ and $R = \{\frac{n}{2} + 1, \frac{n}{2} + 2, \dots, n\}$. First, we created edges $(i, i + |L|) \in L \times R$ for $i = 1, \dots, |L|$ with a weight of 1 to ensure that there always exists at least one perfect matching. Then, for the other pairs of $(i, j) \in L \times R$, we let $w_{ij} = \lfloor \frac{100}{n^2} \times i \times (j - |L|) \rfloor + u$, where u is a noise term drawn uniformly at random from $[-\sigma, +\sigma] \cap \mathbb{Z}$ for some $\sigma > 0$. If $w_{ij} > 0$, we created edges $(i, j) \in L \times R$ with weights w_{ij}. Hence, predictions should be learned to match $i \in L$ and $j \in R$ such that both i and j are large, while $i \in L$ and $i + |L| \in R$ should not be matched since $(i, i + |L|)$ only has an edge weight of 1. We thus created a dataset of $T = 1000$ random weighted bipartite graphs. We repeated this procedure 10 times to obtain 10 independent datasets, with which we calculated the mean and standard deviation of the results. We created such datasets for various noise strengths $\sigma \in \{1, 5, 10, 20\}$.

We consider the online setting where the random weighted bipartite graphs arrive sequentially for $t = 1, \dots, T$. Each method learns predictions $\hat{p}_t \in \mathbb{R}^V$ for $t = 1, \dots, T$ with OGD, where the loss function f_t differs among the meth-

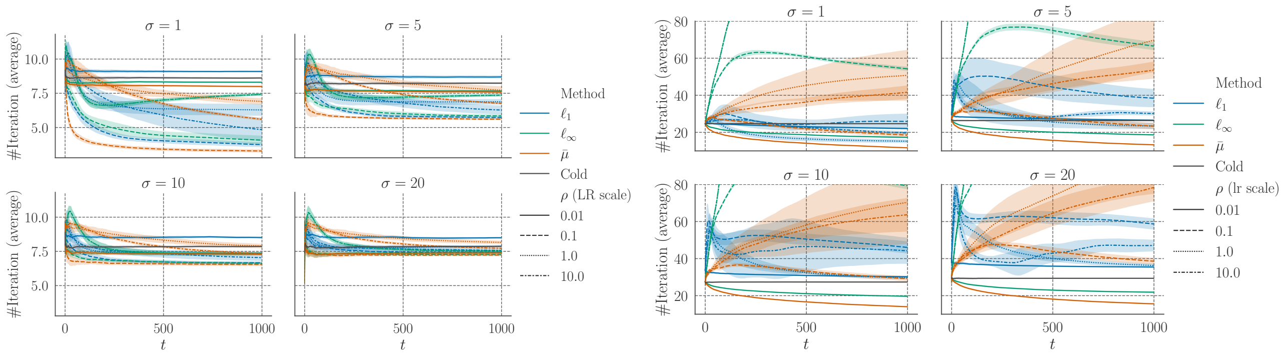


Figure 3: The average number of iterations of Algorithm 1 warm-started with predictions learned by each method for bipartite matching instances with $n = 10$ (left) and 100 (right). σ represents the noise strength and ρ is the scaling factor of the learning rate (LR) of OGD. The error band indicates the standard deviation over 10 independently random datasets.

ods, as described shortly. To improve the empirical performance, we used the following refined variant of OGD: we used the anytime online-to-batch scheme (Cutkosky, 2019) for the last iterate convergence of each tth prediction (see Appendix C for details) and an adaptive learning rate of $\eta_t = \frac{C\sqrt{2n}}{\sqrt{\sum_{t'=1}^{t}\|z_{t'}\|_2^2}}$ (Streeter & Brendan McMahan, 2010) in each tth iteration of OGD, where z_t is the tth subgradient, $C = nW$, and W is the largest edge weight in a dataset (as discussed in Remark 4.5). Furthermore, since OGD's performance was sensitive to the scale of learning rates, we used rescaled learning rates $\rho \times \eta_t$ for $\rho \in \{0.01, 0.1, 1.0, 10.0\}$.

We compared methods with three types of loss functions: $f_t(\hat{p}) = \|p_t^* - \hat{p}\|_1$ (Dinitz et al., 2021), $f_t(\hat{p}) = \|p_t^* - \hat{p}\|_\infty$ (Sakaue & Oki, 2022), and $f_t(\hat{p}) = \bar{\mu}(\hat{p}; g_t)$ (ours), where the first two used p_t^* returned by Algorithm 1 (or the Hungarian method) warm-started by $\hat{p}_t$. We also used the cold-start method as a baseline, which always set $\hat{p}_t = \mathbf{0}$. We denote those methods by ℓ_1, ℓ_∞, $\bar{\mu}$, and Cold, respectively, for short. To convert prediction $\hat{p}_t$ into an initial feasible solution, ℓ_1 used the greedy algorithm in (Dinitz et al., 2021), while ℓ_∞, $\bar{\mu}$, and Cold used the $\ell_\infty^\pm$-projection and rounding, as in Theorem 3.1. We can find a specific $\ell_\infty^\pm$-projection method for bipartite matching in (Sakaue & Oki, 2022, Section 3.1).

Figure 3 shows the average number of iterations of Algorithm 1 (or the Hungarian method) warm-started with predictions $\hat{p}_t$ learned by each method, where the average is taken over the past t instances for $t = 1, \dots, T$. As for $n = 10$, ℓ_1, ℓ_∞, and $\bar{\mu}$ with $\rho = 0.1$ significantly outperformed Cold in the low-noise setting ($\sigma = 1$), while their advantages decrease as the noise strength σ increased, as is also observed in (Dinitz et al., 2021, Section 4). In every case, our $\bar{\mu}$ with $\rho = 0.1$ returned the best prediction, leading to the smallest number of iterations. Moreover, $\bar{\mu}$ with $\rho = 0.1$ decreased the number of iterations more quickly than the other methods, implying that it can learn good predictions from fewer sampled instances. As for $n = 100$, our $\bar{\mu}$ with $\rho = 0.01$ outperformed the other methods in every case. One curious observation with $n = 100$ is the unstable behavior of ℓ_1: it performed best with $\rho = 1.0$ for $\sigma = 1, 5$ and with $\rho = 0.01$ for $\sigma = 10, 20$; also, it failed to outperform Cold in the latter case with larger noise strengths. This is probably because learning predictions with the ℓ_1-loss has caused extreme changes in predictions, since the number of iterations of Algorithm 1 (or the Hungarian method) depends on the ℓ_∞-distance, as implied by Proposition 2.4.

7. Conclusion

We have presented a new warm-start-with-prediction framework for L-/L$^\natural$-convex minimization that provides time complexity bounds proportional to $\bar{\mu}(\hat{p}; g)$, the $\ell_\pm$-distance between a prediction $\hat{p}$ and the set, $\mathrm{conv}(\mathrm{argmin}\, g)$, of optimal solutions. Specifically, we have shown that the steepest descent method warm-started by $\hat{p}$ takes $\mathrm{O}(\bar{\mu}(\hat{p}; g))$ iterations and that we can learn $\hat{p}$ to approximately minimize $\mathbb{E}_g[\bar{\mu}(\hat{p}; g)]$ in polynomial time. At a technical level, we have shown an efficient method for computing subgradients of $\bar{\mu}(\cdot; g)$ to learn $\hat{p}$ with OGD. Our results imply the first polynomial-time learnability of predictions that can provably warm-start algorithms regardless of the non-uniqueness of optimal solutions. This implication would be significant progress in warm-starts with predictions because the non-uniqueness always exists in the broad class of L-/L$^\natural$-convex minimization, as described in Section 1 and Remark 2.3. Studying how to learn predictions with similar guarantees for other problems will be an interesting future direction.

