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

Definitions

LPs

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

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

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

Definitions

Positive LPs

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

$$\begin{aligned} \max\ & \langle c, x \rangle \\ \text{s.t. } Ax &\le b \\ x &\ge 0. \end{aligned}$$

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

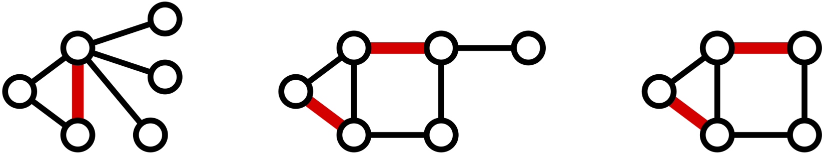
$$\begin{aligned} \max\ & \langle c, x \rangle \\ \text{s.t. } Ax &\le b \\ x &\ge 0. \end{aligned}$$

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.

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

Graph Matching



Positive LPs

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

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

Dominating Set

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

Positive LPs

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

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

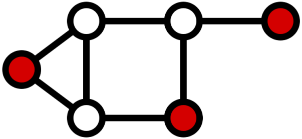
Dominating Set

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

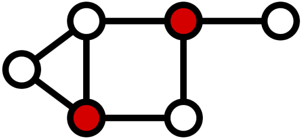
$$\begin{aligned} \min \sum_{v \in V} x_v \text{ s.t.} & \\ x_v + \sum_{u \in N(v)} x_u & \geq 1 \ \forall v \in V \\ x & \geq \mathbb{0}. \end{aligned}$$

Dominating Set

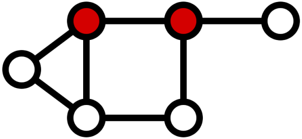
(a)



(b)



(c)



Positive LPs

Positive LPs

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

$$
\begin{aligned}
& \exists x \\
\text{s.t. } & Px \quad b \\
& Cx \quad c \\
& x \quad 0.
\end{aligned}
$$

Solving a Positive Linear System of Equations

Find x s.t. $Ax = b$

Positive LPs

Positive LPs

Let $P; C \; 2 \; \mathbb{R}_+^{m \quad n}$, $b; c \; 2 \; \mathbb{R}_+^m$. **Mixed Packing Covering LP** is

$$
\begin{aligned}
9x & \\
\text{s.t. } Px & \quad b \\
Cx & \quad c \\
x & \quad 0:
\end{aligned}
$$

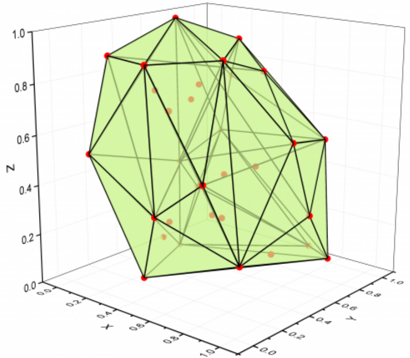
Solving a Positive Linear System of Equations

Find x s.t. $Ax = b$

$$
\begin{aligned}
9x \text{ s.t. } Ax & \quad b \\
Ax & \quad b \\
x & \quad 0:
\end{aligned}
$$

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

$$\max_x c^T x + {}_1 \log \sum_i (b_i \quad a_i^T x) + {}_2 \log \sum_j x_j$$

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

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

Approximation

A positive LP produces an ϵ-approximation answer if

$$\mathbb{1}^T x \geq \text{OPT and } Ax \leq (1 + \epsilon)\mathbb{1}$$

Methods for Solving Positive LPs

Approximate (Normal) Packing LP

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

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

How to solve quickly, accurately, and in parallel?

- Simplex algorithm: Exact but exponential

Methods for Solving Positive LPs

Approximate (Normal) Packing LP

Let A 2 $\mathbb{R}_+^{m \quad n}$ **Normal Packing LP** is

$$\text{Find } x \text{ s.t. } h\mathbb{1}; x i \quad \text{OPT and } Ax \quad (1 + \ )\mathbb{1}; x \quad 0:$$

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

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

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

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

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 $O(\text{polylog}(n) N / \epsilon)$ time where $N = \text{nnz}(A)$

Methods for Solving Positive LPs

Approximate (Normal) Packing LP

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

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

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 $O(\text{polylog}(n) N/\epsilon)$ time where $N = \text{nnz}(A)$
  2. in $O(\log(m)\log(n/\epsilon)/\epsilon^2)$ iterations in parallel

MWU Framework

Recall $A\ 2\ \mathbb{R}_+^{m \quad n}$

Optimization Problem (Packing LP)

Solve $\max h \mathbb{1}; x i$ s.t. $Ax \quad \mathbb{1}$, $x \quad \mathbb{0}$

Find x such that $\mathbb{1}^\mathsf{T} x \quad \mathrm{OPT}; Ax \quad (1 + \ )\mathbb{1}$

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

Optimization Problem (Packing LP)

Solve $\max\langle \mathbb{1}, x\rangle$ s.t. $Ax \quad \mathbb{1}$, $x \quad \mathbb{0}$

Find x such that $\mathbb{1}^{\mathsf{T}} x \quad \mathrm{OPT}, Ax \quad (1 + \ )\mathbb{1}$

