Solving Positive LPs in Parallel using the Multiplicative Weights Update (MWU)

Caleb Ju and Navjot Singh

UIUC Parallel Graph Reading Group

October 8, 2020

Overview

1. Definition and Applications
2. MWU For Solving Packing LPs
3. Experimental Results

Definitions

LPs

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $b, \in \mathbb{R}^n$, and $c \in \mathbb{R}^m$. **LP** is

$$\max \langle c, x \rangle \quad \text{s.t.} \quad \boldsymbol{A}x\{\leq, =, \geq\}b.$$

Used to solve maximum graph matchings, set cover, solutions to linear systems, network flow, discrete optimal transport, ...

Definitions

Positive LPs

Let $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$, $b, \in \mathbb{R}_+^n$, and $c \in \mathbb{R}_+^m$. **Packing LP** is

$$\begin{aligned} \max \ & \langle c, x \rangle \\ \text{s.t.} \ & \boldsymbol{A}x \leq b \\ & x \geq 0. \end{aligned}$$

Maximum Matching

Given a graph $G = (V, E)$, find largest cardinality $F \subseteq E$ such that $\forall v \in V$ is incident to at most one edge in F.

Definitions

Positive LPs

Let $\boldsymbol{A} \in \mathbb{R}_{+}^{m \times n}$, $b, \in \mathbb{R}_{+}^{n}$, and $c \in \mathbb{R}_{+}^{m}$. **Packing LP** is

$$\begin{aligned} \max\ & \langle c, x \rangle \\ \text{s.t. } & \boldsymbol{A}x \leq b \\ & x \geq 0. \end{aligned}$$

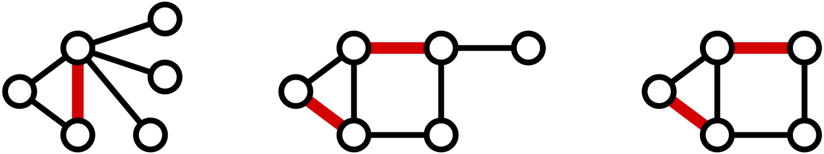
Maximum Matching

Given a graph $G = (V, E)$, find largest cardinality $F \subseteq E$ such that $\forall v \in V$ is incident to at most one edge in F.

$$\max \sum_{e \in E} x_e \text{ s.t. } \sum_{e \in \text{inc}(v)} x_e \leq 1 \ \forall v \in V$$

$$x \geq \mathbb{0}.$$

Graph Matching



Positive LPs

Let $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$, $b, \in \mathbb{R}_+^n$, and $c \in \mathbb{R}_+^m$. **Covering LP** is

$$\begin{aligned} \min \ & \langle c, x \rangle \\ \text{s.t.} \ & \boldsymbol{A}x \geq b \\ & x \geq 0. \end{aligned}$$

Dominating Set

Given a graph $G = (V, E)$, find the smallest subset $D \subseteq V$ then $\forall v \in V$ s.t. $\big(\{v\} \cup N(v)\big) \cap D \neq \emptyset$.

Positive LPs

Let $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$, $b, \in \mathbb{R}_+^n$, and $c \in \mathbb{R}_+^m$. **Covering LP** is

$$\begin{aligned} \min \ & \langle c, x \rangle \\ \text{s.t.} \ & \boldsymbol{A}x \geq b \\ & x \geq 0. \end{aligned}$$

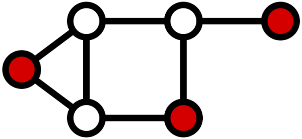
Dominating Set

Given a graph $G = (V, E)$, find the smallest subset $D \subseteq V$ then $\forall v \in V$ s.t. $\big(\{v\} \cup N(v)\big) \cap D \neq \emptyset$.

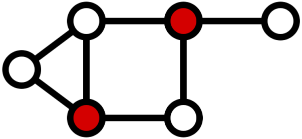
$$\begin{aligned} \min \sum_{v \in V} x_v \ \text{s.t.} & \\ & x_v + \sum_{u \in N(v)} x_u \geq 1 \ \forall v \in V \\ & x \geq \mathbb{0}. \end{aligned}$$

Dominating Set

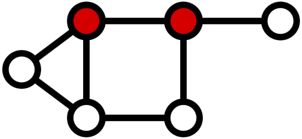
(a)



(b)



(c)



Positive LPs

Positive LPs

Let $\boldsymbol{P}, \boldsymbol{C} \in \mathbb{R}_+^{m \times n}$, $b, c \in \mathbb{R}_+^m$. **Mixed Packing Covering LP** is

$$\begin{aligned} &\exists x \\ \text{s.t. } &\boldsymbol{P}x \leq b \\ &\boldsymbol{C}x \geq c \\ &x \geq 0. \end{aligned}$$

Solving a Positive Linear System of Equations

Find x s.t. $\boldsymbol{A}x = b$

Positive LPs

Positive LPs

Let $\boldsymbol{P}, \boldsymbol{C} \in \mathbb{R}_+^{m \times n}$, $b, c \in \mathbb{R}_+^m$. **Mixed Packing Covering LP** is

$$\begin{aligned} & \exists x \\ \text{s.t. } & \boldsymbol{P}x \leq b \\ & \boldsymbol{C}x \geq c \\ & x \geq 0. \end{aligned}$$

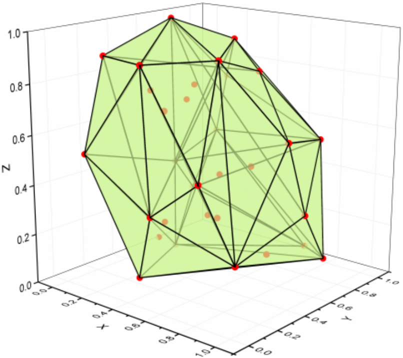
Solving a Positive Linear System of Equations

Find x s.t. $\boldsymbol{A}x = b$

$$\begin{aligned} \exists x \text{ s.t. } & \boldsymbol{A}x \leq b \\ \boldsymbol{A}x & \geq b \\ x & \geq 0. \end{aligned}$$

Simplex Method

- Looks at all adjacent boundary pts in the feasibility region
- Exact solution
- Requires feasible initial guess
- Exponential time