Acknowledgements

The authors thank the anonymous reviewers for their helpful comments. This work was supported by JST ERATO Grant Number JPMJER1903 and JSPS KAKENHI Grant Number JP22K17853.

References

Agrawal, P., Balkanski, E., Gkatzelis, V., Ou, T., and Tan, X. Learning-augmented mechanism design: Leveraging predictions for facility location. In *Proceedings of the 23rd ACM Conference on Economics and Computation (EC 2022)*, pp. 497–528. ACM, 2022. ↰ p.3

Ahuja, R. K., Hochbaum, D. S., and Orlin, J. B. Solving the convex cost integer dual network flow problem. *Manage. Sci.*, 49(7):950–964, 2003. ↰ p.17

Andoni, A. and Beaglehole, D. Learning to hash robustly, guaranteed. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*, volume 162, pp. 599–618. PMLR, 2022. ↰ p.3

Arora, R., Dekel, O., and Tewari, A. Online bandit learning against an adaptive adversary: From regret to policy regret. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML 2012)*, pp. 1747–1754. Omnipress, 2012. ↰ p.12

Azar, Y., Panigrahi, D., and Touitou, N. Online graph algorithms with predictions. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2022)*, pp. 35–66. SIAM, 2022. ↰ p.3

Bamas, E., Maggiori, A., and Svensson, O. The primal-dual method for learning augmented algorithms. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, volume 33, pp. 20083–20094. Curran Associates, Inc., 2020. ↰ p.3

Bertsekas, D. P. *Nonlinear Programming*. Athena Scientific, 3rd edition, 2016. ↰ p.7

Boffa, A., Ferragina, P., and Vinciguerra, G. A learned approach to design compressed rank/select data structures. *ACM Trans. Algorithms*, 18(3):1–28, 2022. ↰ p.3

Cesa-Bianchi, N. and Lugosi, G. *Prediction, Learning, and Games*. Cambridge University Press, 2006. ↰ p.14

Cesa-Bianchi, N., Conconi, A., and Gentile, C. On the generalization ability of on-line learning algorithms. *IEEE Trans. Inf. Theory*, 50(9):2050–2057, 2004. ↰ p.13

Chen, J., Silwal, S., Vakilian, A., and Zhang, F. Faster fundamental graph algorithms via learned predictions. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*, volume 162, pp. 3583–3602. PMLR, 2022. ↰ p.3, ↰ p.7, ↰ p.12

Cutkosky, A. Anytime online-to-batch, optimism and acceleration. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, volume 97, pp. 1446–1454. PMLR, 2019. ↰ p.9, ↰ p.13

Danskin, J. M. The theory of max-min, with applications. *SIAM J. Appl. Math.*, 14(4):641–664, 1966. ↰ p.7

Dinitz, M., Im, S., Lavastida, T., Moseley, B., and Vassilvitskii, S. Faster matchings via learned duals. In *Advances in Neural Information Processing Systems (NeurIPS 2021)*, volume 34, pp. 10393–10406. Curran Associates, Inc., 2021. ↰ p.1, ↰ p.2, ↰ p.3, ↰ p.4, ↰ p.8, ↰ p.9, ↰ p.12

Eden, T., Indyk, P., Narayanan, S., Rubinfeld, R., Silwal, S., and Wagner, T. Learning-based support estimation in sublinear time. In *International Conference on Learning Representations (ICLR 2021)*, 2021. ↰ p.3

Edmonds, J. Matroids and the greedy algorithm. *Math. Program.*, 1:127–136, 1971. ↰ p.15

Ergun, J. C., Feng, Z., Silwal, S., Woodruff, D., and Zhou, S. Learning-augmented k-means clustering. In *International Conference on Learning Representations (ICLR 2022)*, 2022. ↰ p.3

Feijen, W. and Schäfer, G. Dijkstra's algorithm with predictions to solve the single-source many-targets shortest-path problem. *arXiv:2112.11927*, 2023. ↰ p.3

Frank, A. *Connections in Combinatorial Optimization*. Oxford University Press, 2011. ↰ p.16

Fujishige, S., Murota, K., and Shioura, A. Monotonicity in steepest ascent algorithms for polyhedral L-concave functions. *J. Oper. Res. Soc. Japan*, 58(2):184–208, 2015. ↰ p.4

Hazan, E. and Kale, S. Online submodular minimization. *J. Mach. Learn. Res.*, 13(93):2903–2922, 2012. ↰ p.14

Khodak, M., Balcan, M.-F., Talwalkar, A., and Vassilvitskii, S. Learning predictions for algorithms with predictions. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*, volume 35, pp. 3542–3555. Curran Associates, Inc., 2022. ↰ p.2, ↰ p.5, ↰ p.6, ↰ p.12

Kolmogorov, V. and Shioura, A. New algorithms for convex cost tension problem with application to computer vision. *Discrete Optim.*, 6(4):378–393, 2009. ↰ p.16, ↰ p.17

Lee, Y. T., Sidford, A., and Wong, S. C.-W. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS 2015)*, pp. 1049–1065. IEEE, 2015. ↰ p.8, ↰ p.17

Lindermayr, A. and Megow, N. Website for Algorithms with Predictions (ALPS). https://algorithms-with-predictions.github.io/. Accessed: 2023-05-11. ↰ p.3

Lu, P., Ren, X., Sun, E., and Zhang, Y. Generalized sorting with predictions. In *Proceedings of the 4th SIAM Symposium on Simplicity in Algorithms (SOSA 2021)*, pp. 111–117. SIAM, 2021. ↰ p.3

Lykouris, T. and Vassilvitskii, S. Competitive caching with machine learned advice. *J. ACM*, 68(4):1–25, 2021. ↰ p.3

Mitzenmacher, M. and Vassilvitskii, S. Algorithms with predictions. In *Beyond the Worst-Case Analysis of Algorithms*, pp. 646–662. Cambridge University Press, 2021. ↰ p.1

Murota, K. Computing the degree of determinants via combinatorial relaxation. *SIAM J. Comput.*, 24(4):765–796, 1995. ↰ p.7, ↰ p.8

Murota, K. *Discrete Convex Analysis*. Discrete Mathematics and Applications. SIAM, 2003. ↰ p.3, ↰ p.6, ↰ p.8, ↰ p.13, ↰ p.16, ↰ p.17

Murota, K. and Shioura, A. Exact bounds for steepest descent algorithms of L-convex function minimization. *Oper. Res. Lett.*, 42(5):361–366, 2014. ↰ p.4

Orabona, F. *A Modern Introduction to Online Learning*. OpenBU, 2020. ↰ p.5

Polak, A. and Zub, M. Learning-augmented maximum flow. *arXiv:2207.12911*, 2022. ↰ p.3, ↰ p.12, ↰ p.13