1. $x \quad \mathbb{0}^n, \qquad \log(m)/$

Recall A 2 $\mathbb{R}_+^{m \quad n}$

Optimization Problem (Packing LP)

Solve max$h\mathbb{1}; x\, i$ s.t. $Ax \quad \mathbb{1}$, $x \quad \mathbb{0}$

Find x such that $\mathbb{1}^{\mathsf{T}} x \quad \mathrm{OPT}; Ax \quad (1 + \ )\mathbb{1}$

1. $x \quad \mathbb{0}^n$, $\quad$ log(m)=
2. Initialize weights equally $w \quad \frac{1}{m}\mathbb{1}^m$

MWU Framework

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

Optimization Problem (Packing LP)

Solve $\max\langle \mathbb{1}, x \rangle$ s.t. $Ax \quad \mathbb{1}$, $x \quad \mathbb{0}$

Find x such that $\mathbb{1}^\top x \quad \mathrm{OPT}, Ax \quad (1 + \quad)\mathbb{1}$

1. $x \quad \mathbb{0}^n$, $\quad \log(m)/$
2. Initialize weights equally $w \quad \frac{1}{m}\mathbb{1}^m$
3. Solve Lagrangian relaxation, $\max\langle \mathbb{1}, d \rangle$ s.t. $w^\top A d \quad w^\top \mathbb{1}$, $d \quad \mathbb{0}$

MWU Framework

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

Optimization Problem (Packing LP)

Solve $\max\langle \mathbb{1}, x \rangle$ s.t. $Ax \quad \mathbb{1}$, $x \quad \mathbb{0}$

Find x such that $\mathbb{1}^\top x \quad \mathrm{OPT}, Ax \quad (1 + \ )\mathbb{1}$

1. $x \quad \mathbb{0}^n$, $\quad \log(m)/$
2. Initialize weights equally $w \quad \frac{1}{m}\mathbb{1}^m$
3. Solve Lagrangian relaxation, $\max\langle \mathbb{1}, d \rangle$ s.t. $w^\top Ad \quad w^\top \mathbb{1}$, $d \quad \mathbb{0}$
4. Increment $x \quad x + \quad d$ for small

MWU Framework

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

Optimization Problem (Packing LP)

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

Find x such that $\mathbb{1}^\top x \geq \mathrm{OPT}, Ax \leq (1+\epsilon)\mathbb{1}$

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. $w^\top A d \leq w^\top \mathbb{1}$, $d \geq \mathbb{0}$
4. Increment $x \leftarrow x + \eta d$ for small η
5. Update weights, $w_i = w_i \cdot \exp(\eta A_i d)\ \forall i$

MWU Framework

Recall $A \; 2 \; \mathbb{R}_+^{m \quad n}$

Optimization Problem (Packing LP)

Solve $\max h\mathbb{1}; x\, i$ s.t. $Ax \quad \mathbb{1}$, $x \quad \mathbb{0}$

Find x such that $\mathbb{1}^\mathsf{T} x \quad \mathrm{OPT}; Ax \quad (1 + \;)\mathbb{1}$

1. $x \quad \mathbb{0}^n$, $\quad \log(m) =$
2. Initialize weights equally $w \quad \frac{1}{m}\mathbb{1}^m$
3. Solve Lagrangian relaxation, $\max h\mathbb{1}; d\, i$ s.t. $w^\mathsf{T} A d \quad w^\mathsf{T}\mathbb{1}$, $d \quad \mathbb{0}$
4. Increment $x \quad x + \quad d$ for small
5. Update weights, $w_i = w_i \;\; \exp(\quad A_i d) \; 8i$
6. If constraints are not tight, go to step (3)

Solving Lagrangian Relaxation

Lagrangian relaxation

$\max\langle \mathbb{1}, d \rangle$ s.t. $w^\top A d \quad w^\top \mathbb{1}$, $d \quad \mathbb{0}$.

Solving Lagrangian Relaxation

Lagrangian relaxation

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

Knapsack

Equivalent to

$$\max \langle \mathbb{1}, d \rangle \text{ s.t. } \Big\langle \underbrace{\frac{A^\top w}{\langle \mathbb{1}, w \rangle}}_{g}, d \Big\rangle \le 1; \text{ or}$$

$$\begin{aligned} &\max d_1 + \dots + d_n \\ &\text{s.t. } g_1 d_1 + \dots + g_n d_n \le 1. \end{aligned}$$

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

MWU Framework

Optimization Problem (Packing LP)

Solve max$hx; \mathbb{1}i$ s.t. $Ax \quad \mathbb{1}$, $d \quad \mathbb{0}$

Find x such that $\mathbb{1}^{\mathsf{T}}x \quad$ OPT$; Ax \quad (1 + \quad)\mathbb{1}$

1. $x \quad \mathbb{0}^n, \qquad \log(m)=$
2. Initialize weights equally $w \quad \frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \text{argmin}_j\ g_j$ (Lagrangian relaxation)
4. Increment $x \quad x + \quad d$ for small
5. Update weights, $w_i = w_i \ \exp(\quad A_i d)\ 8i$
6. If constraints are not tight, go to step (3)

MWU Framework

Optimization Problem (Packing LP)

