A Primal-Dual Frank-Wolfe Algorithm for Linear Programming*

Matthew Hough† Stephen A. Vavasis‡

February 28, 2024

Abstract

We present two first-order primal-dual algorithms for solving saddle point formulations of linear programs, namely FWLP (Frank-Wolfe Linear Programming) and FWLP-P. The former iteratively applies the Frank-Wolfe algorithm to both the primal and dual of the saddle point formulation of a standard-form LP. The latter is a modification of FWLP in which regularizing perturbations are used in computing the iterates. We show that FWLP-P converges to a primal-dual solution with error $\mathcal{O}(1/\sqrt{k})$ after k iterations, while no convergence guarantees are provided for FWLP. We also discuss the advantages of using FWLP and FWLP-P for solving very large LPs. In particular, we argue that only part of the matrix A is needed at each iteration, in contrast to other first-order methods.

1 Introduction

In recent years, data science applications have given birth to problems of very large scale. This poses a problem for mature LP solvers that require solving a system of linear equations at each iteration. First-order methods (FoMs) for linear programming aim to solve LPs in such a way that their most expensive operation at each iteration is the product of a matrix and a vector. Their goal is to provide an alternative to the practitioner over LP algorithms such as the simplex method or interior point methods for large-scale problems.

The Frank-Wolfe algorithm [7], also referred to as the conditional gradient algorithm, is a FoM for minimizing a smooth convex objective function over a compact convex set. A major benefit of the Frank-Wolfe algorithm is that each iteration requires only the

*This research was supported in part by a Discovery Grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

†Department of Combinatorics & Optimization, University of Waterloo, 200 University Ave. W., Waterloo, ON, N2L 3G1, Canada, mhough@uwaterloo.ca.

‡Department of Combinatorics & Optimization, University of Waterloo, 200 University Ave. W., Waterloo, ON, N2L 3G1, Canada, vavasis@uwaterloo.ca.

solution of a linear optimization problem over a convex constraint set, a problem which can be solved efficiently over many constraint sets used in practice. It is known that the Frank-Wolfe algorithm converges at a rate of $\mathcal{O}(1/k)$ [12].

The focus of this paper is on finding optimal solutions to linear programs in standard form, that is, solving the following optimization problem:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned} \tag{1}$$

where we assume throughout that (1) has an optimal solution. The dual linear program associated with (1) is

$$\begin{aligned} \max \quad & \boldsymbol{b}^T \boldsymbol{y} \\ \text{s.t.} \quad & A^T \boldsymbol{y} \leq \boldsymbol{c}. \end{aligned} \tag{2}$$

We propose two first-order primal-dual algorithms for simultaneously solving (1) and (2) inspired by the Frank-Wolfe algorithm but using the non-standard step-size of $1/(k+1)$ analyzed in [8]. We call our algorithms FWLP and FWLP-P. FWLP, first introduced in [11], is derived from iteratively applying the Frank-Wolfe algorithm to the primal and dual problems of a modified saddle-point formulation of (1):

$$\min_{\boldsymbol{x} \in \Delta} \max_{\boldsymbol{y} \in \Gamma} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := \boldsymbol{c}^T \boldsymbol{x} + \boldsymbol{y}^T (\boldsymbol{b} - A\boldsymbol{x}), \tag{3}$$

where we define

$$\Delta = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{e}^T \boldsymbol{x} \leq \xi\} \text{ and } \Gamma = [-\eta, \eta]^m. \tag{4}$$

Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ denote an optimal primal-dual pair of solutions to (1) and (2). The parameters $\xi, \eta > 0$ are assumed to be chosen large enough to ensure that $2\xi \geq \|\boldsymbol{x}^*\|_1$ and $2\eta \geq \|\boldsymbol{y}^*\|_\infty$, thus describing redundant constraints. Despite their redundance, these constraints are necessary for compactness of the feasible sets corresponding to the primal and dual subproblems solved by FWLP. FWLP-P is a modification of the original FWLP algorithm in which regularizing perturbations are introduced when computing the iterates. FWLP-P is of theoretical importance to FWLP since we are able to prove the convergence of FWLP-P but no convergence proof is known yet for FWLP.

With the ubiquity of very large-scale problems, FoMs for linear programming have seen interest of late [1, 2, 4, 14]. Such methods typically apply existing ideas and algorithms from continuous optimization to linear programming. Our work is related particularly to [9], where Gidel et al. use the Frank-Wolfe algorithm to solve general convex-concave saddle point problems:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y}), \tag{5}$$

where f is a smooth convex-concave function and $\mathcal{X} \times \mathcal{Y}$ is a convex compact set. In analyzing the convergence of their method, the authors introduce a potential function

bounding the distance to optimality and then show that this potential function decreases to zero at a given rate. We employ this technique in Section 3 to analyze the convergence of FWLP-P. The results of [9] do not directly apply to (3) because their strong-convexity assumption is not satisfied. The FWLP algorithm is also closely related to the Generalized Fictitious Play Algorithm proposed by Hammond in her 1984 PhD thesis [10]. Applying Hammond's algorithm to (3) yields an algorithm very similar to FWLP, the only difference being that the update of $\boldsymbol{y}_k$ is based on $\boldsymbol{x}_k$ instead of $\boldsymbol{x}_{k+1}$ like in FWLP. For a more in-depth analysis of the similarities between FWLP and Hammond's Generalized Fictitious Play, see [11, Chapter 6.2].

The algorithms FWLP and FWLP-P are described in Section 2. Iterations of FWLP are simpler and faster than those of FWLP-P, but we do not have a convergence proof of FWLP. Our preliminary computational tests (not reported here) indicate that the two algorithms converge at comparable rates.

Our convergence analysis of FWLP-P, presented in Section 3, first introduces a potential function U_k which we show that for the iterates of FWLP-P has distance $\mathcal{O}(1/\sqrt{k})$ from zero after k iterations. To complete the analysis, we show that as $U_k \to 0$, FWLP-P and FWLP converge to a primal-dual solution of (1). The potential U_k is similar to a traditional primal-dual optimality gap as discussed in Section 4. A secondary contribution of this paper is the discussion in Section 5 of how FWLP and FWLP-P can be implemented efficiently in a way that only part of the matrix A is needed at each iteration.

2 FWLP and FWLP-P

At each iteration, FWLP performs a Frank-Wolfe update on the primal and the dual of (3) using the step-size $1/(k+1)$ instead of the standard step-size of $1/(k+2)$. The step-size $1/(k+1)$ was first analyzed by Freund and Grigas in [8]. Notably, FWLP uses the information obtained from the primal update, $\boldsymbol{x}_{k+1}$, in the computation of the dual update:

$$\boldsymbol{r}_{k+1} := \operatorname{argmin}_{\boldsymbol{r}} \left\{ (\boldsymbol{c} - A^T \boldsymbol{y}_k)^T \boldsymbol{r} : \boldsymbol{r} \in \Delta \right\}, \tag{6}$$

$$\boldsymbol{x}_{k+1} := \frac{k}{k+1} \boldsymbol{x}_k + \frac{1}{k+1} \boldsymbol{r}_{k+1}, \tag{7}$$

$$\boldsymbol{s}_{k+1} := \operatorname{argmax}_{\boldsymbol{s}} \left\{ (\boldsymbol{b} - A\boldsymbol{x}_{k+1})^T \boldsymbol{s} : \boldsymbol{s} \in \Gamma \right\}, \tag{8}$$

$$\boldsymbol{y}_{k+1} := \frac{k}{k+1} \boldsymbol{y}_k + \frac{1}{k+1} \boldsymbol{s}_{k+1}. \tag{9}$$

It is not hard to see that steps (6) and (8) above can be written in closed form (for more detail, see [11, Chapter 6]), giving rise to Algorithm 2.1. In this algorithm and for the remainder of the paper, $\boldsymbol{e}_i$ denotes the ith column of the identity matrix, whose length is determined from the context.

Algorithm 2.1 FWLP: A primal-dual algorithm for (1) based on Frank-Wolfe [11].

Require: Starting points $\boldsymbol{x}_0 \in \mathbb{R}^n_+, \boldsymbol{y}_0 \in \mathbb{R}^m$, constraint data A and $\boldsymbol{b}$, and objective $\boldsymbol{c}$.

Parameters: $\xi, \eta > 0$ such that $2\|\boldsymbol{x}^*\| \leq \xi$ and $2\|\boldsymbol{y}^*\| \leq \eta$.

1: **for** $k = 1, 2, \ldots$ **do**

2: Determine $i = \operatorname{argmin}_t [\boldsymbol{c} - A^\top \boldsymbol{y}_k]_t$.