IPM Intuition

IPM visualised as barrier method for inequality constraints

$$\max_x c^T x + \mu_1 \log \sum_i (b_i - a_i^T x) + \mu_2 \log \sum_j x_j$$

where a_i is the i^{th} row of A

- Uses Newton's iteration to compute steps
- Expensive as requires solve

MWU Intuition

- Can we do something cheaper?
- Look at subclass of LP problems with special properties... ?
- Contribution of each step depends on closeness to violation
- Above observation with positivity constraints resulted in MWU algorithm
- Used as exponentiated gradient descent in solving KL-div objective

Methods for Solving Positive LPs

(Normal) Packing LP

Let $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$ **Normal Packing LP** is

$$\max \langle \mathbb{c}, x \rangle \text{ s.t. } \boldsymbol{A}x \leq b, x \geq 0$$
$$\rightarrow \max \langle \mathbb{1}, x \rangle \text{ s.t. } \boldsymbol{A}x \leq \mathbb{1}, x \geq 0.$$

Approximation

A positive LP produces an ϵ-approximation answer if

$$\mathbb{1}^\mathsf{T} x \geq \mathrm{OPT} \text{ and } \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$$

Methods for Solving Positive LPs

Approximate (Normal) Packing LP

Let $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$ **Normal Packing LP** is

$$\text{Find } x \text{ s.t. } \langle \mathbb{1}, x \rangle \geq \mathrm{OPT} \text{ and } \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}, x \geq 0.$$

How to solve quickly, accurately, and in parallel?

- Simplex algorithm: Exact but exponential

Methods for Solving Positive LPs

Approximate (Normal) Packing LP

Let $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$ **Normal Packing LP** is

$$\text{Find } x \text{ s.t. } \langle \mathbb{1}, x \rangle \geq \mathrm{OPT} \text{ and } \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}, x \geq 0.$$

How to solve quickly, accurately, and in parallel?

- Simplex algorithm: Exact but exponential
- Interior point: $O(n^3 k)$ where k is the number of iterations

Methods for Solving Positive LPs

Approximate (Normal) Packing LP

Let $\boldsymbol{A} \in \mathbb{R}_{+}^{m \times n}$ **Normal Packing LP** is

$$\text{Find } x \text{ s.t. } \langle \mathbb{1}, x \rangle \geq \mathrm{OPT} \text{ and } \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}, x \geq 0.$$

How to solve quickly, accurately, and in parallel?

- Simplex algorithm: Exact but exponential
- Interior point: $O(n^3 k)$ where k is the number of iterations
- Approximately with MWU:

Methods for Solving Positive LPs

Approximate (Normal) Packing LP

Let $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$ **Normal Packing LP** is

$$\text{Find } x \text{ s.t. } \langle \mathbb{1}, x \rangle \geq \mathrm{OPT} \text{ and } \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}, x \geq 0.$$

How to solve quickly, accurately, and in parallel?

- Simplex algorithm: Exact but exponential
- Interior point: $O(n^3 k)$ where k is the number of iterations
- Approximately with MWU:
  1. sequentially in $\mathcal{O}(\mathrm{polylog}(n) N/\epsilon)$ time where $N = \mathrm{nnz}(\boldsymbol{A})$

Methods for Solving Positive LPs

Approximate (Normal) Packing LP

Let $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$ **Normal Packing LP** is

$$\text{Find } x \text{ s.t. } \langle \mathbb{1}, x \rangle \geq \mathrm{OPT} \text{ and } \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}, x \geq 0.$$

How to solve quickly, accurately, and in parallel?

- Simplex algorithm: Exact but exponential
- Interior point: $O(n^3 k)$ where k is the number of iterations
- Approximately with MWU:
  1. sequentially in $\mathcal{O}(\mathrm{polylog}(n)N/\epsilon)$ time where $N = \mathrm{nnz}(\boldsymbol{A})$
  2. in $\mathcal{O}(\log(m)\log(n/\epsilon)/\epsilon^2)$ iterations in parallel

MWU Framework

Recall $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$

Optimization Problem (Packing LP)

Solve $\max\langle \mathbb{1}, x \rangle$ s.t. $\boldsymbol{A}x \leq \mathbb{1}$, $x \geq \mathbb{0}$

Find x such that $\mathbb{1}^\mathsf{T} x \geq \mathrm{OPT}, \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$

MWU Framework

Recall $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$

Optimization Problem (Packing LP)

Solve $\max\langle \mathbb{1}, x \rangle$ s.t. $\boldsymbol{A}x \leq \mathbb{1}$, $x \geq \mathbb{0}$

Find x such that $\mathbb{1}^\mathsf{T} x \geq \mathrm{OPT}, \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$

MWU Framework

Recall $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$

Optimization Problem (Packing LP)

Solve $\max\langle \mathbb{1}, x\rangle$ s.t. $\boldsymbol{A}x \leq \mathbb{1}$, $x \geq \mathbb{0}$

Find x such that $\mathbb{1}^\mathsf{T} x \geq \mathrm{OPT}, \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$

MWU Framework

Recall $\boldsymbol{A} \in \mathbb{R}_{+}^{m \times n}$

Optimization Problem (Packing LP)

Solve $\max\langle \mathbb{1}, x\rangle$ s.t. $\boldsymbol{A}x \leq \mathbb{1}$, $x \geq \mathbb{0}$

Find x such that $\mathbb{1}^{\mathsf{T}}x \geq \mathrm{OPT}, \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^{n}$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^{m}$
3. Solve Lagrangian relaxation, $\max\langle \mathbb{1}, d\rangle$ s.t. $w^{\mathsf{T}}\boldsymbol{A}d \leq w^{\mathsf{T}}\mathbb{1}$, $d \geq \mathbb{0}$

MWU Framework

Recall $\boldsymbol{A} \in \mathbb{R}_{+}^{m \times n}$

Optimization Problem (Packing LP)

Solve $\max \langle \mathbb{1}, x \rangle$ s.t. $\boldsymbol{A}x \le \mathbb{1}$, $x \ge \mathbb{0}$

Find x such that $\mathbb{1}^{\mathsf{T}} x \ge \mathrm{OPT}, \boldsymbol{A}x \le (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Solve Lagrangian relaxation, $\max \langle \mathbb{1}, d \rangle$ s.t. $w^{\mathsf{T}} \boldsymbol{A} d \le w^{\mathsf{T}} \mathbb{1}$, $d \ge \mathbb{0}$
4. Increment $x \leftarrow x + \delta \cdot d$ for small δ