Solve max$hx; \mathbb{1} i$ s.t. $Ax \quad \mathbb{1}$, $d \quad \mathbb{0}$

Find x such that $\mathbb{1}^T x \quad \mathrm{OPT}; Ax \quad (1 + \quad)\mathbb{1}$

1. $x \quad \mathbb{0}^n, \qquad \log(m) =$
2. Initialize weights equally $w \quad \frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \mathrm{argmin}_j \; g_j$ (Lagrangian relaxation)
4. Increment $x \quad x + \quad d$ s.t. $\max_i \quad A_i d \quad =$
5. Update weights, $w_i = w_i \;\; \exp(\quad A_i d) \; 8i$
6. If constraints are not tight, go to step (3)

MWU Framework

Find x such that $\mathbb{1}^T x \quad \text{OPT}; Ax \quad (1 + \quad)\mathbb{1}$

1. $x \quad \mathbb{0}^n, \qquad \log(m)=$
2. Initialize weights equally $w \quad \frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \text{argmin}_j\, g_j$ (Lagrangian relaxation)
4. Increment $x \quad x + \quad d$ s.t. $\max_i \quad A_i d \quad =$
5. Update weights, $w_i = w_i \quad \exp(\quad A_i d)\ 8i$
6. If constraints are not tight, go to step (3)

Find x such that $\mathbb{1}^T x$ OPT; Ax $(1 + \;)\mathbb{1}$

1. x $\mathbb{0}^n$, $\log(m)=$
2. Initialize weights equally w $\frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \text{argmin}_j \; g_j$ (Lagrangian relaxation)
4. Increment x $x +$ d s.t. $\max_i$ $A_i d$ $=$
5. Update weights, $w_i = w_i$ $\exp($ $A_i d)$ $8i$
6. If constraints are not tight, go to step (3)

Lemma

Step (3) is called at most $O(m \quad = \;) = O(m \log(m)= \;^2)$ *times.*

MWU Example

Consider LP $\max x_1 + x_2$

$$\begin{matrix} 2 & 1 \\ 1 & 3 \end{matrix} \quad \begin{matrix} x_1 \\ x_2 \end{matrix} \qquad \begin{matrix} 1 \\ 1 \end{matrix} \; :$$

where $x \;\; = \;\; 2{=}5 \quad 1{=}5 \;^{\mathsf{T}}$. Choose $\;\; = 0{:}1$.

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

MWU Example

Consider LP $\max x_1 + x_2$

$$\begin{matrix} 2 & 1 \\ 1 & 3 \end{matrix} \quad \begin{matrix} x_1 \\ x_2 \end{matrix} \qquad \begin{matrix} 1 \\ 1 \end{matrix} \; .$$

where $x \;\; = \;\; 2{=}5 \quad 1{=}5 \;^{\mathsf{T}}$. Choose $\;\; = 0{:}1$.

- (1,2): Initialize $x \quad \mathbb{0}$, $w \quad \frac{1}{2}\mathbb{1}$, $\quad 10$
- (3): Solve the Lagrangian, argmin $\frac{w^{\mathsf{T}} A}{\mathbb{1}^{\mathsf{T}} w} = \;\; 1{:}5 \quad 2 \;$. Set $d = e_1$

Consider LP $\max x_1 + x_2$

$$\begin{matrix} 2 & 1 \\ 1 & 3 \end{matrix} \quad \begin{matrix} x_1 \\ x_2 \end{matrix} \qquad \begin{matrix} 1 \\ 1 \end{matrix} \; :$$

where $x \;\; = \;\; 2{=}5 \quad 1{=}5 \;^{\mathsf{T}}$. Choose $\;\; = 0{:}1$.

- (1,2): Initialize $x \qquad \mathbb{0}$, $w \qquad \frac{1}{2}\mathbb{1}$, $\qquad\quad 10$
- (3): Solve the Lagrangian, argmin $\frac{w^{\mathsf{T}}A}{\mathbb{1}^{\mathsf{T}}w} = \;\; 1{:}5 \quad 2 \;$. Set $d = e_1$
- (4): Increment $x \qquad x + \frac{1}{4}d = \;\; \frac{1}{4} \quad 0 \;^{\mathsf{T}}$

MWU Example

Consider LP $\max x_1 + x_2$

$$\begin{matrix} 2 & 1 \\ 1 & 3 \end{matrix} \quad \begin{matrix} x_1 \\ x_2 \end{matrix} \qquad \begin{matrix} 1 \\ 1 \end{matrix} \; :$$

where $x \;\; = \;\; 2{=}5 \quad 1{=}5^{\;\mathsf{T}}$. Choose $\;\; = 0{:}1$.