Purohit, M., Svitkina, Z., and Kumar, R. Improving online algorithms via ML predictions. In *Advances in Neural Information Processing Systems (NeurIPS 2018)*, volume 31, pp. 9684–9693. Curran Associates, Inc., 2018. ↰ p.3

Rockafellar, R. T. *Convex Analysis*. Princeton University Press, 1970. ↰ p.5

Sakaue, S. and Oki, T. Discrete-convex-analysis-based framework for warm-starting algorithms with predictions. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*, volume 35, pp. 20988–21000. Curran Associates, Inc., 2022. ↰ p.1, ↰ p.2, ↰ p.3, ↰ p.4, ↰ p.5, ↰ p.6, ↰ p.7, ↰ p.8, ↰ p.9, ↰ p.12, ↰ p.13, ↰ p.15, ↰ p.17

Schrijver, A. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003. ↰ p.4, ↰ p.14, ↰ p.16

Shioura, A. Algorithms for L-convex function minimization: Connection between discrete convex analysis and other research fields. *J. Oper. Res. Soc. Japan*, 60(3):216–243, 2017. ↰ p.4

Streeter, M. and Brendan McMahan, H. Less regret via online conditioning. *arXiv:1002.4862*, 2010. ↰ p.9

A. Details of Issues Caused by Non-uniqueness of Optimal Solutions

We detail the problem caused by ignoring the non-uniqueness of optimal solutions p^*, which arises in many problem settings, including the dual of bipartite matching and L-/L$^\natural$-convex function minimization. In short, the problem is a kind of dilemma: if we fix some optimal p^* independently of prediction $\hat{p}$, the time complexity bounds depending on $\|p^* - \hat{p}\|$ can be poor; if we select optimal p^* depending on prediction $\hat{p}$ to make $\|p^* - \hat{p}\|$ small, we cannot use existing results on the learnability of $\hat{p}$ due to the dependence of p^* on $\hat{p}$. We below detail these two types of problems.

A.1. On Fixing Optimal Solutions Independently of Predictions

If an optimal solution is uniquely associated with each input instance independently of predictions, we can rely on the existing learnability results of predictions. Hence, one may first think of using some tie-breaking rule to handle the non-uniqueness of optimal solutions. Note, however, that no tie-breaking rule can lead to essentially stronger results than ours of using the minimum distance to the set of all optimal solutions. Moreover, seemingly reasonable tie-breaking rules often result in poor bounds. For example, consider a natural tie-braking rule that uniquely selects an optimal p^* closest to some fixed point, say the origin $\mathbf{0} \in \mathbb{R}^V$, in the ℓ_2-norm. If an instance with $\mathrm{conv}(\mathrm{argmin}\, g) = \left\{ p \in \mathbb{R}^2 \;\middle|\; p_2 - p_1 = 100 \right\}$ is given, $p^* \in \mathrm{conv}(\mathrm{argmin}\, g)$ closest to $\mathbf{0}$ is $p^* = (-50, 50)$. Then, if a given prediction is $\hat{p} = (1, 100)$, we have $\|p^* - \hat{p}\|_\infty = 50$ even though $\bar{\mu}(\hat{p}; g) = 1$. Therefore, such a tie-breaking rule that selects p^* closest to some fixed point generally results in poor prediction-dependent time complexity bounds even when $\hat{p}$ is close to the set, $\mathrm{conv}(\mathrm{argmin}\, g)$, of optimal solutions.

A.2. On Existing Results for Learning Predictions

Considering the above drawback of fixing p^*, one may want to let p^* be an optimal solution close to a given prediction $\hat{p}$. Indeed, most experiments in the previous studies seem to be implicitly based on this kind of idea; that is, they let p^* be an optimal solution returned by an algorithm warm-started by $\hat{p}$ and learn $\hat{p}$ to decrease loss values of the form $\|p^* - \hat{p}\|$. This idea, however, makes optimal $p^* \in \mathrm{conv}(\mathrm{argmin}\, g)$ selected depending on a given prediction $\hat{p}$. We below explain why the existing theoretical results for learning predictions $\hat{p}$ cannot deal with the dependence of p^* on $\hat{p}$.

PAC learning approach. A popular approach to obtaining guarantees for learning predictions is to use the PAC learning framework (Dinitz et al., 2021; Chen et al., 2022). With this approach, supposing input instances g to be drawn i.i.d. from a distribution, we usually analyze the *pseudo-dimension* of a function class of the form $\left\{ f_{\hat{p}} : g \mapsto \mathbb{R} \;\middle|\; \hat{p} \in \mathbb{R}^V \right\}$. The existing studies, however, analyzed the pseudo-dimension of a class of functions of the form $f_{\hat{p}}(p^*) = \|p^* - \hat{p}\|$, ignoring the non-uniqueness of optimal p^*. If we want to let p^* be an optimal solution close to $\hat{p}$, we must regard p^* as a function of $\hat{p}$ and specify how p^* is uniquely computed from $\hat{p}$ for each instance g; hence, the function should look like $f_{\hat{p}}(g) = \|p^*(\hat{p}, g) - \hat{p}\|$. No existing PAC learnability results for warm-starts with predictions have discussed such a complicated dependence.

Online learning approach. Another approach is to use online algorithms for learning predictions (Khodak et al., 2022; Sakaue & Oki, 2022). In those studies, the tth loss function takes the form $f_t(\hat{p}) = \|p_t^* - \hat{p}\|$, where p_t^* is some optimal solution selected for the tth instance, and the regret is defined as $\sum_{t=1}^{T} \|p_t^* - \hat{p}_t\| - \min_{\hat{p}^* \in [-C, +C]^V} \sum_{t=1}^{T} \|p_t^* - \hat{p}^*\|$. If we want to let p^* depend on $\hat{p}$, we need to learn $\hat{p}_t$ against an adversary who selects p_t^* depending on $\hat{p}_t$. In this situation, the above regret does not make sense; one obvious issue is that there is room for achieving a small regret by choosing $\hat{p}_t$ to make the second term large since $\hat{p}_t$ can affect p_t^*. A similar issue is discussed in (Arora et al., 2012), but the problem here would be more severe since the adversary acts *after* the learner. The existing studies have not considered this situation. Note that, although our method belongs to this category, our loss function, $\bar{\mu}(\hat{p}; g_t)$, is designed to avoid the non-uniqueness issue.

A.3. On Learnability Result of (Polak & Zub, 2022)

Polak & Zub (2022) have studied a maximum flow algorithm warm-started with predictions. The authors have stated that their method enjoys a time complexity bound proportional to $\|p^* - \hat{p}\|_1$ for an optimal flow p^* closest to $\hat{p}$. Although this means that p^* depends on $\hat{p}$, their analysis for learning $\hat{p}$ seems insufficient for handling the dependence, as detailed below.