MWU Framework

Recall $\boldsymbol{A} \in \mathbb{R}_+^{m \times n}$

Optimization Problem (Packing LP)

Solve $\max\langle \mathbb{1}, x \rangle$ s.t. $\boldsymbol{A}x \le \mathbb{1}$, $x \ge \mathbb{0}$

Find x such that $\mathbb{1}^\mathsf{T} x \ge \mathrm{OPT}, \boldsymbol{A}x \le (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Solve Lagrangian relaxation, $\max\langle \mathbb{1}, d \rangle$ s.t. $w^\mathsf{T} \boldsymbol{A} d \le w^\mathsf{T} \mathbb{1}$, $d \ge \mathbb{0}$
4. Increment $x \leftarrow x + \delta \cdot d$ for small δ
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$

MWU Framework

Recall $\boldsymbol{A} \in \mathbb{R}_{+}^{m \times n}$

Optimization Problem (Packing LP)

Solve $\max\langle \mathbb{1}, x \rangle$ s.t. $\boldsymbol{A}x \leq \mathbb{1}$, $x \geq \mathbb{0}$

Find x such that $\mathbb{1}^{\mathsf{T}} x \geq \mathrm{OPT}, \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Solve Lagrangian relaxation, $\max\langle \mathbb{1}, d \rangle$ s.t. $w^{\mathsf{T}}\boldsymbol{A}d \leq w^{\mathsf{T}}\mathbb{1}$, $d \geq \mathbb{0}$
4. Increment $x \leftarrow x + \delta \cdot d$ for small δ
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

Solving Lagrangian Relaxation

Lagrangian relaxation

$$\max\langle \mathbb{1}, d\rangle \text{ s.t. } w^{\mathsf{T}}\boldsymbol{A}d \leq w^{\mathsf{T}}\mathbb{1},\ d \geq \mathbb{0}.$$

Solving Lagrangian Relaxation

Lagrangian relaxation

$$\max\langle \mathbb{1}, d\rangle \text{ s.t. } w^\mathsf{T} \boldsymbol{A} d \le w^\mathsf{T}\mathbb{1},\ d \ge \mathbb{0}.$$

Knapsack

Equivalent to

$$\max\langle \mathbb{1}, d\rangle \text{ s.t. } \left\langle \underbrace{\frac{\boldsymbol{A}^\mathsf{T} w}{\langle \mathbb{1}, w\rangle}}_{g}, d \right\rangle \le 1, \text{ or}$$

$$\max d_1 + \ldots + d_n$$

$$\text{s.t. } g_1 \cdot d_1 + \ldots + g_n \cdot d_n \le 1.$$

Set $d = e_i$ s.t. $i = \mathrm{argmin}_j g_j$.

MWU Framework

Optimization Problem (Packing LP)

Solve $\max\langle x, \mathbb{1}\rangle$ s.t. $\boldsymbol{A}x \leq \mathbb{1}$, $d \geq \mathbb{0}$

Find x such that $\mathbb{1}^{\mathsf{T}}x \geq \mathrm{OPT}, \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \mathrm{argmin}_j\ g_j$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \delta \cdot d$ for small δ
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

MWU Framework

Optimization Problem (Packing LP)

Solve $\max\langle x, \mathbb{1}\rangle$ s.t. $\boldsymbol{A}x \leq \mathbb{1}$, $d \geq \mathbb{0}$

Find x such that $\mathbb{1}^{\mathsf{T}}x \geq \mathrm{OPT}, \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \operatorname{argmin}_j g_j$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \delta \cdot d$ s.t. $\delta \cdot \max_i \left(\eta \cdot \boldsymbol{A}_i d\right) = \epsilon$
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

MWU Framework

Find x such that $\mathbb{1}^\mathsf{T} x \geq \mathrm{OPT}, \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \mathrm{argmin}_j\ g_j$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \delta \cdot d$ s.t. $\delta \cdot \max_i \left(\eta \cdot \boldsymbol{A}_i d\right) = \epsilon$
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

MWU Framework

Find x such that $\mathbb{1}^{\mathsf{T}} x \geq \mathrm{OPT}, \boldsymbol{A} x \leq (1+\epsilon)\mathbb{1}$

1. $x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \mathrm{argmin}_j\ g_j$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \delta \cdot d$ s.t. $\delta \cdot \max_i \left(\eta \cdot \boldsymbol{A}_i d\right) = \epsilon$
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

Lemma

Step (3) is called at most $\mathcal{O}(m \cdot \eta/\epsilon) = \mathcal{O}(m \log(m)/\epsilon^2)$ *times.*

MWU Example

Consider LP $\max x_1 + x_2$

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

where $x^* = \begin{pmatrix} 2/5 & 1/5 \end{pmatrix}^\mathsf{T}$. Choose $\epsilon = 0.1$.

- (1,2): Initialize $x \leftarrow \mathbb{0}$, $w \leftarrow \frac{1}{2}\mathbb{1}$, $\eta \leftarrow 10$

MWU Example

Consider LP $\max x_1 + x_2$

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

where $x^* = \begin{pmatrix} 2/5 & 1/5 \end{pmatrix}^\mathsf{T}$. Choose $\epsilon = 0.1$.

- (1,2): Initialize $x \leftarrow \mathbb{0}$, $w \leftarrow \frac{1}{2}\mathbb{1}$, $\eta \leftarrow 10$
- (3): Solve the Lagrangian, $\text{argmin } \frac{w^\mathsf{T}\boldsymbol{A}}{\mathbb{1}^\mathsf{T} w} = \begin{pmatrix} 1.5 & 2 \end{pmatrix}$. Set $d = e_1$

MWU Example

Consider LP $\max x_1 + x_2$

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

where $x^* = \begin{pmatrix} 2/5 & 1/5 \end{pmatrix}^\mathsf{T}$. Choose $\epsilon = 0.1$.

- (1,2): Initialize $x \leftarrow \mathbb{0}$, $w \leftarrow \frac{1}{2}\mathbb{1}$, $\eta \leftarrow 10$
- (3): Solve the Lagrangian, $\mathrm{argmin}\ \frac{w^\mathsf{T} \boldsymbol{A}}{\mathbb{1}^\mathsf{T} w} = \begin{pmatrix} 1.5 & 2 \end{pmatrix}$. Set $d = e_1$
- (4): Increment $x \leftarrow x + \frac{1}{4} d = \begin{pmatrix} \frac{1}{4} & 0 \end{pmatrix}^\mathsf{T}$

MWU Example

Consider LP $\max x_1 + x_2$

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

where $x^* = \begin{pmatrix} 2/5 & 1/5 \end{pmatrix}^\mathsf{T}$. Choose $\epsilon = 0.1$.