- (1,2): Initialize $x \quad \mathbb{0}$, $w \quad \frac{1}{2}\mathbb{1}$, $\qquad 10$
- (3): Solve the Lagrangian, argmin $\frac{w^{\mathsf{T}} A}{\mathbb{1}^{\mathsf{T}} w} = \;\; 1{:}5 \quad 2$. Set $d = e_1$
- (4): Increment $x \quad x + \frac{1}{4} d = \;\; \frac{1}{4} \quad 0^{\;\mathsf{T}}$
- (5): Update weights, $\begin{matrix} w_1 \\ w_2 \end{matrix} = \begin{matrix} w_1 \\ w_2 \end{matrix} \quad \exp \quad Ad \qquad \begin{matrix} w_1 \\ w_2 \end{matrix} \qquad \begin{matrix} 9 \\ 1 \end{matrix}$

MWU Example

Consider LP $\max x_1 + x_2$

$$\begin{matrix} 2 & 1 \\ 1 & 3 \end{matrix} \quad \begin{matrix} x_1 \\ x_2 \end{matrix} \qquad \begin{matrix} 1 \\ 1 \end{matrix} \; .$$

where $x \;\; = \;\; 2{=}5 \quad 1{=}5 \;^{\mathsf{T}}$. Choose $\;\; = 0{:}1$.

- (1,2): Initialize $x \quad \mathbb{0}$, $w \quad \frac{1}{2}\mathbb{1}$, $\qquad 10$
- (3): Solve the Lagrangian, argmin $\frac{w^{\mathsf{T}}A}{\mathbb{1}^{\mathsf{T}}w} = \;\; 1{:}5 \quad 2 \;$. Set $d = e_1$
- (4): Increment $x \quad x + \frac{1}{4}d = \;\; \frac{1}{4} \quad 0 \;^{\mathsf{T}}$
- (5): Update weights, $\begin{matrix} w_1 \\ w_2 \end{matrix} = \begin{matrix} w_1 \\ w_2 \end{matrix} \quad \exp \quad Ad \qquad \begin{matrix} w_1 \\ w_2 \end{matrix} \quad \begin{matrix} 9 \\ 1 \end{matrix}$
- (3): Solve the Lagrangian, argmin $\frac{w^{\mathsf{T}}A}{\mathbb{1}^{\mathsf{T}}w} = \;\; \frac{19}{10} \quad \frac{12}{10} \;$. Set $d = e_2$

MWU Example

Consider LP $\max x_1 + x_2$

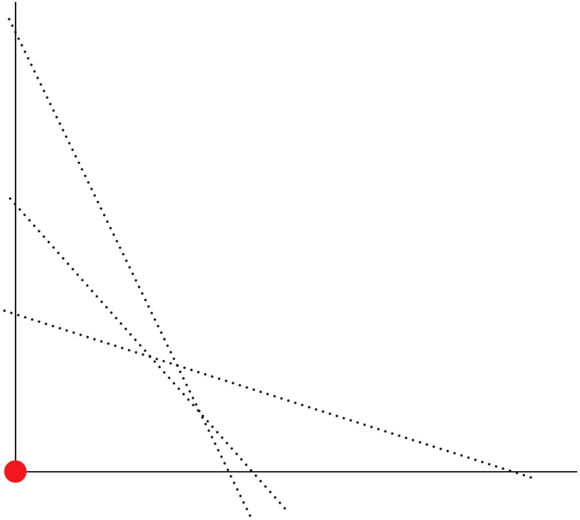
$$\begin{matrix} 2 & 1 \\ 1 & 3 \end{matrix} \quad \begin{matrix} x_1 \\ x_2 \end{matrix} \qquad \begin{matrix} 1 \\ 1 \end{matrix} \; .$$

where $x \;\; = \;\; 2{=}5 \quad 1{=}5 \;\; ^{\mathsf{T}}$. Choose $\;\; = 0{:}1$.

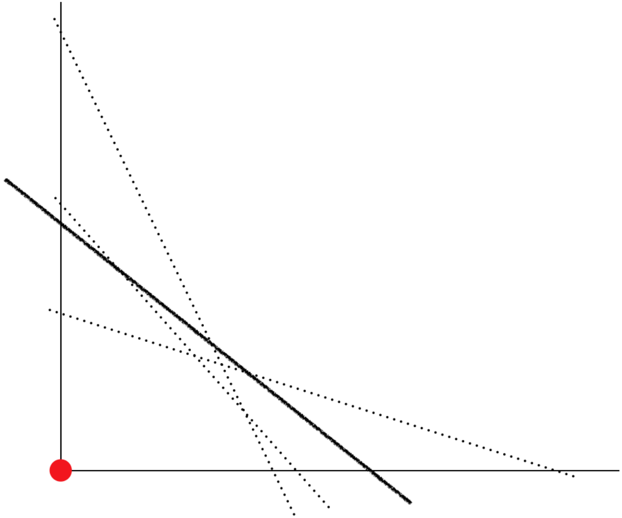
- (1,2): Initialize $x \quad\; \mathbb{0}$, $w \quad \frac{1}{2}\mathbb{1}$, $\qquad 10$
- (3): Solve the Lagrangian, argmin $\frac{w^{\mathsf{T}}A}{\mathbb{1}^{\mathsf{T}}w} = \;\; 1{:}5 \quad 2 \;$. Set $d = e_1$
- (4): Increment $x \quad\; x + \frac{1}{4}d = \;\; \frac{1}{4} \quad 0 \;\; ^{\mathsf{T}}$
- (5): Update weights, $\begin{matrix} w_1 \\ w_2 \end{matrix} \; = \; \begin{matrix} w_1 \\ w_2 \end{matrix} \quad \exp \quad Ad \qquad \begin{matrix} w_1 \\ w_2 \end{matrix} \quad \begin{matrix} 9 \\ 1 \end{matrix}$
- (3): Solve the Lagrangian, argmin $\frac{w^{\mathsf{T}}A}{\mathbb{1}^{\mathsf{T}}w} = \;\; \frac{19}{10} \quad \frac{12}{10} \;$. Set $d = e_2$
- $\ldots$