3: **if** $[\boldsymbol{c} - A^\top \boldsymbol{y}_k]_i \geq 0$ **then**

4: Step towards zero in $\boldsymbol{x}$:

$$\boldsymbol{x}_{k+1} := \frac{k}{k+1}\boldsymbol{x}_k. \tag{10}$$

5: **else**

6: Step toward ξ for $x^{(i)}$, otherwise step toward zero for $x^{(j)}, j \neq i$:

$$\boldsymbol{x}_{k+1} = \frac{k}{k+1}\boldsymbol{x}_k + \frac{\xi}{k+1}\boldsymbol{e}_i. \tag{11}$$

7: **end if**

8: Step towards $\pm\eta$ for each coordinate in $\boldsymbol{y}$ according to the sign pattern of $\boldsymbol{b} - A\boldsymbol{x}_{k+1}$:

$$\boldsymbol{y}_{k+1} := \frac{k}{k+1}\boldsymbol{y}_k + \frac{\eta}{k+1}\operatorname{sgn}(\boldsymbol{b} - A\boldsymbol{x}_{k+1}). \tag{12}$$

9: **end for**

FWLP-P adds the regularizing perturbations $\|\boldsymbol{r}\|^2/(2\sqrt{k})$ and $\|\boldsymbol{s}\|^2/(2\sqrt{k})$ to steps (10)/(11) and (12) in the above description of FWLP yielding (15) and (17) respectively in Algorithm 2.2. We analyze this algorithm in the next section.

Note that the computation of $\boldsymbol{s}_{k+1}$ in FWLP-P is a separable optimization problem allowing a simple median algorithm for each coordinate entry of $\boldsymbol{s}_{k+1}$. The computation of $\boldsymbol{r}_{k+1}$ is more elaborate, but it requires only evaluation of the smallest (most negative) elements of $\boldsymbol{c} - A^T\boldsymbol{y}_k$. We cover this in more detail in Section 5 but of note here is that FWLP-P retains some of the benefit of FWLP, namely sub-linear time computation per iteration, since only a part of A is required. Note also that the formulas (15) and (17) for $\boldsymbol{r}_{k+1}$ and $\boldsymbol{s}_{k+1}$ can be written equivalently as projections:

$$\boldsymbol{r}_{k+1} := \operatorname{proj}_\Delta(\sqrt{k}(A^T\boldsymbol{y}_k - \boldsymbol{c})), \tag{13}$$

$$\boldsymbol{s}_{k+1} := \operatorname{proj}_\Gamma(\sqrt{k}(\boldsymbol{b} - A\boldsymbol{x}_{k+1})). \tag{14}$$

We discuss how to compute these projections efficiently in Section 5.2.

Algorithm 2.2 FWLP-P: FWLP with perturbations.

Require: Starting points $\boldsymbol{x}_0 \in \mathbb{R}^n_+, \boldsymbol{y}_0 \in \mathbb{R}^m$, constraint data A and $\boldsymbol{b}$, and objective $\boldsymbol{c}$.
Parameters: $\xi, \eta > 0$ such that $2\|\boldsymbol{x}^*\| \leq \xi$ and $2\|\boldsymbol{y}^*\| \leq \eta$.

1: **for** $k = 1, 2, \ldots$ **do**

$$\boldsymbol{r}_{k+1} := \text{argmin}_{\boldsymbol{r}} \left\{ (\boldsymbol{c} - A^T\boldsymbol{y}_k)^T\boldsymbol{r} + \frac{\|\boldsymbol{r}\|^2}{2\sqrt{k}} : \boldsymbol{r} \in \Delta \right\}, \tag{15}$$

$$\boldsymbol{x}_{k+1} := \frac{k}{k+1}\boldsymbol{x}_k + \frac{1}{k+1}\boldsymbol{r}_{k+1}, \tag{16}$$

$$\boldsymbol{s}_{k+1} := \text{argmax}_{\boldsymbol{s}} \left\{ (\boldsymbol{b} - A\boldsymbol{x}_{k+1})^T\boldsymbol{s} - \frac{\|\boldsymbol{s}\|^2}{2\sqrt{k}} : \boldsymbol{s} \in \Gamma \right\}, \tag{17}$$

$$\boldsymbol{y}_{k+1} := \frac{k}{k+1}\boldsymbol{y}_k + \frac{1}{k+1}\boldsymbol{s}_{k+1}. \tag{18}$$

2: **end for**

3 Convergence analysis of FWLP-P

The forthcoming analysis derives a recursion (30) for potential U_k below satisfied by the iterates of FWLP-P. The recursion has two perturbation terms δ_{k+1} and ϵ_{k+1} that are subsequently bounded in (32) and (34). We show that the quantity U_k goes to 0 like $1/\sqrt{k}$.

3.1 Deriving the recursion and potential function

Let us introduce

$$\delta_{k+1} := \boldsymbol{r}_{k+2}^T(\boldsymbol{c} - A^T\boldsymbol{y}_{k+1}) + \frac{1}{2\sqrt{k+1}}\|\boldsymbol{r}_{k+2}\|^2 - \boldsymbol{r}_{k+1}^T(\boldsymbol{c} - A^T\boldsymbol{y}_{k+1}) - \frac{k}{2(k+1)\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2. \tag{19}$$

Rearrange this equation:

$$\delta_{k+1} - \boldsymbol{r}_{k+2}^T(\boldsymbol{c} - A^T\boldsymbol{y}_{k+1}) - \frac{1}{2\sqrt{k+1}}\|\boldsymbol{r}_{k+2}\|^2 = -\boldsymbol{r}_{k+1}^T(\boldsymbol{c} - A^T\boldsymbol{y}_{k+1}) - \frac{k}{2(k+1)\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2. \tag{20}$$

Rewrite the first-term on the RHS of (20), that is, $-\boldsymbol{r}_{k+1}^T(\boldsymbol{c} - A^T\boldsymbol{y}_{k+1})$, using (18):

$$\begin{aligned} -\boldsymbol{r}_{k+1}^T(\boldsymbol{c} - A^T\boldsymbol{y}_{k+1}) &= -\boldsymbol{r}_{k+1}^T\left(\boldsymbol{c} - A^T\left(\frac{k}{k+1}\boldsymbol{y}_k + \frac{1}{k+1}\boldsymbol{s}_{k+1}\right)\right) \\ &= -\frac{k}{k+1}\boldsymbol{r}_{k+1}^T(\boldsymbol{c} - A^T\boldsymbol{y}_k) - \frac{1}{k+1}\boldsymbol{r}_{k+1}^T(\boldsymbol{c} - A^T\boldsymbol{s}_{k+1}). \end{aligned} \tag{21}$$

Substitute (21) for the first term on the RHS in (20), moving the second term of (21) $-\boldsymbol{r}_{k+1}^T(\boldsymbol{c}-A^T\boldsymbol{s}_{k+1})/(k+1)$ to the LHS, thereby rewriting (20) as:

$$\begin{aligned}\delta_{k+1}-\boldsymbol{r}_{k+2}^T(\boldsymbol{c}-A^T\boldsymbol{y}_{k+1})-\frac{1}{2\sqrt{k+1}}\|\boldsymbol{r}_{k+2}\|^2+\frac{1}{k+1}\boldsymbol{r}_{k+1}^T(\boldsymbol{c}-A^T\boldsymbol{s}_{k+1})&\\ =-\frac{k}{k+1}\boldsymbol{r}_{k+1}^T(\boldsymbol{c}-A^T\boldsymbol{y}_k)-\frac{k}{2(k+1)\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2&\\ =\frac{k}{k+1}\left(-\boldsymbol{r}_{k+1}^T(\boldsymbol{c}-A^T\boldsymbol{y}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2\right).&\quad(22)\end{aligned}$$

The point of this algebra is that the 2nd and 3rd terms of the LHS correspond to the RHS advanced from k to $k+1$, thus setting up some of the terms of the recursion.