- (1,2): Initialize $x \leftarrow \mathbb{0}$, $w \leftarrow \frac{1}{2}\mathbb{1}$, $\eta \leftarrow 10$
- (3): Solve the Lagrangian, $\mathrm{argmin}\ \frac{w^\mathsf{T}\boldsymbol{A}}{\mathbb{1}^\mathsf{T} w} = \begin{pmatrix} 1.5 & 2 \end{pmatrix}$. Set $d = e_1$
- (4): Increment $x \leftarrow x + \frac{1}{4} d = \begin{pmatrix} \frac{1}{4} & 0 \end{pmatrix}^\mathsf{T}$
- (5): Update weights, $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \odot \exp\Big(\eta \boldsymbol{A} d\Big) \approx \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \odot \begin{bmatrix} 9 \\ 1 \end{bmatrix}$

MWU Example

Consider LP $\max x_1 + x_2$

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

where $x^* = \begin{pmatrix} 2/5 & 1/5 \end{pmatrix}^\mathsf{T}$. Choose $\epsilon = 0.1$.

- (1,2): Initialize $x \leftarrow \mathbb{0}$, $w \leftarrow \frac{1}{2}\mathbb{1}$, $\eta \leftarrow 10$
- (3): Solve the Lagrangian, $\operatorname{argmin} \frac{w^\mathsf{T} \boldsymbol{A}}{\mathbb{1}^\mathsf{T} w} = \begin{pmatrix} 1.5 & 2 \end{pmatrix}$. Set $d = e_1$
- (4): Increment $x \leftarrow x + \frac{1}{4} d = \begin{pmatrix} \frac{1}{4} & 0 \end{pmatrix}^\mathsf{T}$
- (5): Update weights, $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \odot \exp\Big(\eta \boldsymbol{A} d\Big) \approx \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \odot \begin{bmatrix} 9 \\ 1 \end{bmatrix}$
- (3): Solve the Lagrangian, $\operatorname{argmin} \frac{w^\mathsf{T} \boldsymbol{A}}{\mathbb{1}^\mathsf{T} w} = \begin{pmatrix} \frac{19}{10} & \frac{12}{10} \end{pmatrix}$. Set $d = e_2$

MWU Example

Consider LP $\max x_1 + x_2$

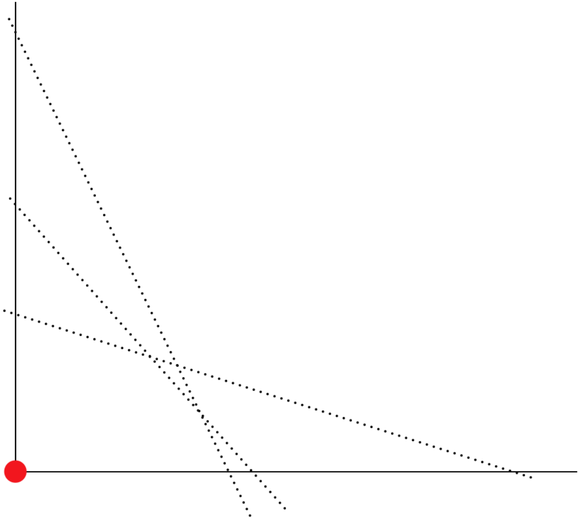
$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

where $x^* = \begin{pmatrix} 2/5 & 1/5 \end{pmatrix}^\mathsf{T}$. Choose $\epsilon = 0.1$.

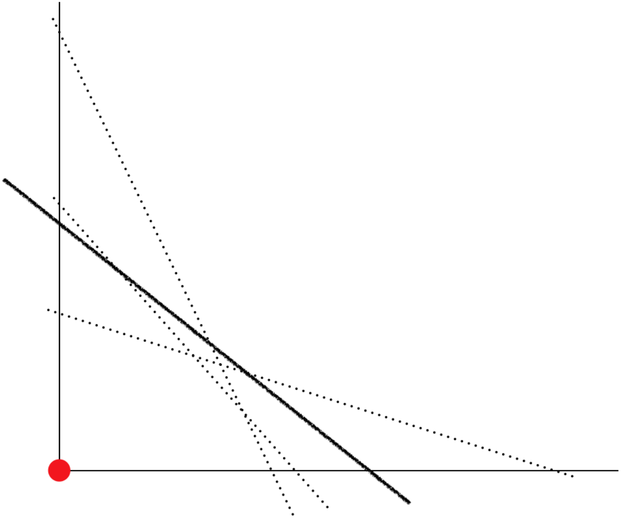
- (1,2): Initialize $x \leftarrow \mathbb{0}$, $w \leftarrow \frac{1}{2}\mathbb{1}$, $\eta \leftarrow 10$
- (3): Solve the Lagrangian, $\operatorname{argmin} \frac{w^\mathsf{T}\boldsymbol{A}}{\mathbb{1}^\mathsf{T} w} = \begin{pmatrix} 1.5 & 2 \end{pmatrix}$. Set $d = e_1$
- (4): Increment $x \leftarrow x + \frac{1}{4}d = \begin{pmatrix} \frac{1}{4} & 0 \end{pmatrix}^\mathsf{T}$
- (5): Update weights, $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \odot \exp\Big(\eta \boldsymbol{A} d\Big) \approx \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \odot \begin{bmatrix} 9 \\ 1 \end{bmatrix}$
- (3): Solve the Lagrangian, $\operatorname{argmin} \frac{w^\mathsf{T}\boldsymbol{A}}{\mathbb{1}^\mathsf{T} w} = \begin{pmatrix} \frac{19}{10} & \frac{12}{10} \end{pmatrix}$. Set $d = e_2$
- $\ldots$

MWU Example

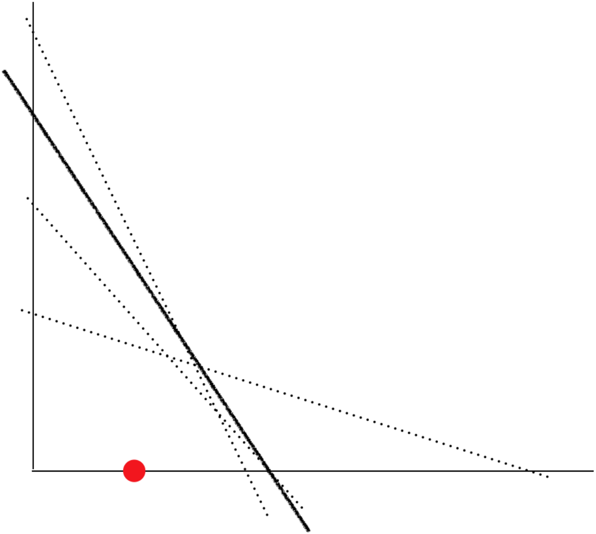
Consider standard LP with $m = 3$ constraints and $n = 2$ vars.