MWU Example

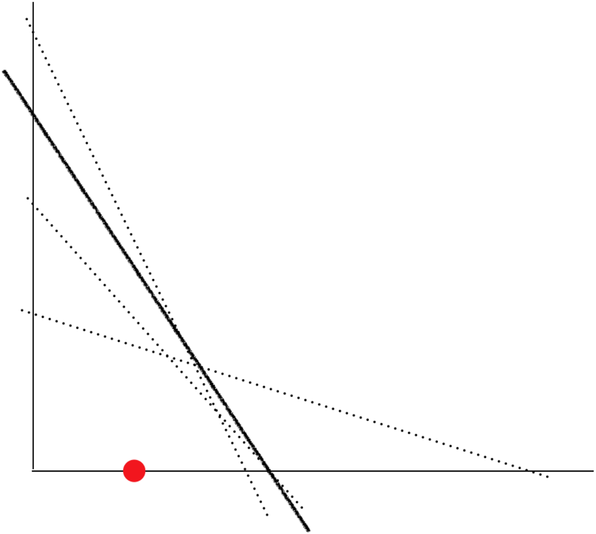
Consider standard LP with $m = 3$ constraints and $n = 2$ vars.



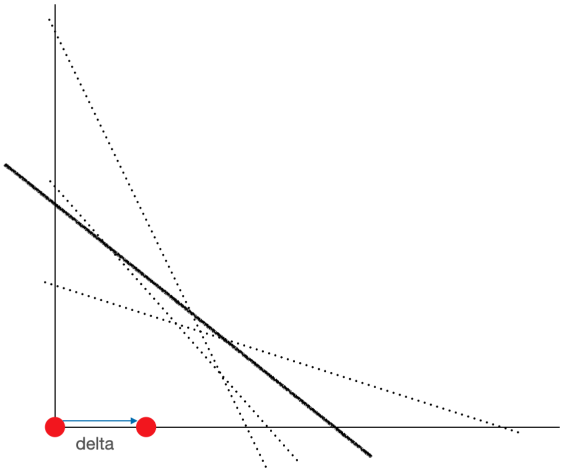
MWU Example



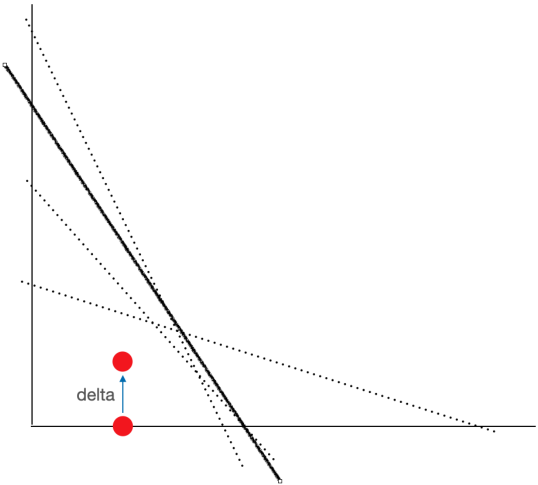
MWU Example



MWU Example



MWU Example



MWU Framework

Find x such that $x \quad \mathrm{OPT}; Ax \quad (1 + \ )\mathbb{1}$

1. $\mathbb{1}^{\mathsf{T}} x \quad \mathbb{0}^n, \quad \log(m)=$
2. Initialize weights equally $w \quad \frac{1}{m}\mathbb{1}^m$
3. Set $d = e_i$ where $i = \mathrm{argmin}_j \ g_j$ (Lagrangian relaxation)
4. Increment $x \quad x + \quad d$ s.t. $\max_i \quad A_i d \quad =$
5. Update weights, $w_i = w_i \ \exp(\quad A_i d)\ 8i$
6. If constraints are not tight, go to step (3)

Sequential ! Parallel

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

Parallelizing MWU

Lagrangian Relaxation

Equivalent to

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

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

Parallelizing MWU

Lagrangian Relaxation

Equivalent to

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

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

$$d_i = \begin{cases} f \quad g_i^{\;\;1} \quad \text{or } 1 \quad f \quad g_i & : g_i < (1 + \quad) g_{\min} \\ 0 & : g_i \quad (1 + \quad) g_{\min} \end{cases}.$$