Next, introduce for $k\geq 2$

$$\epsilon_{k+1}:=\frac{k}{k+1}\left(-\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{x}_k)+\frac{\sqrt{k+1}}{2k}\|\boldsymbol{s}_{k+1}\|^2+\boldsymbol{s}_k^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{s}_k\|^2\right).\quad(23)$$

Rewrite this equation as

$$\epsilon_{k+1}+\frac{k}{k+1}\left(\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{\sqrt{k+1}}{2k}\|\boldsymbol{s}_{k+1}\|^2\right)=\frac{k}{k+1}\left(\boldsymbol{s}_k^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{s}_k\|^2\right).\quad(24)$$

Rearranging (16) yields $\boldsymbol{x}_k=(k+1)\boldsymbol{x}_{k+1}/k-\boldsymbol{r}_{k+1}/k$. This means that the factor $\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{x}_k)$ appearing in the LHS of (24) may be rewritten

$$\begin{aligned}\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{x}_k)&=\boldsymbol{s}_{k+1}^T\left(\boldsymbol{b}-A\left(\frac{k+1}{k}\boldsymbol{x}_{k+1}-\frac{1}{k}\boldsymbol{r}_{k+1}\right)\right)\\&=\frac{k+1}{k}\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{x}_{k+1})-\frac{1}{k}\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{r}_{k+1}).\quad(25)\end{aligned}$$

Thus, the LHS of (24) is rewritten:

$$\begin{aligned}\text{LHS of (24)}&=\epsilon_{k+1}+\frac{k}{k+1}\left(\frac{k+1}{k}\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{x}_{k+1})-\frac{1}{k}\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{r}_{k+1})-\frac{\sqrt{k+1}}{2k}\|\boldsymbol{s}_{k+1}\|^2\right)\\&=\epsilon_{k+1}+\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{x}_{k+1})-\frac{1}{k+1}\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{r}_{k+1})-\frac{1}{2\sqrt{k+1}}\|\boldsymbol{s}_{k+1}\|^2.\quad(26)\end{aligned}$$

Thus, we have rewritten (24) as

$$\begin{aligned}\epsilon_{k+1}+\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{x}_{k+1})-\frac{1}{k+1}\boldsymbol{s}_{k+1}^T(\boldsymbol{b}-A\boldsymbol{r}_{k+1})-\frac{1}{2\sqrt{k+1}}\|\boldsymbol{s}_{k+1}\|^2&\\ =\frac{k}{k+1}\left(\boldsymbol{s}_k^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{s}_k\|^2\right).&\quad(27)\end{aligned}$$

Here, we see the correspondence between the 2nd and 4th term on the LHS with the two terms on the RHS (with k advanced by 1).

Multiply (16) by $\boldsymbol{c}$ and rearrange to obtain

$$\boldsymbol{c}^T\boldsymbol{x}_{k+1} - \frac{1}{k+1}\boldsymbol{c}^T\boldsymbol{r}_{k+1} = \frac{k}{k+1}\boldsymbol{c}^T\boldsymbol{x}_k. \tag{28}$$

Similarly, from (18),

$$-\boldsymbol{b}^T\boldsymbol{y}_{k+1} + \frac{1}{k+1}\boldsymbol{b}^T\boldsymbol{s}_{k+1} = -\frac{k}{k+1}\boldsymbol{b}^T\boldsymbol{y}_k. \tag{29}$$

Now add (22), (27), (28), and (29), noting that many quantities on the LHS cancel, while all quantities on the RHS contain the factor $k/(k+1)$, to obtain a recursion:

$$\delta_{k+1} + \epsilon_{k+1} + U_{k+1} = \frac{k}{k+1}U_k, \tag{30}$$

where for $k \geq 2$

$$U_k := -\boldsymbol{r}_{k+1}^T(\boldsymbol{c} - A^T\boldsymbol{y}_k) - \frac{1}{2\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2 + \boldsymbol{s}_k^T(\boldsymbol{b} - A\boldsymbol{x}_k) - \frac{1}{2\sqrt{k}}\|\boldsymbol{s}_k\|^2 + \boldsymbol{c}^T\boldsymbol{x}_k - \boldsymbol{b}^T\boldsymbol{y}_k. \tag{31}$$

We call U_k the potential function, and we will use this in our convergence analysis to bound the distance from optimality of (1) and (2).

3.2 Bounding the potential function

Our next task is to lower bound δ_{k+1} and ϵ_{k+1} so that we can use (30) to develop a more useful bound on the potential function.

Lemma 1. *Recall ϵ_{k+1} defined in* (23)*. We have the bound*

$$\epsilon_{k+1} \geq -\frac{m\eta^2}{6k^2\sqrt{k-1}}.$$

Proof. By adding and subtracting multiples of $\|\boldsymbol{s}_{k+1}\|^2$ and and $\|\boldsymbol{s}_k\|^2$ inside (23), we obtain

$$\epsilon_{k+1} = \frac{k}{k+1}(\epsilon'_{k+1} + \epsilon''_{k+1} + \epsilon'''_{k+1}),$$

where

$$\begin{aligned}
\epsilon'_{k+1} &:= -\boldsymbol{s}_{k+1}^T(\boldsymbol{b} - A\boldsymbol{x}_k) + \frac{1}{2\sqrt{k-1}}\|\boldsymbol{s}_{k+1}\|^2 + \boldsymbol{s}_k^T(\boldsymbol{b} - A\boldsymbol{x}_k) - \frac{1}{2\sqrt{k-1}}\|\boldsymbol{s}_k\|^2, \\
\epsilon''_{k+1} &:= \left(\frac{1}{2\sqrt{k-1}} - \frac{1}{2\sqrt{k}}\right)\|\boldsymbol{s}_k\|^2, \\
\epsilon'''_{k+1} &:= \left(\frac{\sqrt{k+1}}{2k} - \frac{1}{2\sqrt{k-1}}\right)\|\boldsymbol{s}_{k+1}\|^2.
\end{aligned}$$

We see that $\epsilon'_{k+1} \geq 0$ because $\boldsymbol{s}_k$ is the maximizer of $\boldsymbol{s}^T(\boldsymbol{b} - A\boldsymbol{x}_k) - \|\boldsymbol{s}\|^2/(2\sqrt{k-1})$ according to the definition (17), while $\boldsymbol{s}_{k+1}$ is some other feasible point, and ϵ'_{k+1} is the difference between the two objective values. We also see that $\epsilon''_{k+1} \geq 0$.

As for ϵ'''_{k+1}, use the estimate $\|\boldsymbol{s}_k\| \leq \sqrt{m}\eta$ according to (17). Finally, by taking a common denominator and observing that $(k-1/(3k))^2 \leq k^2 - 1$, we obtain a lower bound of $-1/(6k^2\sqrt{k-1})$ on the parenthesized factor. Thus, adding the three contributions,

$$\epsilon_{k+1} \geq -\frac{m\eta^2}{6k^2\sqrt{k-1}} \cdot \frac{k}{k+1} \geq -\frac{m\eta^2}{6k^2\sqrt{k-1}}. \tag{32}$$

□

Lemma 2. *Recall δ_{k+1} defined in* (19)*. We have the bound*

$$\delta_{k+1} \geq -D/k^{3/2},$$

where D is a positive constant depending on the data defined in (35) *below.*

Proof. Split δ_{k+1} into four terms:

$$\delta_{k+1} = \delta'_{k+1} + \delta''_{k+1} + \delta'''_{k+1} + \delta^{\mathrm{iv}}_{k+1}$$

where

$$\begin{aligned}
\delta'_{k+1} &:= \boldsymbol{r}_{k+2}^T(\boldsymbol{c} - A^T\boldsymbol{y}_k) + \frac{1}{2\sqrt{k}}\|\boldsymbol{r}_{k+2}\|^2 - \boldsymbol{r}_{k+1}^T(\boldsymbol{c} - A^T\boldsymbol{y}_k) - \frac{1}{2\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2, \\
\delta''_{k+1} &:= (\boldsymbol{r}_{k+2} - \boldsymbol{r}_{k+1})^T A^T(\boldsymbol{y}_k - \boldsymbol{y}_{k+1}), \\
\delta'''_{k+1} &:= \left(\frac{1}{2\sqrt{k+1}} - \frac{1}{2\sqrt{k}}\right)\|\boldsymbol{r}_{k+2}\|^2, \\
\delta^{\mathrm{iv}}_{k+1} &:= \left(\frac{1}{2\sqrt{k}} - \frac{k}{2(k+1)\sqrt{k}}\right)\|\boldsymbol{r}_{k+1}\|^2.
\end{aligned}$$

We observe that $\delta'_{k+1} \geq 0$ because $\boldsymbol{r}_{k+1}$ is the minimizer of $\boldsymbol{r}^T(\boldsymbol{c} - A^T\boldsymbol{y}_k) + \|\boldsymbol{r}\|^2/(2\sqrt{k})$ according to (15), whereas $\boldsymbol{r}_{k+2}$ is some other feasible point.

Next, we turn to δ''_{k+1}, a product of three factors. Starting on the first factor,

$$\begin{aligned}
\|\boldsymbol{r}_{k+2} - \boldsymbol{r}_{k+1}\| &= \|\mathrm{proj}_\Delta(\sqrt{k+1}(A^T\boldsymbol{y}_{k+1} - \boldsymbol{c})) - \mathrm{proj}_\Delta(\sqrt{k}(A^T\boldsymbol{y}_k - \boldsymbol{c})\| \\
&\leq \|\sqrt{k+1}(A^T\boldsymbol{y}_{k+1} - \boldsymbol{c})) - \sqrt{k}(A^T\boldsymbol{y}_k - \boldsymbol{c})\| \qquad (33) \\
&= \|(\sqrt{k+1} - \sqrt{k})(A^T\boldsymbol{y}_{k+1} - \boldsymbol{c}) + \sqrt{k}(A^T\boldsymbol{y}_{k+1} - \boldsymbol{c} - (A^T\boldsymbol{y}_k - \boldsymbol{c}))\| \\
&= \|(\sqrt{k+1} - \sqrt{k})(A^T\boldsymbol{y}_{k+1} - \boldsymbol{c}) + \sqrt{k}A^T(\boldsymbol{y}_{k+1} - \boldsymbol{y}_k)\| \\
&\leq |\sqrt{k+1} - \sqrt{k}| \cdot \|A^T\boldsymbol{y}_{k+1} - \boldsymbol{c}\| + \sqrt{k}\|A\| \cdot \|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\|.
\end{aligned}$$

Here, the first line follows from (13), the second (that is, (33)) from the fact that the Lipschitz constant of $\mathrm{proj}_C(\cdot)$ is 1 for any closed nonempty convex set C, the third line

adds and subtracts the same term, and the last line applies the triangle inequality and submultiplicativity.