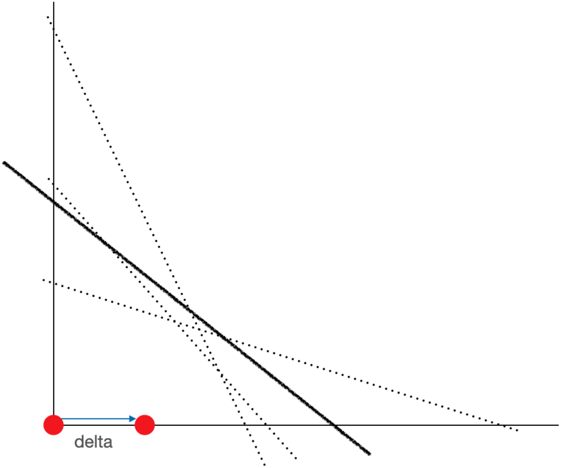
MWU Example



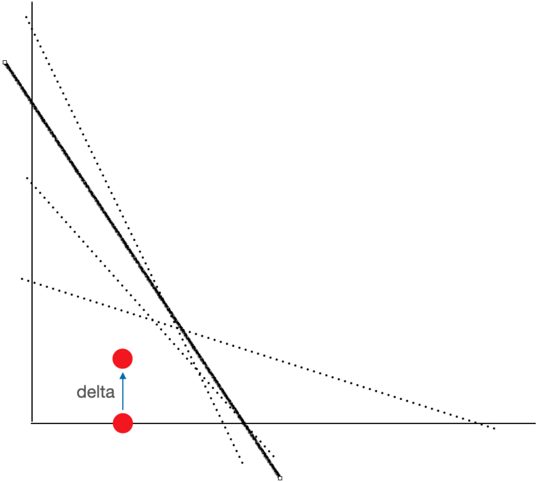
MWU Example



MWU Example



MWU Example



MWU Framework

Find x such that $x \geq \mathrm{OPT}, \boldsymbol{A}x \leq (1+\epsilon)\mathbb{1}$

1. $\mathbb{1}^{\mathsf{T}}x \leftarrow \mathbb{0}^n$, $\eta \leftarrow \log(m)/\epsilon$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \mathrm{argmin}_j\, g_j$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \delta \cdot d$ s.t. $\delta \cdot \max_i \left(\eta \cdot \boldsymbol{A}_i d\right) = \epsilon$
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

Sequential $\rightarrow$ Parallel

How can we parallelize this (i.e., increase multiple coordinates simultaneously)?

Parallelizing MWU

Lagrangian Relaxation

Equivalent to

$$\max\langle \mathbb{1}, d\rangle \text{ s.t. } \Big\langle \underbrace{\frac{\boldsymbol{A}^{\mathsf{T}} w}{\langle \mathbb{1}, w\rangle}}_{g}, d \Big\rangle \leq 1, , d \geq \mathbb{0}.$$

Sequentially, set $d = e_i$ s.t. $i = \mathrm{argmin}_j g_j$.

Parallelizing MWU

Lagrangian Relaxation

Equivalent to

$$\max\langle \mathbb{1}, d\rangle \text{ s.t. } \left\langle \underbrace{\frac{\boldsymbol{A}^{\mathsf{T}} w}{\langle \mathbb{1}, w\rangle}}_{g}, d \right\rangle \leq 1, , d \geq \mathbb{0}.$$

~~Sequentially, set $d = e_i$ s.t. $i = \operatorname{argmin}_j g_j$~~ Set

$$d_i = \begin{cases} f\left(g_i^{-1}\right) \text{ or } 1 - f\left(g_i\right) & : g_i < (1+\epsilon) g_{\min} \\ 0 & : g_i \geq (1+\epsilon) g_{\min} \end{cases}.$$

Parallelizing MWU

Lagrangian Relaxation

Equivalent to

$$\max\langle \mathbb{1}, d\rangle \text{ s.t. } \left\langle \underbrace{\frac{\mathbf{A}^\top w}{\langle \mathbb{1}, w\rangle}}_{g}, d\right\rangle \le 1, , d \ge \mathbb{0}.$$

Set

$$d_i = \begin{cases} f\left(g_i^{-1}\right) \text{ or } 1 - f\left(g_i\right) & : g_i < (1+\epsilon)\cancel{g_{\min}}\ \mathrm{OPT}^{-1} \\ 0 & : g_i \ge (1+\epsilon)\cancel{g_{\min}}\ \mathrm{OPT}^{-1} \end{cases}.$$

Parallelizing MWU

Lagrangian Relaxation

Equivalent to

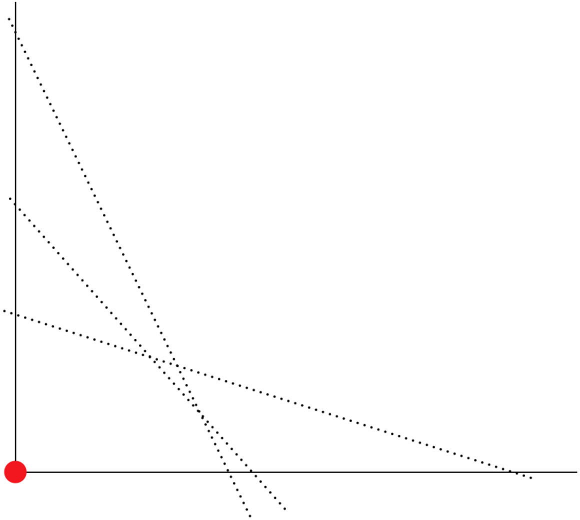
$$\max\langle \mathbb{1}, d\rangle \text{ s.t. } \left\langle \underbrace{\frac{\boldsymbol{A}^{\mathsf{T}} w}{\langle \mathbb{1}, w\rangle}}_{g}, d \right\rangle \leq 1, , d \geq \mathbb{0}.$$