In (Polak & Zub, 2022, Lemma 7), the authors present a uniform bound on the difference between empirical and expected losses. Specifically, given T instances $g_1, \dots, g_T$ drawn i.i.d. from a distribution $\mathcal{D}$, the lemma says that a bound of the form $\left| \frac{1}{T} \sum_{t=1}^{T} \|p^*(g_t) - p\|_1 - \mathbb{E}_{g \sim \mathcal{D}}[\|p^*(g) - p\|_1] \right| \le 1$ holds for all $p \in \mathbb{Z}^V$ satisfying some constraints with high probability. Then, in the proof of (Polak & Zub, 2022, Theorem 4), the authors use Lemma 7 substituting prediction $\hat{p}$ into p, where the "for all $p \in \mathbb{Z}^V$" part is justified by using the fact that prediction $\hat{p}$ is chosen after instances g_t are sampled from $\mathcal{D}$. This justification is correct if $p^*(g)$ is independent of $\hat{p}$, i.e., we can uniquely define function $f_g(p) = \|p^*(g) - p\|_1$ from g. However, if we let $p^*(g)$ be an optimal solution closest to $\hat{p}$, this justification is incorrect. For a bound like Lemma 7 to hold for $p^*(g)$ depending on $\hat{p}$, we need to derive a uniform convergence by, e.g., defining how $p^*(g)$ is computed from a pair of $(\hat{p}, g)$ and bounding the pseudo-dimension for the computation procedure. Considering the above, the sample complexity bound of (Polak & Zub, 2022) for learning $\hat{p}$ seems to be true only when $p^*(g)$ is fixed for each g independently of $\hat{p}$.

B. Proof of Lemma 2.5

Lemma 2.5. *Let $g : \mathbb{Z}^V \to \mathbb{R} \cup \{+\infty\}$ be an L-/L$^\natural$-convex function. For every $p \in \mathbb{Z}^V$, it holds that $\mu(p; g) = \bar{\mu}(p; g)$.*

Proof. Let $p \in \mathbb{Z}^V$. As discussed in Section 4.3, $\bar{\mu}(p; g)$ is equal to the negative of the total weight of a shortest path in $(\tilde{V}, \tilde{E})$, which we can represent as an optimal value of the following LP (see (Sakaue & Oki, 2022, Appendix D)):

$$\begin{aligned} &\text{maximize} && q_t - q_s \\ &\text{subject to} && q_j - q_i \le \tilde{w}_{ij}(p) \quad \forall ij \in \tilde{E}, \\ &&& q_0 = 0, \end{aligned} \tag{9}$$

where

$$\tilde{w}_{ij}(p) = \begin{cases} \gamma_{ij} - p_j + p_i & \text{if } ij \in E, \\ -\alpha_i + p_i & \text{if } i \in V \text{ and } j = 0, \\ \beta_j - p_j & \text{if } i = 0 \text{ and } j \in V, \\ 0 & \text{if } i = s \text{ or } j = t \end{cases}$$

for $-\alpha_i, \beta_j, \gamma_{ij} \in \mathbb{Z} \cup \{+\infty\}$. From $p \in \mathbb{Z}^V$, all $\tilde{w}_{ij}(p)$ are integers. Furthermore, the constraints in (9) can be written with a totally unimodular matrix. Therefore, the LP (9) has an integer optimal solution $q^* \in \mathbb{Z}^{\tilde{V}}$. Let $q^*_V \in \mathbb{Z}^V$ be the restriction of q^* to $V = \tilde{V} \setminus \{0, s, t\}$. Then, $p^* = p + q^*_V$ attains $\|p^* - p\|^{\pm}_\infty = \bar{\mu}(p; g)$, as shown in (Sakaue & Oki, 2022, Appendix D). Moreover, p^* is an integer vector due to $p \in \mathbb{Z}^V$ and $q^*_V \in \mathbb{Z}^V$, and thus we have $p^* \in \mathrm{conv}(\operatorname{argmin} g) \cap \mathbb{Z}^V = \operatorname{argmin} g$, where the equality is known as the *hole-free property* of L-/L$^\natural$-convex sets (Murota, 2003, Theorem 5.2 and Section 5.5). From $\|p^* - p\|^{\pm}_\infty = \bar{\mu}(p; g)$ and $p^* \in \operatorname{argmin} g$, we have $\mu(p; g) = \|p^* - p\|^{\pm}_\infty$, hence $\mu(p; g) = \|p^* - p\|^{\pm}_\infty = \bar{\mu}(p; g)$. □

C. Proof of Corollary 4.4

Corollary 4.4. *Let $\mathcal{D}$ be an (unknown) distribution over L-/L$^\natural$-convex functions $g : \mathbb{Z}^V \to \mathbb{R} \cup \{+\infty\}$, $\delta \in (0, 1]$, and $\varepsilon > 0$. Given $T = \Omega\left(\left(\frac{C}{\varepsilon}\right)^2 \left(n + \log \frac{1}{\delta}\right)\right)$ i.i.d. draws of $g_1, \dots, g_T \sim \mathcal{D}$, we can obtain $\hat{p} \in \mathbb{R}^V$ that satisfies*

$$\mathbb{E}_{g \sim \mathcal{D}}[\bar{\mu}(\hat{p}; g)] \le \min_{\hat{p}^* \in [-C, +C]^V} \mathbb{E}_{g \sim \mathcal{D}}[\bar{\mu}(\hat{p}^*; g)] + \varepsilon$$

with probability at least $1 - \delta$. Under the assumptions of Theorem 4.3, we can compute $\hat{p}$ in $\mathrm{O}(T \cdot (T_{\mathrm{ineq}} + n^2))$ time.

Proof. The basic proof idea is to use online-to-batch conversion (Cesa-Bianchi et al., 2004) to convert the regret bound (Theorem 4.3) into the sample complexity bound. Here, we use a refined variant, called anytime online-to-batch conversion (Cutkosky, 2019, Theorem 1), which is useful for ensuring the last iterate convergence of predictions computed for stochastic loss functions. We also use this technique in the experiments in Section 6.

To use (Cutkosky, 2019, Theorem 1), we slightly modify the online algorithm for computing predictions. In each tth round, let $q_t = \frac{1}{t} \sum_{t'=1}^t \hat{p}_{t'}$ and compute a subgradient z_t at q_t, i.e., $z_t \in \partial \bar{\mu}(q_t, g_t)$. The tth loss function revealed to an online learner is a linear loss function $f_t(\hat{p}) = \langle z_t, \hat{p} \rangle$, and the learner uses an online algorithm to compute $\hat{p}_1, \dots, \hat{p}_T \in [-C, +C]^V$ that satisfy a regret bound of $\sum_{t=1}^T f_t(\hat{p}_t) - \sum_{t=1}^T f_t(\hat{p}^*) = \sum_{t=1}^T \langle z_t, \hat{p}_t - \hat{p}^* \rangle = \mathrm{O}(C\sqrt{nT})$ for any $\hat{p}^* \in [-C, +C]^V$. Here, the learner can use OGD described in Section 4.1; one can easily confirm that it enjoys the $C\sqrt{2nT}$-regret bound also for the linearized loss $f_t(\hat{p}) = \langle z_t, \hat{p} \rangle$. Then, by substituting $\|z_t\|_1 \le 2$ and $\max\left\{ \|p - q\|_\infty \;\middle|\; p, q \in [-C, +C]^V \right\} \le 2C$ into (Cutkosky, 2019, Theorem 1), we can show that $\hat{p} = q_T$ satisfies the following inequality with a probability of at least $1 - \delta$:

$$\mathbb{E}_{g \sim \mathcal{D}}[\bar{\mu}(\hat{p}; g)] - \min_{\hat{p}^* \in [-C, +C]^V} \mathbb{E}_{g \sim \mathcal{D}}[\bar{\mu}(\hat{p}^*; g)] \le \frac{C\sqrt{2nT} + 8C\sqrt{T \log(2/\delta)}}{T} = \frac{C}{\sqrt{T}} \left(\sqrt{2n} + 8\sqrt{\log \frac{2}{\delta}} \right).$$

Thus, the sample size of $T = \Omega\Big(\big(\frac{C}{\varepsilon}\big)^2\big(n + \log\frac{1}{\delta}\big)\Big)$ is sufficient for ensuring that the right-hand side is at most ε.