Now we use the facts that $\sqrt{k+1} - \sqrt{k} \leq 1/\sqrt{k}$, and, from (18),

$$\boldsymbol{y}_{k+1} - \boldsymbol{y}_k = \frac{1}{k+1}(\boldsymbol{s}_{k+1} - \boldsymbol{y}_k),$$

and finally, the bounds $\|\boldsymbol{y}_k\| \leq \sqrt{m}\eta$, $\|\boldsymbol{s}_k\| \leq \sqrt{m}\eta$ to conclude that $\|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\| \leq 2\sqrt{m}\eta/(k+1)$ and thus

$$\|\boldsymbol{r}_{k+2} - \boldsymbol{r}_{k+1}\| \leq (1/\sqrt{k}) \cdot (\|A\|\sqrt{m}\eta + \|\boldsymbol{c}\|) + (2/\sqrt{k})\|A\|\sqrt{m}\eta.$$

This takes care of the first factor in δ''_{k+1}. The middle factor is bound by $\|A\|$, and the third factor $\|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\|$ is bounded by (see the previous paragraph) $2\sqrt{m}\eta/(k+1)$. Thus, overall, we obtain

$$\begin{aligned}\delta''_{k+1} &\geq -\|\boldsymbol{r}_{k+2} - \boldsymbol{r}_{k+1}\| \cdot \|A\| \cdot \|\boldsymbol{y}_k - \boldsymbol{y}_{k+1}\| \\ &\geq -C/k^{3/2},\end{aligned}$$

where C depends on the problem data:

$$C \leq 2\|A\|\sqrt{m}\eta(3\|A\|\sqrt{m}\eta + \|\boldsymbol{c}\|).$$

For δ'''_{k+1}, by finding a common denominator and then multiplying the resulting fraction by $\sqrt{k} + \sqrt{k+1}$, we obtain

$$\delta'''_{k+1} \geq -1/(4k^{3/2}) \cdot \|\boldsymbol{r}_{k+2}\|^2 \geq -1/(4k^{3/2}) \cdot \xi^2.$$

Finally, one sees that $\delta^{\text{iv}}_{k+1} \geq 0$. Putting all of these terms together yields:

$$\delta_{k+1} \geq -D/k^{3/2}, \tag{34}$$

where

$$D := 2\|A\|\sqrt{m}\eta(3\|A\|\sqrt{m}\eta + \|\boldsymbol{c}\|) + \frac{\xi^2}{4}. \tag{35}$$

□

Combining (30), Lemma 1, and Lemma 2, we now have for all $k \geq 2$

$$U_{k+1} \leq \frac{k}{k+1}U_k + \frac{D}{k^{3/2}} + \frac{m\eta^2}{6k^2\sqrt{k-1}}.$$

In fact, since $\sqrt{k} \geq 1/\sqrt{k-1}$ for all $k \geq 2$, we can enlarge D to obtain the bound

$$U_{k+1} \leq \frac{k}{k+1}U_k + \frac{\bar{D}}{k^{3/2}}, \tag{36}$$

where

$$\bar{D} := D + \frac{m\eta^2}{6}.$$

With this bound, we prove the following bound on U_k.

Theorem 1. *Recall U_k defined in* (31). *We have the bound*

$$U_{k+1} \leq \frac{F}{\sqrt{k}},$$

for all $k \geq 2$, where $F := \max\{\sqrt{2}U_2, 6\bar{D}\}$.

Proof. The $k = 2$ case follows immediately from the definition of F. Suppose inductively that $U_k \leq F/\sqrt{k}$. Then we check:

$$\begin{aligned}
U_{k+1} &\leq \frac{k}{k+1}U_k + \bar{D}/k^{3/2} \\
&\leq \frac{kF}{(k+1)\sqrt{k}} + \bar{D}/k^{3/2} \\
&= \frac{k^2F + \bar{D}(k+1)}{(k+1)k^{3/2}} \\
&= \frac{1}{\sqrt{k+1}} \cdot \frac{k^2F + \bar{D}(k+1)}{\sqrt{k+1} \cdot k^{3/2}} \\
&\leq \frac{1}{\sqrt{k+1}} \cdot \frac{k^2F + \bar{D}(k+1)}{(\sqrt{k} + 1/(3\sqrt{k}))k^{3/2}} \\
&= \frac{1}{\sqrt{k+1}} \cdot \frac{k^2F + \bar{D}(k+1)}{k^2 + k/3} \\
&\leq \frac{1}{\sqrt{k+1}} \cdot \frac{k^2F + kF/3}{k^2 + k/3} \\
&= \frac{1}{\sqrt{k+1}} \cdot F.
\end{aligned}$$

Here, we used (36) for the first line, the induction hypothesis for the second line, the inequality $\sqrt{k+1} \geq \sqrt{k} + 1/(3\sqrt{k})$ for $k \geq 1$ on the 5th line (easy to confirm by squaring) and the assumption $\bar{D} \leq F/6$ on the 7th line, which implies for all $k \geq 1$ that $\bar{D}(k+1) \leq Fk/3$.

Thus, $U_k \leq F/\sqrt{k}$. □

The perturbation terms from (15) and (17) in FWLP-P were essential to this analysis. Without them, as in FWLP, there is no useful bound on how much the primal step $\boldsymbol{r}_{k+2}$ can differ from $\boldsymbol{r}_{k+1}$ when the index of the most-violated constraint changes from one iteration to the next. With the perturbations, we are able to obtain the inequality (33).

3.3 Proving convergence of the potential function implies convergence of the iterates

We have shown that the potential function U_k decreases at a rate of $\mathcal{O}(1/\sqrt{k})$ as k increases. It remains to show that the potential function bounds the distance of the iterates from an optimizer. We first need the following lemmas. Both lemmas have the

same flavor: if an infeasible point for an LP improves on the optimal objective value, then the amount of objective improvement is bounded in terms of the amount of infeasibility.

Lemma 3. *Let p^* denote the optimal value of* (1)*. Assume p^* is finite, i.e.,* (1) *is feasible and bounded. Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be an arbitrary optimizing primal-dual pair of* (1) *and* (2)*. Then for an arbitrary $\hat{\boldsymbol{x}} \geq \mathbf{0}$,*

$$\boldsymbol{c}^T \hat{\boldsymbol{x}} \geq p^* - \|\boldsymbol{y}^*\|_\infty \|\boldsymbol{b} - A\hat{\boldsymbol{x}}\|_1. \tag{37}$$

Proof. Select an arbitrary optimizing pair $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ for (1). Let $\boldsymbol{k} := \boldsymbol{b} - A\hat{\boldsymbol{x}}$. Consider the LP,

$$\begin{array}{rl} \min_{\boldsymbol{x}} & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} - \boldsymbol{k}, \\ & \boldsymbol{x} \geq \mathbf{0}. \end{array} \tag{38}$$

This LP is clearly feasible since $\hat{\boldsymbol{x}}$ satisfies the constraints. It is also bounded, as argued by contradiction: If (38) were feasible and unbounded, then there would exist a certificate of unboundedness, which is a vector $\boldsymbol{w}$ such that $\boldsymbol{w} \geq \mathbf{0}$, $A\boldsymbol{w} = \mathbf{0}$, and $\boldsymbol{c}^T \boldsymbol{w} < 0$. However, such a $\boldsymbol{w}$ would also certify unboundedness of (1), which we have already assumed to be bounded.