Parallelizing MWU

Lagrangian Relaxation

Equivalent to

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

Set

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

Parallelizing MWU

Lagrangian Relaxation

Equivalent to

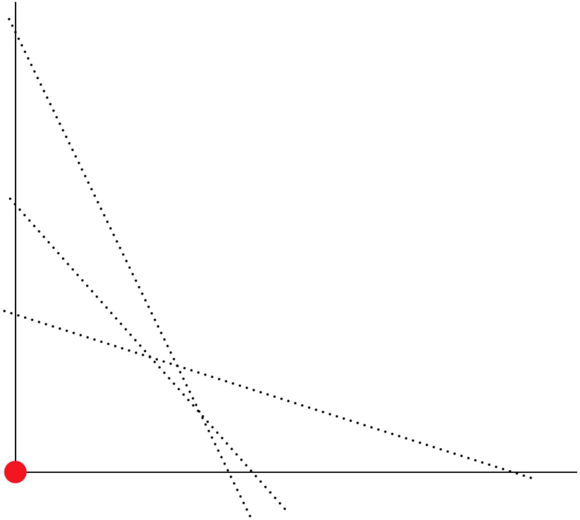
$$\max \langle \mathbb{1}, d \rangle \text{ s.t. } \left\langle \underbrace{\frac{A^{T} w}{\langle \mathbb{1}, w \rangle}}_{g}, d \right\rangle \quad 1, \quad d \quad \mathbb{0}.$$

Set

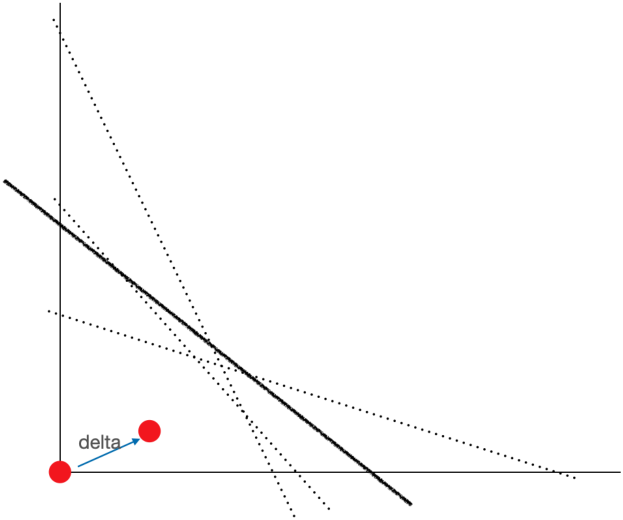
$$d_i = \begin{cases} f \;\; g_i^{\;\;1} \text{ or } 1 \quad f \;\; g_i & : g_i < (1 + \quad) g_{\min} \; \text{OPT}^{\;\;1} \\ 0 & : g_i \quad (1 + \quad) g_{\min} \; \text{OPT}^{\;\;1} \end{cases}.$$

Since $g_{\min} \quad \mathbb{1}^{T} x \quad g^{T} x \quad 1$, then $g_{\min} \quad \text{OPT}^{\;\;1}$

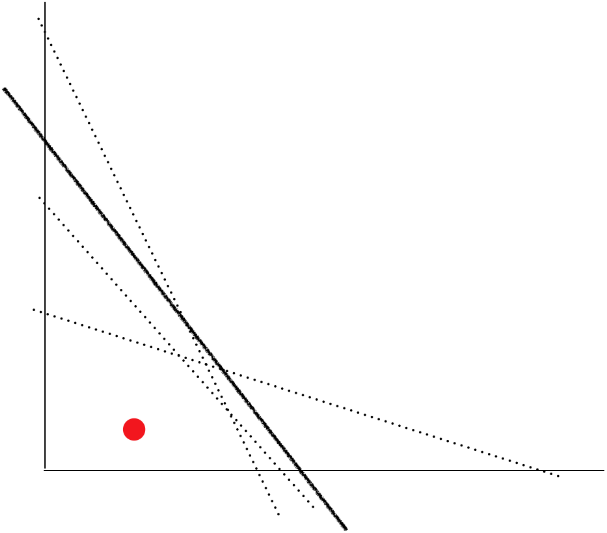
MWU Example (Parallel)



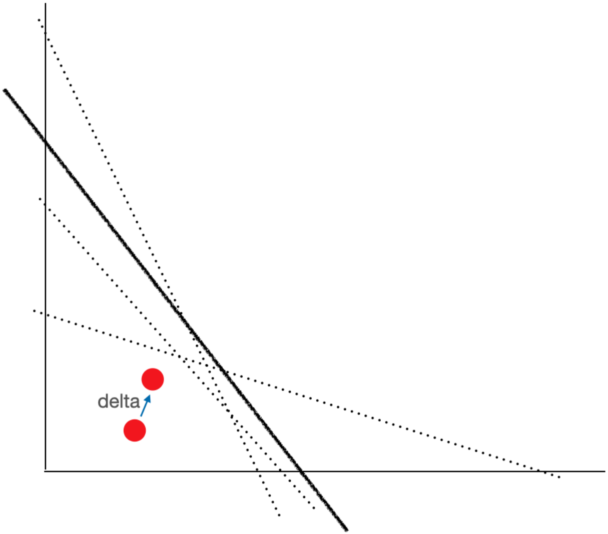
MWU Example (Parallel)



MWU Example (Parallel)



MWU Example (Parallel)



MWU Framework

1. $x > \mathbb{0}$ s.t. Ax , log(m)= , M (1)OPT
2. Initialize weights equally* $w \quad \frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max \; 0; 1 \quad g_i M g$ (Lagrangian relaxation)
4. Increment $x \quad x + \underbrace{{}^{1}x \quad \Delta}_{d}$
5. Update weights, $w_i = w_i \;\; \exp(\quad A_i d)$ $8i$
6. If constraints are not tight, go to step (3)