Set

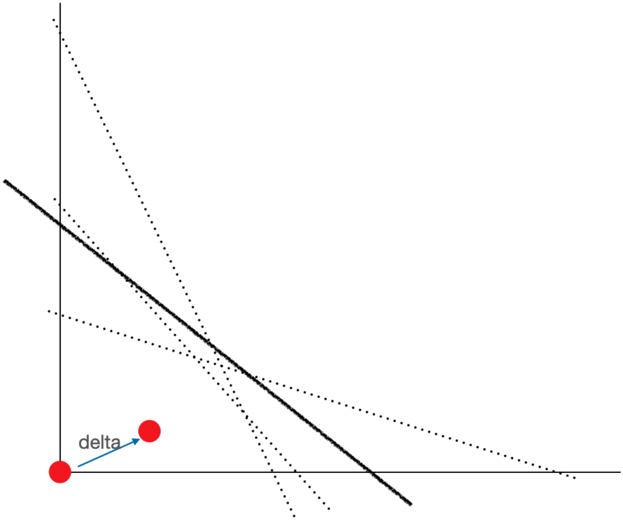
$$d_i = \begin{cases} f\left(g_i^{-1}\right) \text{ or } 1 - f\left(g_i\right) & : g_i < (1+\epsilon)\cancel{g_{\min}}\ \mathrm{OPT}^{-1} \\ 0 & : g_i \geq (1+\epsilon)\cancel{g_{\min}}\ \mathrm{OPT}^{-1} \end{cases}.$$

Since $g_{\min} \cdot \mathbb{1}^{\mathsf{T}} x^* \leq g^{\mathsf{T}} x^* \leq 1$, then $g_{\min} \leq \mathrm{OPT}^{-1}$

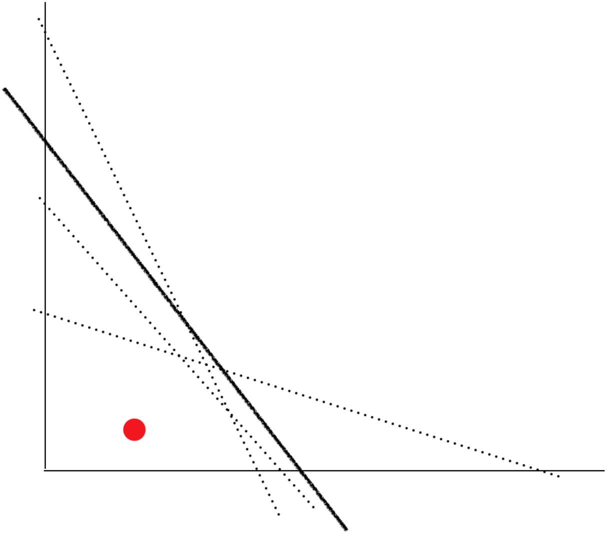
MWU Example (Parallel)



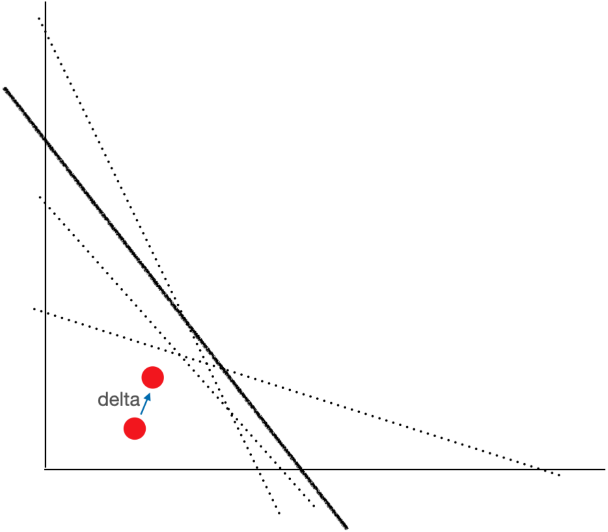
MWU Example (Parallel)



MWU Example (Parallel)



MWU Example (Parallel)



MWU Framework

1. $x > \mathbb{0}$ s.t. $\boldsymbol{A}x \leq \epsilon$, $\eta \leftarrow \log(m)/\epsilon$, $M \leftarrow (1-\epsilon)\mathrm{OPT}$
2. Initialize weights equally* $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max\{0, 1 - g_i M\}$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \underbrace{\eta^{-1} x \odot \Delta}_{d}$
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

MWU Framework

1. $x > \mathbb{0}$ s.t. $\boldsymbol{A}x \leq \epsilon$, $\eta \leftarrow \log(m)/\epsilon$, $M \leftarrow (1-\epsilon)\mathrm{OPT}$
2. Initialize weights equally* $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max\{0, 1 - g_i M\}$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \underbrace{\eta^{-1} x \odot \Delta}_{d}$
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

*Technically, set $w = \exp(\eta \boldsymbol{A} x)$

MWU Framework

1. $x > \mathbb{0}$ s.t. $\boldsymbol{A}x \leq \epsilon$, $\eta \leftarrow \log(m)/\epsilon$, $M \leftarrow (1-\epsilon)\mathrm{OPT}$
2. Initialize weights equally* $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max\{0, 1 - g_i M\}$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \underbrace{\eta^{-1} x \odot \Delta}_{d}$
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

*Technically, set $w = \exp(\eta \boldsymbol{A} x)$

Lemma (Fake)

*Each iteration satisfies the invariant**,*

$$\frac{\langle \mathbb{1}, d\rangle}{\max(\boldsymbol{A}x^{(new)}) - \max(\boldsymbol{A}x^{(old)})} \geq M.$$

MWU Framework

1. $x > \mathbb{0}$ s.t. $\boldsymbol{A}x \leq \epsilon$, $\eta \leftarrow \log(m)/\epsilon$, $M \leftarrow (1-\epsilon)\mathrm{OPT}$
2. Initialize weights equally* $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max\{0, 1 - g_i M\}$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \underbrace{\eta^{-1} x \odot \Delta}_{d}$
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)\ \forall i$
6. If constraints are not tight, go to step (3)