Therefore, the dual of (38), which is,

$$\begin{array}{rl} \max_{\boldsymbol{y}} & (\boldsymbol{b} - \boldsymbol{k})^T \boldsymbol{y} \\ \text{s.t.} & A^T \boldsymbol{y} \leq \boldsymbol{c}, \end{array} \tag{39}$$

has an optimal solution, say $\hat{\boldsymbol{y}}$. Since $\boldsymbol{y}^*$ is also feasible for (39), we have the following chain of inequalities

$$\begin{aligned} p^* &= \boldsymbol{b}^T \boldsymbol{y}^* && \text{(by strong duality of (1))} \\ &= (\boldsymbol{b} - \boldsymbol{k})^T \boldsymbol{y}^* + \boldsymbol{k}^T \boldsymbol{y}^* \\ &\leq (\boldsymbol{b} - \boldsymbol{k})^T \boldsymbol{y}^* + \|\boldsymbol{k}\|_1 \cdot \|\boldsymbol{y}^*\|_\infty \\ &\leq (\boldsymbol{b} - \boldsymbol{k})^T \hat{\boldsymbol{y}} + \|\boldsymbol{k}\|_1 \cdot \|\boldsymbol{y}^*\|_\infty && \text{(since } \hat{\boldsymbol{y}} \text{ maximizes (39))} \\ &\leq \boldsymbol{c}^T \hat{\boldsymbol{x}} + \|\boldsymbol{k}\|_1 \cdot \|\boldsymbol{y}^*\|_\infty. && \text{(by weak duality between (38) and (39))} \end{aligned}$$

Recalling $\boldsymbol{k} = \boldsymbol{b} - A\hat{\boldsymbol{x}}$, the final line in this chain establishes (37). □

Lemma 4. *Let d^* denote the optimal value of* (2)*. Assume d^* is finite, i.e.,* (2) *is feasible and bounded. Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be an arbitrary optimizing primal-dual pair of* (1) *and* (2)*. Then for an arbitrary $\hat{\boldsymbol{y}}$,*

$$\boldsymbol{b}^T \hat{\boldsymbol{y}} \leq d^* + \|\boldsymbol{x}^*\|_1 \cdot l, \tag{40}$$

where l measures the infeasibility of $\hat{\boldsymbol{y}}$, that is,

$$l := \max(0, \max_j \{\boldsymbol{e}_j^T (A^T \hat{\boldsymbol{y}} - \boldsymbol{c})\}). \tag{41}$$

Proof. This proof is analogous to the previous proof. Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be an arbitrary primal-dual optimizer of (2). Consider the dual-form LP given by

$$\begin{array}{rl} \max_{\boldsymbol{y}} & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.} & A^T\boldsymbol{y} \leq \boldsymbol{c} + l\boldsymbol{e}, \end{array} \tag{42}$$

where l is from (41) and $\boldsymbol{e}$ denotes the vector of all 1's. This LP is clearly feasible since $\hat{\boldsymbol{y}}$ satisfies the constraint. Furthermore, it is bounded as argued by contradiction. If (42) were feasible and unbounded, there would exist a certificate of unboundedness, that is, a vector $\boldsymbol{z}$ such that $A^T\boldsymbol{z} \leq \boldsymbol{0}$ and $\boldsymbol{b}^T\boldsymbol{z} > 0$. However, this certificate would also certify the unboundedness of (2), which is assumed to be bounded.

Therefore, the dual of (42), which is

$$\begin{array}{rl} \min_{\boldsymbol{x}} & (\boldsymbol{c} + l\boldsymbol{e})^T\boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{array} \tag{43}$$

has an optimal solution which we denote $\hat{\boldsymbol{x}}$. Since $\boldsymbol{x}^*$ is also feasible for (43), we have the following chain of inequalities:

$$\begin{aligned} d^* &= \boldsymbol{c}^T\boldsymbol{x}^* && \text{(by strong duality of (2))} \\ &= (\boldsymbol{c} + l\boldsymbol{e})^T\boldsymbol{x}^* - l\boldsymbol{e}^T\boldsymbol{x}^* \\ &\geq (\boldsymbol{c} + l\boldsymbol{e})^T\boldsymbol{x}^* - l\|\boldsymbol{x}^*\|_1 \\ &\geq (\boldsymbol{c} + l\boldsymbol{e})^T\hat{\boldsymbol{x}} - l\|\boldsymbol{x}^*\|_1 && \text{(by the optimality of } \hat{\boldsymbol{x}} \text{ in (43))} \\ &\geq \boldsymbol{b}^T\hat{\boldsymbol{y}} - l\|\boldsymbol{x}^*\|_1 && \text{(by weak duality betwen (42) and (43)).} \end{aligned}$$

The final line after rearrangement is (40). □

We are now ready to prove the penultimate theorem that shows convergence of FWLP-P. We show that, assuming a primal-dual optimal LP solution exists and ξ, η have been chosen correctly, the U_k plus terms that tend to 0 bound the distance from optimality. Note that an analog of Theorem 2 can be derived for FWLP using a similar proof. This analog is not presented here since we are not able to show that $U_k \to 0$ for FWLP.

Theorem 2. *Suppose $k \geq 2$. Then,*

$$\boldsymbol{x}_k \geq \boldsymbol{0}, \tag{44}$$

$$\|\boldsymbol{b} - A\boldsymbol{x}_k\|_1 \leq \frac{2U_k}{\eta} + \frac{\xi^2}{\eta\sqrt{k}} + \frac{\eta}{\sqrt{k-1}}, \tag{45}$$

$$\max(0, \max_j\{\boldsymbol{e}_j^T(A^T\boldsymbol{y}_k - \boldsymbol{c})\}) \leq \frac{2U_k}{\xi} + \frac{\xi}{\sqrt{k}} + \frac{\eta^2}{\xi\sqrt{k-1}}, \tag{46}$$

$$\boldsymbol{c}^T\boldsymbol{x}_k - \boldsymbol{b}^T\boldsymbol{y}_k \leq U_k, \tag{47}$$

provided

$$\xi \geq 2\|\boldsymbol{x}^*\|_1, \qquad \eta \geq 2\|\boldsymbol{y}^*\|_\infty. \tag{48}$$

Proof. It is immediate from the algorithm that (44) holds. Next, notice that $\boldsymbol{s}_k$ is a solution to

$$\max_{\boldsymbol{s}\in\Gamma}\left\{\boldsymbol{s}^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{1}{2\sqrt{k-1}}\|\boldsymbol{s}\|^2\right\},$$

so for arbitrary $\boldsymbol{s}\in\Gamma$,

$$\begin{aligned}\boldsymbol{s}_k^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{s}_k\|^2 &\geq \boldsymbol{s}_k^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{1}{2\sqrt{k-1}}\|\boldsymbol{s}_k\|^2,\\ &\geq \boldsymbol{s}^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{1}{2\sqrt{k-1}}\|\boldsymbol{s}\|^2.\end{aligned} \tag{49}$$

Setting $\boldsymbol{s}=\boldsymbol{0}$ gives

$$\boldsymbol{s}_k^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{s}_k\|^2\geq 0.$$

Similarly, $\boldsymbol{r}_{k+1}$ is a solution to

$$\min_{\boldsymbol{r}}\left\{\boldsymbol{r}^T(\boldsymbol{c}-A^T\boldsymbol{y}_k)+\frac{1}{2\sqrt{k}}\|\boldsymbol{r}\|^2\right\}, \tag{50}$$

so for arbitrary $\boldsymbol{r}\in\Delta$,

$$\boldsymbol{r}_{k+1}^T(\boldsymbol{c}-A^T\boldsymbol{y}_k)+\frac{1}{2\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2\leq \boldsymbol{r}^T(\boldsymbol{c}-A^T\boldsymbol{y}_k)+\frac{1}{2\sqrt{k}}\|\boldsymbol{r}\|^2. \tag{51}$$

Setting $\boldsymbol{r}=\boldsymbol{0}$ gives

$$-\boldsymbol{r}_{k+1}^T(\boldsymbol{c}-A^T\boldsymbol{y}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2\geq 0. \tag{52}$$

The above two results imply with (31) that $U_k\geq \boldsymbol{c}^T\boldsymbol{x}_k-\boldsymbol{b}^T\boldsymbol{y}_k$, thus establishing (47).

Setting $\boldsymbol{s}:=\eta\cdot\operatorname{sgn}(\boldsymbol{b}-A\boldsymbol{x}_k)\in\Gamma$ in (49) gives the bound

$$\boldsymbol{s}_k^T(\boldsymbol{b}-A\boldsymbol{x}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{s}_k\|^2\geq \eta\|\boldsymbol{b}-A\boldsymbol{x}_k\|_1-\frac{1}{2\sqrt{k-1}}m\eta^2. \tag{53}$$

Similarly, set $\boldsymbol{r}:=\xi\boldsymbol{e}_j$ where $j=\operatorname{argmax}_j\{\boldsymbol{e}_j^T(A^T\boldsymbol{y}_k-\boldsymbol{c})\}$. Noting $\boldsymbol{r}\in\Delta$, we can use (51) to obtain the bound

$$-\boldsymbol{r}_{k+1}^T(\boldsymbol{c}-A^T\boldsymbol{y}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2\geq \xi\cdot\max_j\{\boldsymbol{e}_j^T(A^T\boldsymbol{y}_k-\boldsymbol{c})\}-\frac{1}{2\sqrt{k}}\xi^2.$$