As for the time complexity, the per-round running time of OGD is $T_{\mathrm{ineq}} + \mathrm{O}(n^2)$ by Lemma 4.8 (since $p_t^* \in \operatorname{argmin} g_t$ is available), and this is repeated T times to obtain $\hat{p} = q_T$. Therefore, it takes $\mathrm{O}(T \cdot (T_{\mathrm{ineq}} + n^2))$ time in total. □

D. Regret Lower Bound

We show an $\Omega(C\sqrt{nT})$ regret lower bound for online minimization of $\bar{\mu}(\hat{p}; g_t)$ to complement Theorem 4.3. The proof idea is based on (Hazan & Kale, 2012, Theorem 14), which presents a regret lower bound for online submodular minimization.

For ease of analysis, we only consider a learner who selects $\hat{p}_1, \dots, \hat{p}_T$ from $[-C, +C]^V$. Our OGD satisfies this condition due to the ℓ_2-projection onto $[-C, +C]^V$; hence the lower bound implies the tightness of the $\mathrm{O}(C\sqrt{nT})$ upper bound. Another remark is that we below obtain a lower bound by using g_t such that $\operatorname{argmin} g_t \cap [-C, +C]^V = \emptyset$, while predictions $\hat{p}$ are constrained to $[-C, +C]^V$. We leave it for future work to prove a lower bound using g_t with $\operatorname{argmin} g_t \cap [-C, +C]^V \neq \emptyset$.

Theorem D.1. *Let $C > 0$ be an integer. For any online leaner who plays $\hat{p}_1, \dots, \hat{p}_T \in [-C, +C]^V$, there is a sequence of L$^\natural$-convex functions $g_1, \dots, g_T$ such that the learner incurs an $\Omega(C\sqrt{nT})$ regret.*

Proof. Let $n = |V|$ be even and $i(t) = (t \bmod n/2) + 1 \in \{1, \dots, n/2\}$ for $t = 1, \dots, T$. In each tth round, choose a Rademacher random variable $\sigma_t \in \{-1, +1\}$ independently of all other random variables. Let g_t be an indicator function such that $\operatorname{dom} g_t$ is a singleton, $\{p_t^*\}$, where $p_t^* \in \mathbb{Z}^V$ has two non-zeros: the $i(t)$th entry is $3\sigma_t C$, the $(i(t) + n/2)$th entry is $-3\sigma_t C$, and the others are zero. Since we have $\operatorname{argmin} g_t = \{p_t^*\}$, for any $\hat{p} \in [-C, +C]^V$, it holds that

$$\bar{\mu}(\hat{p}; g_t) = \|p_t^* - \hat{p}\|_\infty^{\pm} = \max_{i \in V} \max\big\{0, p_{t,i}^* - \hat{p}_i\big\} + \max_{i \in V} \max\big\{0, \hat{p}_i - p_{t,i}^*\big\} = 6C + \sigma_t(\hat{p}_{i(t)+n/2} - \hat{p}_{i(t)}).$$

Thus, for any learner's choice $\hat{p}_t \in [-C, +C]^V$, we have $\mathbb{E}[\bar{\mu}(\hat{p}_t; g_t)] = 6C$, where the expectation is taken over the randomness of σ_t. Therefore, the expected total loss of any online learner is $6CT$.

We then show that there exists $\hat{p}^* \in [-C, +C]^V$ that has an $\Omega(C\sqrt{nT})$ advantage over the learner's expected loss, implying an $\Omega(C\sqrt{nT})$ regret lower bound. Let $\operatorname{sign}(x)$ denote a function that returns $+1$ if $x > 0$, 0 if $x = 0$, and -1 if $x < 0$. Let $X_i = \sum_{t : i(t) = i} \sigma_t$ for $i = 1, \dots, n/2$. Set the ith entry of $\hat{p}^*$ to $\operatorname{sign}(X_i) \times C$ for $i = 1, \dots, n/2$ and $-\operatorname{sign}(X_i) \times C$ for $i = n/2 + 1, \dots, n$. Then, in each tth round, the $i(t)$th entry of $p_t^* - \hat{p}^*$ causes a loss value of $2C$ if $\operatorname{sign}(X_{i(t)}) = \sigma_t$, $3C$ if $X_{i(t)} = 0$, and $4C$ otherwise. Similarly, the $(i(t) + n/2)$th entry causes a loss value of $2C$, $3C$, or $4C$. Hence we have

$$\sum_{t=1}^{T} \bar{\mu}(\hat{p}^*; g_t) = \sum_{t=1}^{T} \|p_t^* - \hat{p}^*\|_\infty^{\pm} = 3CT - C\sum_{i=1}^{n/2} |X_i| + 3CT - C\sum_{i=1}^{n/2} |X_i| = 6CT - 2C\sum_{i=1}^{n/2} |X_i|,$$

which implies that the expected regret is at least $2C \times \mathbb{E}\Big[\sum_{i=1}^{n/2} |X_i|\Big]$. Since each X_i is a sum of at least $\Big\lfloor \frac{T}{n/2} \Big\rfloor$ independent Rademacher random variables, Khintchine's inequality (see, e.g., (Cesa-Bianchi & Lugosi, 2006, Appendix A.1.4)) implies $\mathbb{E}[|X_i|] \geq \sqrt{\frac{1}{2}\Big\lfloor \frac{T}{n/2} \Big\rfloor}$. Thus, the expected regret is at least $2C \times \frac{n}{2}\sqrt{\frac{1}{2}\Big\lfloor \frac{T}{n/2} \Big\rfloor} = \Omega(C\sqrt{nT})$. This expected lower bound implies that there is a specific choice of σ_t values such that the learner incurs an $\Omega(C\sqrt{nT})$ regret. □

E. Transformation into Non-negative Edge Weights

We show how to transform the shortest path problem in Section 4.3 into another one with non-negative edge weights. Once we obtain such a transformed problem, we can use Dijkstra's algorithm to find a shortest path. The transformation is based on a so-called *potential*, which has been well studied in combinatorial optimization (Schrijver, 2003, Section 8.2).