*Technically, set $w = \exp(\eta \boldsymbol{A} x)$

Lemma (Fake)

*Each iteration satisfies the invariant**,*

$$\frac{\langle \mathbb{1}, d \rangle}{\max(\boldsymbol{A}x^{(new)}) - \max(\boldsymbol{A}x^{(old)})} \geq M.$$

***Hard to analyze since* max *is not smooth*

MWU Framework

1. $x_i \leftarrow \frac{\epsilon}{n \cdot \left\| \boldsymbol{A}_{:,i} \right\|}$ $\forall i \in [n]$, $\eta \leftarrow \log(m)/\epsilon$, $M \leftarrow (1-\epsilon)\mathrm{OPT}$
2. Initialize weights equally $w \leftarrow \frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max\left\{0, 1 - g_i M\right\}$ (Lagrangian relaxation)
4. Increment $x \leftarrow x + \underbrace{\eta^{-1} x \odot \Delta}_{d}$
5. Update weights, $w_i = w_i \cdot \exp(\eta \cdot \boldsymbol{A}_i d)$ $\forall i$
6. If constraints are not tight, go to step (3)

Lemma

Each iteration satisfies the invariant,

$$\frac{\langle \mathbb{1}, d \rangle}{\mathrm{smax}(\boldsymbol{A}x^{(new)}) - \mathrm{smax}(\boldsymbol{A}x^{(old)})} \geq M.$$

Parallelizing MWU

Lemma

The algorithm converges in $\mathcal{O}(\log(m)\log(n/\epsilon)/\epsilon^2)$ *iterations.*

Lemma

The algorithm converges to x s.t. $\max(\boldsymbol{A}x) \leq \mathrm{smax}_\eta(\boldsymbol{A}x) \leq 1$.

Lemma

The algorithm converges to x s.t. $\langle \mathbb{1}, x \rangle \geq M$.

MWU Framework

Recap

- Approximately solve special LPs using MWU up to error ϵ
- Reduce “hard” LP into series of ”easy” LPs via Lagrangian relaxation
- Parallelize by doing more work simultaneously

Experimental Results

Implemented in Python using sparse matrices (`scipy`). Run parallel LP solver (denote as Alina's alg) on personal laptop (2.3 GHz Dual-Core Intel i5, 8GB of memory) to solve:

- Maximum matchings (packing LP)*
- Dominating set (covering LP)

Compare with other par/seq. LP solvers:

- Mahoney et al* mixed-PC LP solver
- Kent's sequential solver*

Also run against general optimization libraries:

- CPLEX*
- CVXOPT
- MS-BFS-Graft (for bipartite graph matchings)

*Results included in these slides

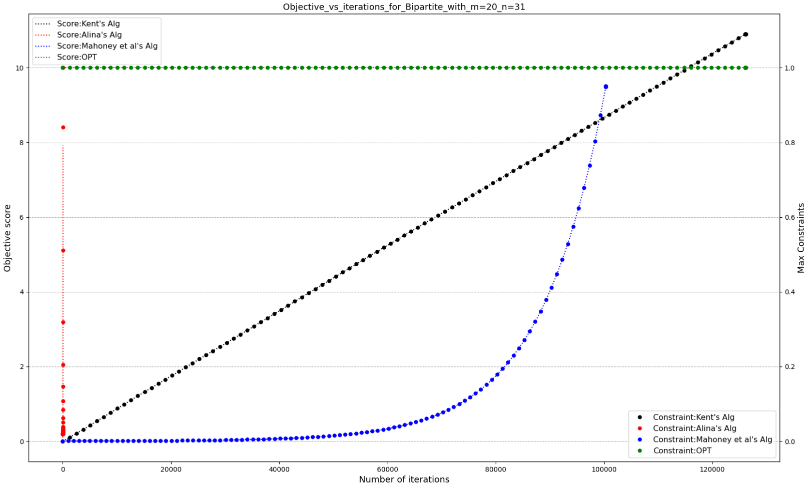
Problem Definition

Maximum Matching

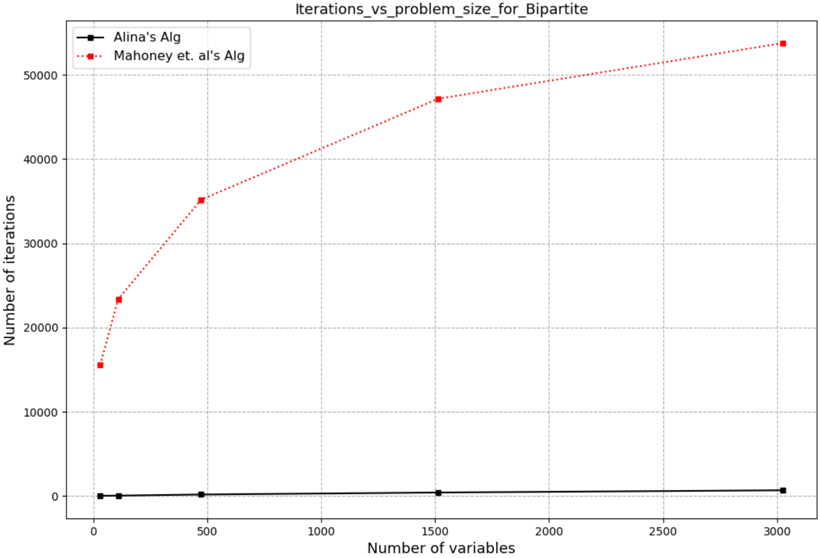
Given a graph $G = (V, E)$, find largest cardinality $F \subseteq E$ such that $\forall v \in V$ is incident to at most one edge in F.

$$\max \sum_{e \in E} x_e \text{ s.t.}$$
$$\sum_{e \in \text{inc}(v)} x_e \leq 1 \; \forall v \in V$$
$$x \geq \mathbb{0}.$$