In fact, from (52) the left-hand side above is nonnegative. It follows that we can strengthen the above bound to

$$-\boldsymbol{r}_{k+1}^T(\boldsymbol{c}-A^T\boldsymbol{y}_k)-\frac{1}{2\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2\geq \xi\cdot l-\frac{1}{2\sqrt{k}}\xi^2, \tag{54}$$

where $l = \max(0, \max_j\{\boldsymbol{e}_j^T(A^T\boldsymbol{y}_k - \boldsymbol{c})\})$. Using Lemma 3, Lemma 4, (31), (53), and (54), we may write

$$\begin{aligned}
U_k &= -\boldsymbol{r}_{k+1}^T(\boldsymbol{c} - A^T\boldsymbol{y}_k) - \frac{1}{2\sqrt{k}}\|\boldsymbol{r}_{k+1}\|^2 + \boldsymbol{s}_k^T(\boldsymbol{b} - A\boldsymbol{x}_k) - \frac{1}{2\sqrt{k}}\|\boldsymbol{s}_k\|^2 + \boldsymbol{c}^T\boldsymbol{x}_k - \boldsymbol{b}^T\boldsymbol{y}_k, \\
&\geq \xi \cdot l - \frac{1}{2\sqrt{k}}\xi^2 + \eta\|\boldsymbol{b} - A\boldsymbol{x}_k\|_1 - \frac{1}{2\sqrt{k-1}}m\eta^2 + \boldsymbol{c}^T\boldsymbol{x}_k - \boldsymbol{b}^T\boldsymbol{y}_k, \\
&\geq \xi \cdot l - \frac{1}{2\sqrt{k}}\xi^2 + \eta\|\boldsymbol{b} - A\boldsymbol{x}_k\|_1 - \frac{1}{2\sqrt{k-1}}m\eta^2 + p^* - \|\boldsymbol{y}^*\|_\infty\|\boldsymbol{b} - A\boldsymbol{x}_k\|_1 - d^* - \|\boldsymbol{x}^*\|_1 \cdot l, \\
&\geq \xi \cdot l - \frac{1}{2\sqrt{k}}\xi^2 + \eta\|\boldsymbol{b} - A\boldsymbol{x}_k\|_1 - \frac{1}{2\sqrt{k-1}}m\eta^2 + p^* - d^* - \frac{\eta}{2}\|\boldsymbol{b} - A\boldsymbol{x}_k\|_1 - \frac{\xi}{2} \cdot l \\
&= \frac{\xi}{2} \cdot l - \frac{1}{2\sqrt{k}}\xi^2 + \frac{\eta}{2}\|\boldsymbol{b} - A\boldsymbol{x}_k\|_1 - \frac{1}{2\sqrt{k-1}}m\eta^2,
\end{aligned}$$

where the fourth line used the assumption (48) and the final line used the fact that $p^* = d^*$. We are left with the bound

$$\frac{\xi}{2} \cdot l + \frac{\eta}{2}\|\boldsymbol{b} - A\boldsymbol{x}_k\|_1 \leq U_k + \frac{\xi^2}{2\sqrt{k}} + \frac{m\eta^2}{2\sqrt{k-1}},$$

where both terms on the left are nonnegative. It follows that each term must be individually bounded by the right-hand side above. This establishes (45) and (46). □

Theorem 3. *The iterates of FWLP-P converge to an ϵ-optimal solution of* (1) *and its dual* (2) *after $\mathcal{O}(1/\epsilon^2)$ iterations.*

Proof. This follows immediately by applying the bound from Theorem 1 in Theorem 2. □

4 Relating U_k to the standard primal-dual gap

Consider the general saddle-point problem

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y}).$$

A standard measure of the optimality gap of an iteration $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ used in the analysis of many primal-dual algorithms for saddle-point problems (for example, [6, 9, 15]) is the primal-dual gap:

$$\max\left\{f(\boldsymbol{x}_k, \hat{\boldsymbol{y}}_k) - f(\hat{\boldsymbol{x}}_k, \boldsymbol{y}_k) : \hat{\boldsymbol{x}}_k \in \mathcal{X}, \hat{\boldsymbol{y}}_k \in \mathcal{Y}\right\}. \tag{55}$$

Obviously, the solution to (55) is

$$\begin{aligned}
\hat{\boldsymbol{x}}_k &= \operatorname{argmin}\{f(\boldsymbol{x}, \boldsymbol{y}_k) : \boldsymbol{x} \in \mathcal{X}\}, \\
\hat{\boldsymbol{y}}_k &= \operatorname{argmax}\{f(\boldsymbol{x}_k, \boldsymbol{y}) : \boldsymbol{y} \in \mathcal{Y}\}.
\end{aligned}$$

It is noted in [2] that for the saddle-point problem associated with LP

$$\min_{\boldsymbol{x}\geq\boldsymbol{0}} \max_{\boldsymbol{y}\in\mathbb{R}^m} \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{y}^T(\boldsymbol{b} - A\boldsymbol{x}),$$

the primal-dual gap can be infinite, since the feasible set $\mathbb{R}^n_+ \times \mathbb{R}^m$ is unbounded. This is not an issue for the modified saddle-point formulation (3) used in our analysis, since the feasible set $\Delta \times \Gamma$ is bounded by virtue of the redundant constraints.

Recall the definition of $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$ from (3). Let

$$\mathcal{M}_k := \max\{\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{s}) - \mathcal{L}(\boldsymbol{r}, \boldsymbol{y}_k) : (\boldsymbol{r}, \boldsymbol{s}) \in \Delta \times \Gamma\},$$

which is the specialization of (55) to our setting. We argue that $\mathcal{M}_k$ is a perturbation of U_k via the following bound:

$$\begin{aligned}
|\mathcal{M}_k - U_k| &= \left| \max_{\boldsymbol{s}\in\Gamma}[(\boldsymbol{b} - A\boldsymbol{x}_k)^T\boldsymbol{s} + \boldsymbol{c}^T\boldsymbol{x}_k] - \min_{\boldsymbol{r}\in\Delta}[(\boldsymbol{c} - A^T\boldsymbol{y}_k)^T\boldsymbol{r} + \boldsymbol{b}^T\boldsymbol{y}_k] - U_k \right| \\
&\le \left| \max_{\boldsymbol{s}\in\Gamma} \left[(\boldsymbol{b} - A\boldsymbol{x}_k)^T\boldsymbol{s} + \boldsymbol{c}^T\boldsymbol{x}_k - \frac{\|\boldsymbol{s}\|^2}{2\sqrt{k-1}}\right] - \min_{\boldsymbol{r}\in\Delta} \left[(\boldsymbol{c} - A^T\boldsymbol{y}_k)^T\boldsymbol{r} + \boldsymbol{b}^T\boldsymbol{y}_k + \frac{\|\boldsymbol{r}\|^2}{2\sqrt{k}}\right] - U_k \right| \\
&\quad + \max_{(\boldsymbol{r},\boldsymbol{s})\in\Delta\times\Gamma} \left| \frac{\|\boldsymbol{s}\|^2}{2\sqrt{k-1}} + \frac{\|\boldsymbol{r}\|^2}{2\sqrt{k}} \right| \\
&\le \left| \left[(\boldsymbol{b} - A\boldsymbol{x}_k)^T\boldsymbol{s}_k + \boldsymbol{c}^T\boldsymbol{x}_k - \frac{\|\boldsymbol{s}_k\|^2}{2\sqrt{k-1}}\right] - \left[(\boldsymbol{c} - A^T\boldsymbol{y}_k)^T\boldsymbol{r}_{k+1} + \boldsymbol{b}^T\boldsymbol{y}_k + \frac{\|\boldsymbol{r}_{k+1}\|^2}{2\sqrt{k}}\right] - U_k \right| \\
&\quad + \frac{m\eta^2}{2\sqrt{k-1}} + \frac{\xi^2}{2\sqrt{k}} \\
&\le \frac{m\eta^2 + \xi^2}{2\sqrt{k-1}} + \mathcal{O}(k^{-3/2}).
\end{aligned}$$

Here, the second line adds and subtracts the same terms and then applies the triangle inequality for $|\cdot|$. The third line uses the definitions of $\boldsymbol{s}_k$ from (17) and $\boldsymbol{r}_k$ from (15). The fourth line uses (31), noting that the terms all cancel out except for the difference between a denominator of $2\sqrt{k-1}$ versus a denominator of $2\sqrt{k}$. This small remainder is written as $\mathcal{O}(k^{-3/2})$ on the fourth line.

5 Efficient implementation of FWLP and FWLP-P

A major advantage of FWLP and FWLP-P is their low computational cost per iteration. Naïve implementations of Algorithms 2.1 and 2.2 have iteration cost bounded by the cost of a full matrix-vector product. We discuss below how this can be significantly improved with an efficient implementation.