Recall that the vertex set is $\tilde{V} = V_0 \cup \{s, t\}$, where $V_0 = \{0\} \cup V$, and the edge set is

$$\tilde{E} = E \cup \{\{0\} \times V\} \cup \{V \times \{0\}\} \cup \{\{s\} \times V_0\} \cup \{V_0 \times \{t\}\},$$

where $E = \{ij \mid i, j \in V; i \neq j\}$. The set of all simple s–t paths in $(\tilde{V}, \tilde{E})$ is denoted by $\mathcal{P} \subseteq 2^{\tilde{E}}$. We also have the following inequality-system representation of $\operatorname{conv}(\operatorname{argmin} g)$, as in Assumption 4.6:

$$\operatorname{conv}(\operatorname{argmin} g) = \left\{ p \in \mathbb{R}^V \;\middle|\; \begin{array}{l} \alpha_i \leq p_i \leq \beta_i \text{ for } i \in V, \\ p_j - p_i \leq \gamma_{ij} \text{ for distinct } i, j \in V \end{array} \right\}. \tag{10}$$

For any given prediction $\hat{p} \in \mathbb{R}^V$, the original (possibly negative) edge weights $\tilde{w}_{ij}(\hat{p})$ ($ij \in \tilde{E}$) are defined as follows:

$$\tilde{w}_{ij}(\hat{p}) = \begin{cases} \gamma_{ij} - \hat{p}_j + \hat{p}_i & \text{if } ij \in E, \\ -\alpha_i + \hat{p}_i & \text{if } i \in V \text{ and } j = 0, \\ \beta_j - \hat{p}_j & \text{if } i = 0 \text{ and } j \in V, \\ 0 & \text{if } i = s \text{ or } j = t. \end{cases}$$

We transform them into non-negative weights. We call $q \in \mathbb{R}^{\tilde{V}}$ a *potential* if $\tilde{w}_{ij}(\hat{p}) - q_j + q_i \geq 0$ holds for $ij \in \tilde{E}$. If we have a potential, we can define non-negative edge weights $\tilde{w}^+_{ij}(\hat{p}) \coloneqq \tilde{w}_{ij}(\hat{p}) - q_j + q_i$ for $ij \in \tilde{E}$. For any simple s–t path $P \in \mathcal{P}$ in $(\tilde{V}, \tilde{E})$, the telescoping sum implies

$$\sum_{ij \in P} \tilde{w}^+_{ij}(\hat{p}) = \sum_{ij \in S} (\tilde{w}_{ij}(\hat{p}) - q_j + q_i) = q_s - q_t + \sum_{ij \in S} \tilde{w}_{ij}(\hat{p}),$$

where q_s and q_t are independent of the choice of $P \in \mathcal{P}$. Hence $P \in \mathcal{P}$ is the shortest with respect to edge weights $\tilde{w}_{ij}(\hat{p})$ if and only if P is the shortest with respect to $\tilde{w}^+_{ij}(\hat{p})$. Therefore, once a potential is given, we can obtain non-negative edge weights that do not change the set of shortest paths in $\mathrm{O}(\tilde{E})$ time, and we can find a shortest path with Dijkstra's algorithm in $\mathrm{O}(|\tilde{E}| + |\tilde{V}| \log |\tilde{V}|)$ time. We below present how to obtain a potential from an arbitrary optimal solution $p^* \in \operatorname{argmin} g$, which is available for free since the tth instance is assumed to be solved in Theorem 4.3.

To simply notation, we add elements $\hat{p}_0 = \hat{p}_s = \hat{p}_t = 0$ to $\hat{p} \in \mathbb{R}^V$. Also, let $\gamma_{i0} = -\alpha_i$ for $i \in V$, $\gamma_{0j} = \beta_j$ for $j \in V$, $\gamma_{sj} = \hat{p}_j$ for $j \in V_0$, and $\gamma_{it} = -\hat{p}_i$ for $i \in V_0$. Then, the original edge weights $\tilde{w}_{ij}(\hat{p})$ can be written as

$$\tilde{w}_{ij}(\hat{p}) = \gamma_{ij} - \hat{p}_j + \hat{p}_i \quad \text{for } ij \in \tilde{E}. \tag{11}$$

Since we have $p^* \in \operatorname{argmin} g \subseteq \operatorname{conv}(\operatorname{argmin} g)$, $p^* \in \mathbb{R}^V$ satisfies the inequalities in (10). Thus, by additionally defining $p^*_0 = 0$, $p^*_s = \max_{j \in V_0}(p^*_j - \hat{p}_j)$, and $p^*_t = \min_{i \in V_0}(p^*_i - \hat{p}_i)$, we have

$$p^*_j - p^*_i \leq \gamma_{ij} \quad \text{for } ij \in \tilde{E}. \tag{12}$$

Then, $q \coloneqq p^* - \hat{p} \in \mathbb{R}^{\tilde{V}}$ is indeed a potential since we have

$$\tilde{w}_{ij} - q_j + q_i = \tilde{w}_{ij} - (p^*_j - \hat{p}_j) + (p^*_i - \hat{p}_i) \overset{(11)}{=} \gamma_{ij} - \hat{p}_j + \hat{p}_i - (p^*_j - \hat{p}_j) + (p^*_i - \hat{p}_i) = \gamma_{ij} - p^*_j + p^*_i \overset{(12)}{\geq} 0$$

for all $ij \in \tilde{E}$. By using this potential, we can obtain non-negative edge weights $\tilde{w}^+_{ij}(\hat{p})$ as described above.

F. Missing Proofs in Section 5

We detail how to obtain an inequality system of $\operatorname{conv}(\operatorname{argmin} g)$ for weighted matroid intersection (Appendix F.1), discrete energy minimization (Appendix F.2), and general L-/L$^\natural$-convex minimization under the value-oracle model (Appendix F.3).

F.1. Proof of Theorem 5.2

We discuss the weighted matroid intersection problem, a generalization of various problems such as bipartite matching and packing spanning trees. A *matroid* $\mathbf{M}$ consists of a finite set V and a non-empty set family $\mathcal{B} \subseteq 2^V$ of *bases* satisfying the following: for any $B_1, B_2 \in \mathcal{B}$ and $i \in B_1 \setminus B_2$, there exists $j \in B_2 \setminus B_1$ such that $B_1 \setminus \{i\} \cup \{j\}, B_2 \setminus \{j\} \cup \{i\} \in \mathcal{B}$. For any $v \in \mathbb{Z}^V$ and $X \subseteq V$, let $v(X) = \sum_{i \in X} v_i$. For any $v \in \mathbb{Z}^V$ and matroid $\mathbf{M} = (V, \mathcal{B})$, let $\mathcal{B}^v \coloneqq \operatorname{argmax}_{B \in \mathcal{B}} v(B)$ and $\mathbf{M}^v \coloneqq (V, \mathcal{B}^v)$, which also forms a matroid (Edmonds, 1971).

Let $\mathbf{M}_1 = (V, \mathcal{B}_1)$ and $\mathbf{M}_2 = (V, \mathcal{B}_2)$ be two matroids on an identical ground set V equipped with weights $w \in \mathbb{Z}^V$. We assume that the rank of each matroid is at most r (i.e., $\max_{B \in \mathcal{B}_k} |B| \leq r$ for $k = 1, 2$) and that independence oracles of $\mathbf{M}_1$ and $\mathbf{M}_2$ run in τ time, each of which returns whether input $X \subseteq V$ is a subset of some $B \in \mathcal{B}_k$ or not ($k = 1, 2$). The weighted matroid intersection problem asks to find $B \in \mathcal{B}_1 \cap \mathcal{B}_2$ that maximizes $w(B)$. Its dual problem is written as minimization of the following L-convex function g (see (Sakaue & Oki, 2022, Section 3.2)):

$$\underset{p \in \mathbb{Z}^V}{\text{minimize}} \quad g(p) = \max_{B \in \mathcal{B}_1} p(B) + \max_{B \in \mathcal{B}_2} (w - p)(B).$$

We below prove the following theorem.