MWU Framework

1. $x > \mathbb{0}$ s.t. Ax , $\log(m)=$, M $(1$ $)\mathrm{OPT}$
2. Initialize weights equally* w $\frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max \; 0; 1 \quad g_i M g$ (Lagrangian relaxation)
4. Increment x $x + \underbrace{{}^{1}x \quad \Delta}_{d}$
5. Update weights, $w_i = w_i \; \exp(\quad A_i d)$ $8i$
6. If constraints are not tight, go to step (3)

*Technically, set $w = \exp(\; Ax)$

MWU Framework

1. $x > \mathbb{0}$ s.t. Ax , $\log(m)=$, M $(1$ $)\mathrm{OPT}$
2. Initialize weights equally* w $\frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max \; 0; 1 \quad g_i M g$ (Lagrangian relaxation)
4. Increment x $x + \underbrace{{}^{1}x \quad \Delta}_{d}$
5. Update weights, $w_i = w_i \; \exp(\quad A_i d) \; 8i$
6. If constraints are not tight, go to step (3)

*Technically, set $w = \exp(\; Ax)$

Lemma (Fake)

*Each iteration satis es the invariant***,

$$\frac{h\mathbb{1}; d\, i}{\max(Ax^{(new)}) \quad \max(Ax^{(old)})} \qquad M:$$

MWU Framework

1. $x > \mathbb{0}$ s.t. Ax , $\log(m)=$, M $(1$ $)\mathrm{OPT}$
2. Initialize weights equally* w $\frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max \; 0; 1 \quad g_i M g$ (Lagrangian relaxation)
4. Increment x $x + \underbrace{{}^{1}x \quad \Delta}_{d}$
5. Update weights, $w_i = w_i \;\; \exp(\quad A_i d) \; 8i$
6. If constraints are not tight, go to step (3)

*Technically, set $w = \exp(\; Ax)$

Lemma (Fake)

*Each iteration satis es the invariant**,*

$$\frac{h\mathbb{1}; d\, i}{\max(Ax^{(new)}) \quad \max(Ax^{(old)})} \quad M:$$

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

MWU Framework

1. $x_i \qquad \frac{}{n\,kA_{:,i}k}$ $8i\ 2\ [n]$, $\qquad \log(m)= \,,\ M \qquad (1 \qquad)\mathrm{OPT}$
2. Initialize weights equally $w \qquad \frac{1}{m}\mathbb{1}^m$
3. Set $\Delta_i = \max \quad 0; 1 \quad g_i M g$ (Lagrangian relaxation)
4. Increment $x \qquad x + \underbrace{{}^{1}x \quad \Delta}_{d}$
5. Update weights, $w_i = w_i \quad \exp(\quad A_i d)\ 8i$
6. If constraints are not tight, go to step (3)

Lemma

Each iteration satis es the invariant,

$$\frac{h\mathbb{1}; d\,i}{\text{smax}(Ax^{(new)}) \qquad \text{smax}(Ax^{(old)})} \qquad M:$$

Lemma

The algorithm converges in $O(\log(m)\log(n= \ )= \ ^2)$ *iterations.*

Lemma

The algorithm converges to x *s.t.* $\max(Ax) \quad \mathrm{smax}\ (Ax) \quad 1$.

Lemma

The algorithm converges to x *s.t.* $h\mathbb{1}; x\, i \quad M$.

MWU Framework

Recap

- Approximately solve special LPs using MWU up to error
- 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