5.1 FWLP

Consider, e.g., the iteration $k = 1$. Suppose we store $A\boldsymbol{x}_1$. In Algorithm 2.1 we can perform an extra step to update $A\boldsymbol{x}_2$ by either computing

$$A\boldsymbol{x}_2 = \frac{k}{2} A\boldsymbol{x}_1,$$

or

$$A\boldsymbol{x}_2 = \frac{k}{2}A\boldsymbol{x}_1 + \frac{\xi}{2}A\boldsymbol{e}_i,$$

where i is the index of the most violated dual constraint, computed as $\text{argmin}\{c_i - \boldsymbol{e}_i^T A^T \boldsymbol{y}_1\}$. Ignoring the cost of computing i, such an update runs in $\mathcal{O}(m+n)$ operations per iteration (cost of updating $\boldsymbol{x}_k$ and $A\boldsymbol{x}_k$). Note that $\mathcal{O}(m+n)$ can be reduced to $\mathcal{O}(m)$ if we keep track of the product of scaling factors $k/(k+1)$ of $\boldsymbol{x}_{k+1}$ in a separate variable. In contrast, the naïve implementation of the primal update in FWLP, where one instead performs matrix-vector products, runs in $\mathcal{O}(mn)$ operations per iteration.

The dual update for, e.g., $k = 1$, is given by

$$\boldsymbol{y}_2 = \frac{1}{2}\boldsymbol{y}_1 + \frac{\eta}{2}\text{sgn}(\boldsymbol{b} - A\boldsymbol{x}_2),$$

which also runs in $\mathcal{O}(m)$ since $A\boldsymbol{x}_2$ has already been computed in the primal step. The naïve implementation again takes $\mathcal{O}(mn)$ operations.

To improve the cost of computing the index i, first note that in Algorithm 2.1 we only need to compute $\boldsymbol{e}_i^T A^T \boldsymbol{y}_k$ for indices i such that there is a possibility that $i = \text{argmin}_j\{c_j - \boldsymbol{e}_j^T A\boldsymbol{y}_k\}$. This could be implemented by a data structure to store the indices $[n]$ in some order so that only the possibly most infeasible indices need to be considered at each iteration. In more detail, suppose j_k indexes the most violated dual constraint on iteration k, i.e., $j_k = \text{argmin}_j[\boldsymbol{c} - A^T\boldsymbol{y}_k]_j$. Suppose $j \in [n]$ is some other index. Based on the value of $[\boldsymbol{c} - A^T\boldsymbol{y}_k]_j - [\boldsymbol{c} - A^T\boldsymbol{y}_k]_{j_k}$ and prior knowledge of the stepsize (which tends to 0 with k), one knows in advance that constraint j could not be the most violated constraint prior to iteration k', where k' is a computable index satisfying $k' > k$. Then constraint j does not even have to be considered by the algorithm on all iterations between k and k'.

Thus, the algorithm maintains some subset of constraints $\mathcal{S}_k$ on iteration k that need to be examined for possibly being the most violated. The computation of $A^T\boldsymbol{y}_k$ in the primal update thus runs in $\mathcal{O}(|\mathcal{S}_k| \cdot m)$ plus the cost of updating the data structure to obtain $\mathcal{S}_{k+1}$. Naturally, the estimate $\mathcal{O}(|\mathcal{S}_k| \cdot m)$ is further reduced if A is sparse.

The total cost of iteration k for this implementation of FWLP is thus

$$\mathcal{O}(n + |\mathcal{S}_k| \cdot m),$$

plus the cost of updating the proposed data structure and precomputation. For large problems, one would expect that $|\mathcal{S}_k| \ll n$, allowing significant speedup to be achieved by using the proposed efficient implementation of FWLP. And, as mentioned earlier, the "n" term may be dropped with a careful implementation for updating $\boldsymbol{x}_k$, and the "$|\mathcal{S}_k| \cdot m$" term is reduced in the presence of sparsity.

5.2 FWLP-P

The iterations of FWLP-P differ from FWLP in that the steps (15) and (17) are more computationally involved.

To solve (15), we equivalently consider the projection form (13), which amounts to solving the quadratic programming problem

$$\begin{aligned} \arg\min_{\boldsymbol{x}} \quad & \frac{1}{2}\|\boldsymbol{w}_0 - \boldsymbol{x}\|^2 \\ \text{s.t.} \quad & \boldsymbol{e}^T\boldsymbol{x} \leq \xi, \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

where $\boldsymbol{w}_0 = \sqrt{k}(A^T\boldsymbol{y}_k - \boldsymbol{c})$. Expanding, rescaling, and dropping constant terms gives the equivalent problem

$$\begin{aligned} \arg\min_{\boldsymbol{x}} \quad & -\boldsymbol{w}^T\boldsymbol{x} + \frac{1}{2}\|\boldsymbol{x}\|^2 \\ \text{s.t.} \quad & \boldsymbol{e}^T\boldsymbol{x} \leq 1, \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned} \tag{56}$$

where $\boldsymbol{w} = \boldsymbol{w}_0/\xi$. The KKT conditions of (56) are as follows

$$-\boldsymbol{w} + \boldsymbol{x} + \mu\boldsymbol{e} - \boldsymbol{z} = \boldsymbol{0}, \tag{57}$$

$$\boldsymbol{e}^T\boldsymbol{x} \leq 1, \tag{58}$$

$$\mu(\boldsymbol{e}^T\boldsymbol{x} - 1) = 0, \tag{59}$$

$$\boldsymbol{z}^T\boldsymbol{x} = 0, \tag{60}$$

$$\boldsymbol{x}, \mu, \boldsymbol{z} \geq \boldsymbol{0}. \tag{61}$$

We first prove the following lemma, which will become useful once we define our algorithm for solving (56).

Lemma 5. *Suppose $(\boldsymbol{x}, \mu, \boldsymbol{z})$ form a KKT solution for (56) with $\boldsymbol{e}^T\boldsymbol{x} = 1$. There must exist an index $j \in [n]$ such that $w_j > \mu$.*

Furthermore, we have the following cases regardless of whether $\boldsymbol{e}^T\boldsymbol{x} = 1$.

(i) If $w_j > \mu$ for some $j \in [n]$, then $x_j > 0$ and $z_j = 0$.

(ii) If $w_j < \mu$ for some $j \in [n]$, then $x_j = 0$ and $z_j > 0$.

(iii) If $w_j = \mu$ for some $j \in [n]$, then $x_j = z_j = 0$.

Proof. By (57), we have

$$w_i = x_i - z_i + \mu \tag{62}$$

for all $i \in [n]$. The condition $\boldsymbol{e}^T\boldsymbol{x} = 1$ along with the nonnegativity of $\boldsymbol{x}$ from (61) implies that there must exist an index $j \in [n]$ such that $x_j > 0$. Now, the complementarity condition (60) along with the nonnegativity of $\boldsymbol{z}$ from (61) imply that $z_j = 0$. It follows from (62) that $w_j = x_j + \mu > \mu$.

Now for (i): suppose, $w_j > \mu$ for some $j \in [n]$. By (62), $x_j - z_j + \mu > \mu$, which implies $x_j > z_j$. But since $z_j \geq 0$ it follows that $x_j > 0$. By applying (60) we see that $z_j = 0$. The proof for (ii) is analogous.

For (iii) suppose, $w_j = \mu$ for some $j \in [n]$. By (62), $x_j - z_j + \mu = \mu$, which implies $x_j = z_j$. By applying (60) we must have $x_j = z_j = 0$. □

We now state Algorithm 5.1 for solving (56). Note that the algorithm computes the KKT multipliers $\mu, \boldsymbol{z}$ as well as $\boldsymbol{x}$ in order to illustrate its correctness, but in the FWLP-P code, the computation of $\boldsymbol{z}$ is not needed.

Algorithm 5.1 An efficient algorithm for finding a KKT solution of (56).

Require: Linear term coefficient $\boldsymbol{w} \in \mathbb{R}^n$.

Sort $\boldsymbol{w}$ and store result in $\bar{\boldsymbol{w}}$, along with the permutation function $\sigma : [n] \to [n]$ which maps the indices of $\bar{\boldsymbol{w}}$ back to their original positions in $\boldsymbol{w}$:

$$(\bar{\boldsymbol{w}}, \sigma) := \text{sort}(\boldsymbol{w}). \tag{63}$$

Compute the cumulative sum of $\bar{\boldsymbol{w}}$:

$$S := \text{cumsum}(\bar{\boldsymbol{w}}). \tag{64}$$

for $j = 1, 2, \ldots, n$ **do**
    Compute

$$\mu := \frac{S(j) - 1}{j} \tag{65}$$

    if $\bar{w}_j \geq \mu$ and either $j = n$, or $\bar{w}_{j+1} \leq \mu$ **then**
        Record index j in the variable J.
        break
    end if
end for
if $\mu \geq 0$ **then**
    Construct a KKT solution such that $\boldsymbol{e}^T\boldsymbol{x} = 1$:

$$\begin{aligned} &\boldsymbol{x} := \boldsymbol{0}, \\ &\boldsymbol{x}(\sigma(1:J)) := \bar{\boldsymbol{w}}(1:J) - \mu \cdot \boldsymbol{e}(1:J), \\ &\boldsymbol{z} := \mu\boldsymbol{e} - \boldsymbol{w}, \\ &\boldsymbol{z}(\sigma(1:J)) := \boldsymbol{0}. \end{aligned} \tag{66}$$

    return
end if
Otherwise, construct a KKT solution such that $\boldsymbol{e}^T\boldsymbol{x} < 1$:

$$\begin{aligned} &\boldsymbol{x} := \max(\boldsymbol{0}, \boldsymbol{w}), \\ &\mu := 0, \\ &\boldsymbol{z} := \boldsymbol{x} - \boldsymbol{w}. \end{aligned} \tag{67}$$

return

We now prove that Algorithm 5.1 is finite and correct.