Theorem 5.2. *Consider the dual problem,* $\min_{p \in Z^V} g(p)$*, of weighted matroid intersection defined on a ground set* V *of size* n*. If a maximum weight common base is available (or the problem is already solved), we can obtain an inequality system of the form* (4) *representing* $\mathrm{conv}(\mathrm{argmin}\, g)$ *in* $T_{\mathrm{ineq}} = \mathrm{O}(\tau n r)$ *time, where* r *is the rank of the matroids and* τ *is the running time of independence oracles.*

Proof. Let $B^* \in \mathcal{B}_1 \cap \mathcal{B}_2$ be any common base that maximizes $w(B^*)$. By the strong duality (Frank, 2011, Theorem 13.2.4), it holds that $w(B^*) = g(p^*)$ for any optimal dual solution $p^* \in \mathbb{Z}^V$. This means that $p \in \mathbb{Z}^V$ is optimal if and only if

$$B^* \in \operatorname*{argmax}_{B \in \mathcal{B}_1} p(B) \cap \operatorname*{argmax}_{B \in \mathcal{B}_2} (w - p)(B). \tag{13}$$

Furthermore, an optimality condition for linear maximization on matroid bases says that for any $v \in \mathbb{Z}^V$ and matroid $\mathbf{M} = (V, \mathcal{B})$, $B \in \mathrm{argmax}_{B' \in \mathcal{B}}\, v(B')$ holds if and only if $v_i \geq v_j$ for all $i \in B$ and $j \in V \setminus B$ with $B \setminus \{i\} \cup \{j\} \in \mathcal{B}$ (see, e.g., (Schrijver, 2003, Corollary 39.12b)). If we apply this condition to each of $\mathrm{argmax}_{B \in \mathcal{B}_1}\, p(B)$ and $\mathrm{argmax}_{B \in \mathcal{B}_2} (w - p)(B)$ in (13), the "if and only if" condition of (13) implies that we can represent $\mathrm{argmin}\, g$ as follows:

$$\left\{ p \in \mathbb{Z}^V \;\middle|\; \begin{array}{l} p_i \geq p_j \text{ for } (i,j) \in E_1(B^*), \\ w_i - p_i \geq w_j - p_j \text{ for } (i,j) \in E_2(B^*) \end{array} \right\}, \tag{14}$$

where B^* is an arbitrary maximum weight common base and $E_k(B^*) = \{\, (i,j) \mid i \in B^*, j \in V \setminus B^*, B^* \setminus \{i\} \cup \{j\} \in \mathcal{B}_k \,\}$ for $k = 1, 2$. The same inequality system on $\mathbb{R}^V$ represents $\mathrm{conv}(\mathrm{argmin}\, g)$, as in Proposition 2.2.

By the assumption in Theorem 5.2, a maximum weight common base B^* is available. Once B^* is given, we can construct the inequality system (14) in $\mathrm{O}(\tau |B^*||V \setminus B^*|) \lesssim \mathrm{O}(\tau n r)$ time, hence $T_{\mathrm{ineq}} = \mathrm{O}(\tau n r)$. □

We additionally show that the per-round running time of OGD is $\mathrm{O}(\tau n r + n \log n)$. Since $|E_k| = \mathrm{O}(|B^*||V \setminus B^*|) \lesssim \mathrm{O}(nr)$ for $k = 1, 2$, the inequality system (14) yields a graph $(\tilde{V}, \tilde{E})$ such that $|\tilde{E}| = \mathrm{O}(nr)$. Therefore, Dijkstra's algorithm in the proof of Lemma 4.8 takes $\mathrm{O}(nr + n \log n)$ time, and the per-round running time of OGD is $T_{\mathrm{ineq}} + \mathrm{O}(nr + n \log n) = \mathrm{O}(\tau n r + n \log n)$. Since solving local local optimization in Algorithm 1 takes $T_{\mathrm{loc}} = \mathrm{O}(\tau n r^{1.5})$ time as in Table 1, OGD's per-round running time is as short as T_{loc} if $\log n \lesssim \mathrm{O}(\tau r^{1.5})$.

F.2. Proof of Theorem 5.3

We discuss discrete energy minimization (Kolmogorov & Shioura, 2009), which appears in computer-vision (CV) applications. (Although Kolmogorov & Shioura (2009) considered undirected graphs, a similar result holds for directed graphs as follows; see also (Murota, 2003, Section 9).) Let (V, A) be a directed graph with $|V| = n$ and $|A| = m$. Each vertex $i \in V$ and edge $a \in A$ are associated with univariate convex functions $\phi_i : \mathbb{Z} \to \mathbb{R} \cup \{+\infty\}$ and $\psi_a : \mathbb{Z} \to \mathbb{R} \cup \{+\infty\}$, respectively, which we can evaluate in constant time. Then, a discrete energy minimization problem is written as

$$\operatorname*{minimize}_{p \in \mathbb{Z}^V} \quad g(p) = \sum_{i \in V} \phi_i(p_i) + \sum_{a = ij \in A} \psi_a(p_j - p_i), \tag{15}$$

which is an L$^\natural$-convex minimization problem and a generalization of minimum-cost flow. We assume $\mathrm{dom}\, g \subseteq [0, W]^V$ for some $W > 0$; this is usually true in CV applications since the range of pixel values is bounded. This assumption implies that we can restrict $\mathrm{dom}\, \phi_i$ and $\mathrm{dom}\, \psi_a$ to $[0, W] \cap \mathbb{Z}$ and $[-W, +W] \cap \mathbb{Z}$, respectively.

We then introduce the dual problem of (15). For a vector $\xi \in \mathbb{R}^A$, called a *flow*, we define its *boundary* $\partial \xi \in \mathbb{R}^V$ by $(\partial \xi)_i = \sum_{a \in \delta^+(i)} \xi_a - \sum_{a \in \delta^-(i)} \xi_a$ for $i \in V$, where $\delta^+(i)$ (resp. $\delta^-(i)$) is the set of edges entering to (resp. leaving from) $i \in V$. We can write the dual problem of (15) as the following convex cost flow problem:

$$\operatorname*{maximize}_{\xi \in \mathbb{R}^A} \quad f(\xi) = \sum_{i \in V} \min_{x \in \mathbb{Z}} \phi_i[-(\partial \xi)_i](x) + \sum_{a \in A} \min_{y \in \mathbb{Z}} \psi_a[-\xi_a](y),$$

where, for any $u : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ and $b \in \mathbb{R}$, $u[b]$ denotes a function defined by $u[b](x) = u(x) + bx$ for $x \in \mathbb{R}$. The following proposition is useful to characterize the primal-dual structure.