Iteration Count Comparison



Iteration Count Growth Comparison

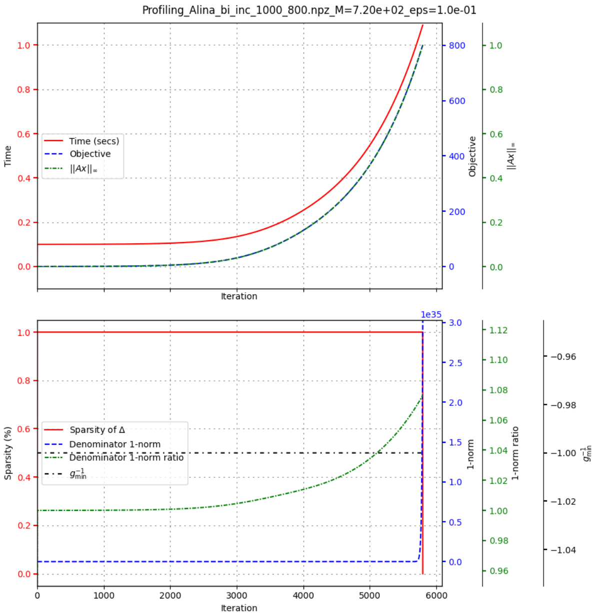


Experimental Results

Table: CPLEX vs. `ParPacLP` runtime and iteration count for graph matching with $\epsilon = 0.1$. Breakdown is (`matvec`,`vecop`)

Problem	**CPLEX**	**LP**	**LP Arith.**	**LP Iters**
bi_20_30	0.0s	0.445s	(0.18s,0.12s)	2215
bi_200_300	0.45s	4.73s	(1.73s,1.70s)	4350
bi_800_1000	1.66s	106s	(40.3s,45.1s)	5800

Iteration Count Growth Comparison



Increasing Step Sizes (Safely)

Can we take larger steps?

Is the step size,

$$d = \eta^{-1}(x \odot \Delta)$$

too conservative?

Solution is optimal (i.e., $\langle \mathbb{1}, x \rangle \geq M$) as long as

$$\frac{\langle \mathbb{1}, x + d \rangle - \langle \mathbb{1}, x \rangle}{\mathrm{smax}_\eta\big(\boldsymbol{A}(x + d)\big) - \mathrm{smax}_\eta\big(\boldsymbol{A}x\big)} = \frac{\langle \mathbb{1}, d \rangle}{\delta t} \geq M$$

Increasing Step Sizes (Safely)

Can we take larger steps?

Is the step size,

$$d = \eta^{-1}(x \odot \Delta)$$

too conservative?

Solve line search: $\max \alpha$ s.t.

$$\frac{\alpha\langle \mathbb{1}, d\rangle}{\mathrm{smax}_\eta\big(\boldsymbol{A}(x+\alpha\cdot d)\big) - \mathrm{smax}_\eta\big(\boldsymbol{A}x\big)} \geq M$$

$$\max\big(\boldsymbol{A}(x+\alpha\cdot d)\big) \leq 1.$$

Binary search over α.

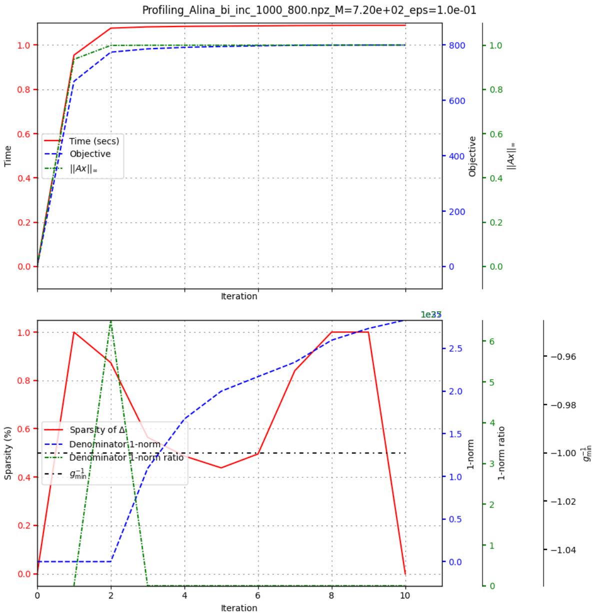
Table: CPLEX vs. ParPacLP+BS runtime and iteration count for graph matching with $\epsilon = 0.1$. Breakdown is (matvec,vecop,BS)

Problem	CPLEX	LP	LP Arith.	LP Iters
bi_800_1000	1.66s	0.26s	(0.1,0.1,0.0)	1
bi_2000_3000	17.52s	1.93s	(0.6,0.7,0.4)	8
A_2000_10	0.46s	0.07s	(0.0,0.0,0.0)	17
A_10000_16	38.17s	1.54s	(0.4,0.6,0.3)	16
wiki-Vote	0.44s	0.18s	(0.0,0.0,0.1)	25
amazon0312	1766.7s	23.6s	(6.2,5.5,9.8)	52
web-Google	173.2s	50.0s	(14.6,9.9,21.8)	49
cit-Patents	8171.5s	306s	(88.8,78.0,118)	62

Table: Dimensions of graphs from SuiteSparse

Problem	m	n	nnz
wiki-Vote	1.7e4	2.1e6	4.1e5
amazon0312	8.0e5	6.4e6	1.3e7
web-Google	1.8e6	1.0e7	2.0e7
cit-Patents	7.5e6	3.3e7	6.6e7

Experimental Results



Next Steps and Questions

- Binary step search takes the most time. Find ways to reduce the time.
- Without M, we need binary search to approximate OPT. Can we avoid/reduce this overhead?
- The parallel algorithm spends majority of time in initial stages. Can we accelerate this?
- Pair/couple LP with another LP solver that is fast in the beginning
- Solving an "easier" problem (subset of constraint, loosen constraints, etc.)

Tuning opportunities

- Introduce ϵ_1, ϵ_2 where $M = (1 - \epsilon_1)\mathrm{OPT}$ and $\epsilon \leftarrow \epsilon_2$
- We can incorporate "easier" problem's solution by scaling down. Opportunities to trade-off how much time is spent in "easier" problem vs. parallel LP.

References

- Arora, Sanjeev, Elad Hazan, and Satyen Kale. "The multiplicative weights update method: a meta-algorithm and applications." Theory of Computing 8.1 (2012): 121-164.
- Ene, Alina, personal communications.
- Mahoney, Michael W., et al. "Approximating the Solution to Mixed Packing and Covering LPs in Parallel $\tilde{O}(\epsilon^{-3})$ Time." 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
- Quanrud, Kent. Fast approximations for combinatorial optimization via multiplicative weight updates. Diss. University of Illinois at Urbana-Champaign, 2019.