Theorem 4. *Algorithm 5.1 finds a KKT solution to* (56) *in finite time.*

Proof. It is clear that the algorithm runs in linear time except for the sorting.

Consider the case where $(\boldsymbol{x}, \mu, \boldsymbol{z})$ satisfy the KKT conditions with $\boldsymbol{e}^T\boldsymbol{x} = 1$. Then by Lemma 5 there must exist an index $j \in [n]$ such that $w_j > \mu$, so J must be well-defined on line 6 of the algorithm. Moreover, this means that for all $j \in [J], \bar{w}_j \geq \mu_j$, and by applying Lemma 5 we get that $x_j \geq 0$ and $z_j = 0$ for all $j \in [J]$. A similar argument tells us that $\bar{w}_j < \mu_j$ and thus $x_j = 0, z_j > 0$ for $j \in J' := [n] \setminus [J]$. Using that $\boldsymbol{e}^T\boldsymbol{x} = 1$ and that the entries of $\boldsymbol{x}$ must be zero outside of $[J]$, as well as $z_j = 0$ for all $j \in [J]$, we may multiply (57) by $\boldsymbol{e}_J^T$ and rewrite to obtain:

$$\boldsymbol{e}_J^T\boldsymbol{w} = \boldsymbol{e}_J^T\boldsymbol{x} - \boldsymbol{e}_J^T\boldsymbol{z} + J\mu = 1 + J\mu,$$

where $\boldsymbol{e}_J$ is taken to be the vector with ones in entries $[J]$ and zero everywhere else. Rearranging gives the formula for μ used in Algorithm 5.1:

$$\mu = \frac{S(J) - 1}{J}$$

where we note that $S(J) = \boldsymbol{e}_J^T\boldsymbol{w}$. It follows from assumption of this case $\boldsymbol{e}^T\boldsymbol{x} = 1$ that $\mu \geq 0$ in line 10 of the algorithm. After equations (66), the algorithm has constructed an $\boldsymbol{x}$ such that

$$\boldsymbol{x}_J = \boldsymbol{w}_J - \mu\boldsymbol{e}_J, \quad \boldsymbol{x}_{J'} = \boldsymbol{0},$$

and a $\boldsymbol{z}$ such that

$$\boldsymbol{z}_J = \boldsymbol{0}, \quad \boldsymbol{z}_{J'} = \mu\boldsymbol{e}_{J'} - \boldsymbol{w}_{J'}.$$

It is now easy to check that the $(\boldsymbol{x}, \mu, \boldsymbol{z})$ constructed by the algorithm satisfy the KKT conditions.

Now consider the case where $(\boldsymbol{x}, \mu, \boldsymbol{z})$ satisfy the KKT conditions with $\boldsymbol{e}^T\boldsymbol{x} < 1$. From (59), $\mu = 0$ and clearly the $\boldsymbol{x}$ and $\boldsymbol{z}$ defined in (67) satisfy the remaining KKT conditions. □

The cost of Algorithm 5.1 is bounded below by the cost of either the matrix-vector product $A^T\boldsymbol{y}_k$ in computing $\boldsymbol{w}$ or the sort (63), which take $\mathcal{O}(mn)$ and $\mathcal{O}(n\log(n))$ operations respectively. Meanwhile, (16) takes $\mathcal{O}(n)$ operations, so overall the two run in $\mathcal{O}(n + mn)$ operations if we assume $m \geq \log(n)$.

Algorithm 5.1 could be sped up by noticing that in order to construct $\boldsymbol{x}$,we need only knowledge of the most-violated dual constraints, i.e., those for which $\boldsymbol{c} - A^T\boldsymbol{y}_k$ is most negative. Recall from Section 5.2 that the computation of $A^T\boldsymbol{y}_k$ would take $\mathcal{O}(|\mathcal{S}_k| \cdot m)$ operations, plus the cost of updating the data structure storing the indices $\mathcal{S}_k$. The sorting computation cost would also be decreased since only the largest entries would need sorting.

We now consider the dual updates (17) and (18). Step (17) can be solved by considering the projection form $\text{proj}_\Gamma(\sqrt{k}(\boldsymbol{b} - A\boldsymbol{x}_{k+1}))$ and noting that this has a well known closed-form solution [5, Lemma 6.26]:

$$\boldsymbol{s}_{k+1}(i) = \max(-\eta, \min(\eta, \sqrt{k}[\boldsymbol{b} - A\boldsymbol{x}_{k+1}]_i)),$$

for each $i \in [m]$. The cost of computing $A\boldsymbol{x}_{k+1}$ is again $\mathcal{O}(|\mathcal{S}_k| \cdot m)$ because we can scale $A\boldsymbol{x}_{k-1}$ and then add the update $A\boldsymbol{r}_k/k$. The number of nonzero entries in $\boldsymbol{r}_k$ is $|\mathcal{S}_{k+1}|$, which is the number of entries that partake in the projection.

6 Conclusion

We proposed two primal-dual first-order algorithms, namely FWLP and FWLP-P, for solving linear programming problems and discussed how both algorithms can be implemented in such a way that significantly improves their efficiency, especially for large-scale problems. Our convergence analysis of FWLP-P shows that the algorithm converges to a primal-dual solution with error $\mathcal{O}(1/\sqrt{k})$ after k iterations. Despite this, no convergence proof is known at this time for the simpler and faster algorithm FWLP, and analysis of this algorithm is a topic for future research.

Another interesting question is how FWLP and FWLP-P can cope with primal or dual infeasibility. It should be noted that our proof that $U_k \to 0$ (Theorem 1) did not depend on feasibility nor on the correct choice of ξ and η, so monitoring U_k cannot diagnose these conditions. We remark that other first-order algorithms have a theory for infeasibility detection; see, e.g., [3] and [13]. A related question is how the algorithm can detect whether the two parameters ξ and η of FWLP and FWLP-P have been chosen correctly.

We showed that FWLP-P converges in the sense that the primal and dual infeasibility measures tend to 0, as does the duality gap. However, we did not prove convergence of the iterates. In the case that the LP has multiple optimizers, an open question is whether the algorithms converge to a particular optimizer.

References

[1] D. Applegate, M. Díaz, O. Hinder, H. Lu, M. Lubin, B. O'Donoghue, and W. Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20243–20257. Curran Associates, Inc., 2021.

[2] D. Applegate, O. Hinder, H. Lu, and M. Lubin. Faster first-order primal-dual methods for linear programming using restarts and sharpness. *Mathematical Programming*, 201(1):133–184, Sep 2023.

[3] David Applegate, Mateo Díaz, Haihao Lu, and Miles Lubin. Infeasibility detection with primal-dual hybrid gradient for large-scale linear programming. *SIAM Journal on Optimization*, 34(1):459–484, 2024.

[4] K. Basu, A. Ghoting, R. Mazumder, and Y. Pan. ECLIPSE: An extreme-scale linear program solver for web-applications. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 704–714. PMLR, 13–18 Jul 2020.

[5] Amir Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[6] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, May 2011.

[7] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

[8] Robert M. Freund and Paul Grigas. New analysis and results for the Frank–Wolfe method. *Mathematical Programming*, 155(1):199–230, Jan 2016.

[9] G. Gidel, T. Jebara, and S. Lacoste-Julien. Frank-Wolfe algorithms for saddle point problems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[10] J.H. Hammond. *Solving Asymmetric Variational Inequality Problems and Systems of Equations with Generalized Nonlinear Programming Algorithms*. PhD thesis, Massachussetts Institute of Technology, 1984.

[11] Matthew Hough. Solving saddle point formulations of linear programs with Frank-Wolfe. Master's thesis, University of Waterloo, 2023.

[12] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 427–435, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[13] Tao Jiang, Walaa M Moursi, and Stephen A Vavasis. Range of the displacement operator of PDHG with applications to quadratic and conic programming. *arXiv preprint arXiv:2309.15009*, 2023.

[14] X. Li, D. Sun, and KC Toh. An asymptotically superlinearly convergent semismooth Newton augmented Lagrangian method for linear programming. *SIAM Journal on Optimization*, 30(3):2410–2440, 2020.

[15] Arkadi Nemirovski. Prox-method with rate of convergence O(1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.