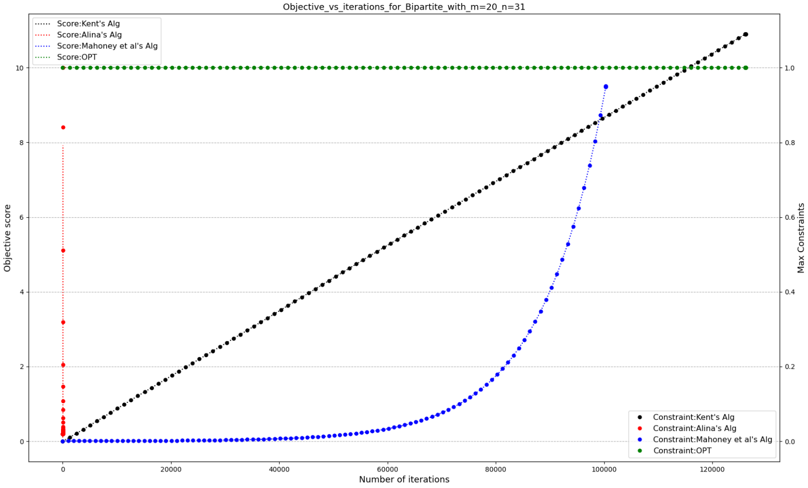
Problem Definition

Maximum Matching

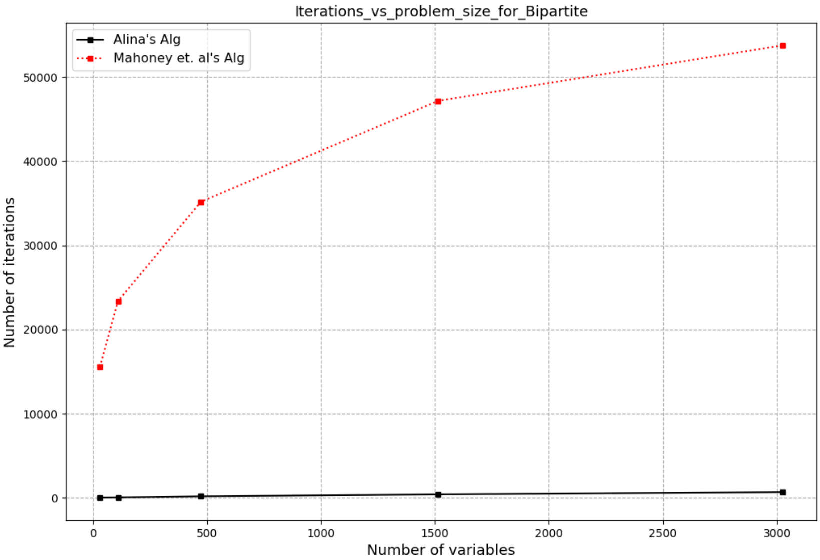
Given a graph $G = (V, E)$, find largest cardinality $F \subseteq E$ such that $\forall v \in V$ is incident to at most one edge in F.

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

Iteration Count Comparison



Iteration Count Growth Comparison

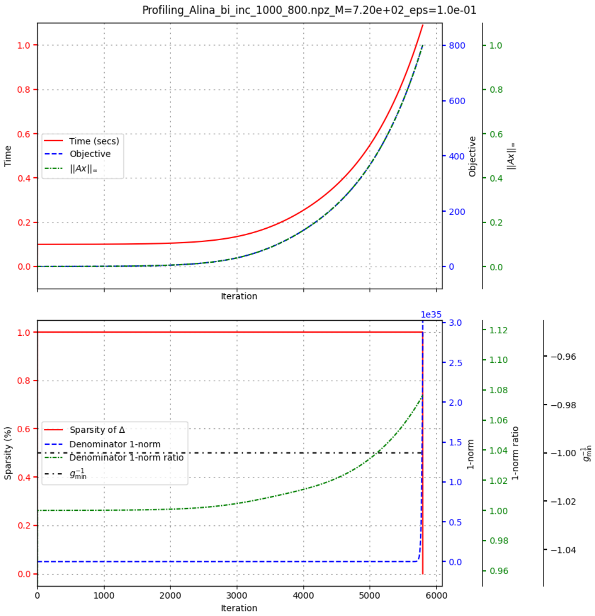


Experimental Results

Table: CPLEX vs. ParPacLP runtime and iteration count for graph matching with $= 0.1$. 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,

$$d = \quad {}^{1}(x \quad \Delta)$$

too conservative?

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

$$\frac{\langle \mathbb{1}; x + d \rangle \quad \langle \mathbb{1}; x \rangle}{\operatorname{smax} \; A(x+d) \quad \operatorname{smax} \; Ax} = \frac{\langle \mathbb{1}; d \rangle}{t} \quad M$$

Can we take larger steps?

Is the step size,

$$d = \quad {}^{1}(x \quad \Delta)$$

too conservative?

Solve line search: max   s.t.

$$\frac{\langle \mathbb{1}, d \rangle}{\text{smax} \quad A(x + \quad d) \quad \text{smax} \quad Ax} \quad M$$

$$\max \; A(x + \quad d) \quad 1.$$

Binary search over   .

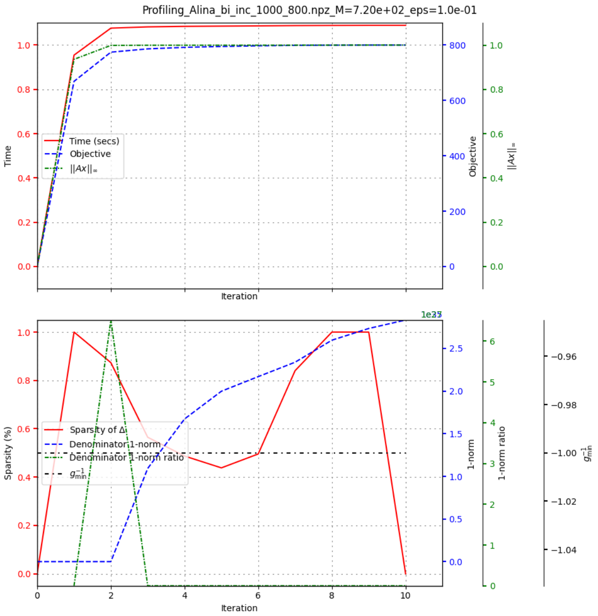
Table: CPLEX vs. `ParPacLP+BS` runtime and iteration count for graph matching with $= 0:1$. Breakdown is (`matvec`,`vecop`,`BS`)

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 ${}_1;\ {}_2$ where $M = (1 \quad {}_1)\mathrm{OPT}$ and $\qquad {}_2$
- 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 $\tilde{O}(^{-3})$ 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.