Proposition F.1 ((Kolmogorov & Shioura, 2009, Theorem 3.1); cf. (Murota, 2003, Theorem 9.4))**.** *Take any flow $\xi^* \in \mathbb{R}^A$ maximizing $f(\xi)$. Let α_i, β_i be integers with $\operatorname{argmin} \phi_i[-(\partial\xi^*)_i] = [\alpha_i, \beta_i] \cap \mathbb{Z}$ for $i \in V$, and let $\check{\gamma}_a, \hat{\gamma}_a$ be integers with $\operatorname{argmin} \psi_a[-{\xi^*}_a] = [\check{\gamma}_a, \hat{\gamma}_a] \cap \mathbb{Z}$ for $a \in A$. Then, we can represent $\operatorname{argmin} g$ as follows:*

$$\left\{ p^* \in \mathbb{Z}^V \;\middle|\; \begin{array}{l} \alpha_i \le p^*_i \le \beta_i \text{ for } i \in V, \\ \check{\gamma}_a \le p^*_j - p^*_i \le \hat{\gamma}_a \text{ for } a = ij \in A \end{array} \right\}. \tag{16}$$

We now prove the following theorem.

Theorem 5.3. *Consider discrete energy minimization, $\min_{p \in Z^V} g(p)$, defined on a graph with n vertices and m edges. If $\operatorname{dom} g \subseteq [0, W]^V$ for some $W > 0$ (which is true in most computer-vision applications), we can obtain an inequality system of the form (4) representing $\operatorname{conv}(\operatorname{argmin} g)$ in $T_{\text{ineq}} = \mathrm{O}(mn \log(n^2/m) \log(nW))$ time.*

Proof. From Proposition F.1, given an arbitrary optimal flow ξ^*, we can represent $\operatorname{argmin} g$ by the inequality system (16). We can find one such ξ^* by using an algorithm of (Ahuja et al., 2003) in $\mathrm{O}(mn \log(n^2/m) \log(nW))$ time. Then, since we can restrict $\operatorname{dom} \phi_i$ and $\operatorname{dom} \psi_a$ to $[0, W] \cap \mathbb{Z}$ and $[-W, +W] \cap \mathbb{Z}$, respectively, we can locate values of $\alpha_i, \beta_i, \check{\gamma}_a, \hat{\gamma}_a \in \mathbb{Z}$ for all $i \in V$ and $a \in A$ in $\mathrm{O}((n + m) \log W)$ time via binary search (which is faster than the algorithm for computing ξ^*). Therefore, we can obtain the inequality system (16) in $T_{\text{ineq}} = \mathrm{O}(mn \log(n^2/m) \log(nW))$ time, and the same system on $\mathbb{R}^V$ represents $\operatorname{conv}(\operatorname{argmin} g)$. □

F.3. Proof of Theorem 5.4

We consider L-/L$^\natural$-convex function minimization, $\min_{p \in \mathbb{Z}^V} g(p)$, with a value oracle of g. We prove the following theorem.

Theorem 5.4. *For general L-/L$^\natural$-convex function minimization, $\min_{p \in Z^V} g(p)$, such that $\operatorname{argmin} g \cap [-C, +C]^V \neq \emptyset$ and $\min g > -\infty$, we can obtain an inequality system of a subset of $\operatorname{conv}(\operatorname{argmin} g)$ that is sufficient for the subgradient computation in $T_{\text{ineq}} = \mathrm{O}(n^2 \log^2 C \cdot (\mathrm{EO} \cdot n^3 \log^2 n + n^4 \log^{\mathrm{O}(1)} n))$ time, where EO is the time for evaluating g.*

Proof. Recall that prediction $\hat{p} \in \mathbb{R}^V$, at which we compute a subgradient of $\bar{\mu}(\cdot; g)$, is always contained in $[-C, +C]^V$ since OGD performs the ℓ_2-projection onto $[-C, +C]^V$. First, we show that an inequality system of $S = \operatorname{conv}(\operatorname{argmin} g) \cap [-2C, +2C]^V$ is sufficient for computing a subgradient; that is, we only need to obtain an inequality system of the form

$$\alpha_i \le p_i \le \beta_i \text{ for } i \in V \quad \text{and} \quad p_j - p_i \le \gamma_{ij} \text{ for distinct } i, j \in V$$

such that

$$\alpha_i \in [-2C, +2C] \cap \mathbb{Z}, \qquad \beta_i \in [-2C, +2C] \cap \mathbb{Z}, \qquad \text{and} \qquad \gamma_{ij} \in [-4C, +4C] \cap \mathbb{Z}. \tag{17}$$

As discussed in Section 4.3, we can compute a subgradient of $\bar{\mu}(\hat{p}; g)$ by finding a shortest s–t path in $(\tilde{V}, \tilde{E})$ with weights $\tilde{w}_{ij}(\hat{p})$. This procedure is indeed equivalent to computing an $\ell_\infty^\pm$-projection of $\hat{p}$ onto the convex hull of an L$^\natural$-convex set, $\operatorname{conv}(\operatorname{argmin} g)$ (see (Sakaue & Oki, 2022, Appendix D)). Since we have $\hat{p} \in [-C, +C]^V$ and $\operatorname{argmin} g \cap [-C, +C]^V \neq \emptyset$, such an $\ell_\infty^\pm$-projection of $\hat{p}$ never goes out of $[-2C, +2C]^V$. Therefore, among all inequalities representing $\operatorname{conv}(\operatorname{argmin} g)$, those that do not intersect with $[-2C, +2C]^V$ can be ignored when computing a shortest path in $(\tilde{V}, \tilde{E})$. This implies that we can compute a subgradient of $\bar{\mu}(\hat{p}; g)$ if we have an inequality system of $S = \operatorname{conv}(\operatorname{argmin} g) \cap [-2C, +2C]^V$.

We then describe how to obtain an inequality system of S. From the above discussion, we can focus on searching for appropriate values of α_i, β_i, and γ_{ij} satisfying (17). We seek such values via binary search as follows. Consider, for example, doing binary search on $[-4C, +4C]$ to find an appropriate γ_{ij} value. Given a current γ_{ij} value, the inequality $p_j - p_i \le \gamma_{ij}$ is valid for all minimizers $p \in S$ if and only if adding a constraint $p_j - p_i \ge \gamma_{ij} + 1$ to $\min\{\, g(p) \mid p \in [-2C, +2C]^V \cap \mathbb{Z}^V \,\}$ increases the minimum value. We can check whether the latter is true by solving the L$^\natural$-convex minimization problem with an effective domain restricted to $[-2C, +2C]^V \cap \{\, p \in \mathbb{Z}^V \mid p_j - p_i \ge \gamma_{ij} + 1 \,\}$. By using the steepest descent scaling algorithm (Murota, 2003, Section 10.3.2), we can solve the problem via $\mathrm{O}(\log C)$ times submodular function minimization, each of which can be solved with an $\mathrm{O}(\mathrm{EO} \cdot n^3 \log^2 n + n^4 \log^{\mathrm{O}(1)} n)$-time algorithm of (Lee et al., 2015). By repeating this test $\mathrm{O}(\log C)$ times, the binary search produces a tight inequality $p_j - p_i \le \gamma_{ij}$ (i.e., it defines a facet of $\operatorname{conv}(S)$). Similarly, we can find α_i and β_i values. To find all α_i, β_i, and γ_{ij} values, we perform binary search $\mathrm{O}(n^2)$ times. Therefore, the total computation time for obtaining the desired inequality system is $T_{\text{ineq}} = \mathrm{O}(n^2 \log^2 C \cdot (\mathrm{EO} \cdot n^3 \log^2 n + n^4 \log^{\mathrm{O}(1)} n))$. □