Revisiting Tardos's Framework for Linear Programming

Faster Exact Solutions using Approximate Solvers

Bento Natura

London School of Economics and Political Science

joint work with

Daniel Dadush (CWI Amsterdam)

and

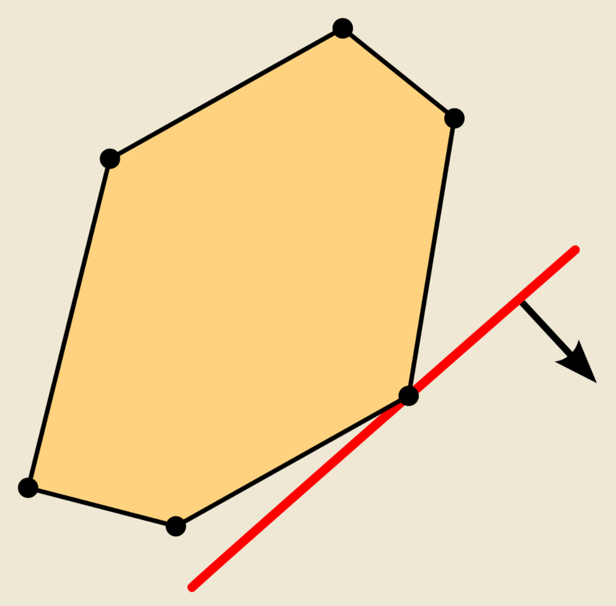
László Végh (LSE)

FOCS 2020

Linear Programming

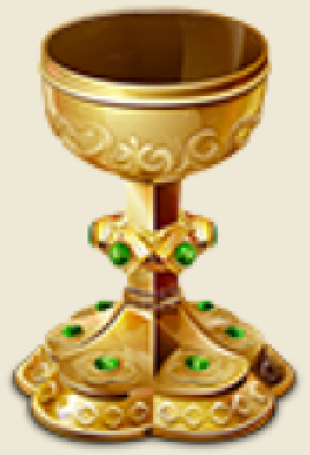
In standard form for $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n$,

$$
\begin{aligned}
\min\ &\langle c, x\rangle \\
Ax &= b \\
x &\geq 0
\end{aligned}
\qquad\qquad
\begin{aligned}
\max\ &\langle y, b\rangle \\
A^\top y + s &= c \\
s &\geq 0
\end{aligned}
$$



Timeline

?	**Strongly polynomial algorithm for LP** Smale's 9th question
1980s	**Interior point methods** Karmarkar
1970s	**Ellipsoid Method** Khachiyan
1940s	**Simplex Method** Dantzig
1820s	**Origins** Fourier



Weakly vs Strongly Polynomial Algorithms for LP

LP with n variables, m constraints
L: encoding length of the input.

weakly polynomial

- $\text{poly}(m, n, L)$ basic arithmetic operations.
- Standard variants of Ellipsoid and interior point methods: running time bound heavily relies on L.

strongly polynomial

- $\text{poly}(m, n)$ basic arithmetic operations.
- PSPACE: all numbers occurring in the algorithm must remain polynomially bounded in input size.

Fast Weakly Polynomial Algorithms for LP

ε-approximate solution

- Approximately optimal: $\langle c, x\rangle \leq \mathrm{OPT} + \varepsilon\|c\|R$
- Approximately feasible: $\|Ax - b\| \leq \varepsilon(\|A\|_F R + \|b\|)$
- $\log(1/\varepsilon)$ dependence $\Rightarrow$ exact algorithm with L dependence.

Recent progress

- Randomized $O((\mathrm{nnz}(A) + m^2)\sqrt{m}\log^{O(1)}(n)\log(n/\varepsilon))$ Lee, Sidford '13–'19
- Randomized $O(n^\omega \log^{O(1)}(n)\log(n/\varepsilon))$ Cohen, Lee, Sidford '19
- Deterministic $O(n^\omega \log^2(n)\log(n/\varepsilon))$ van den Brand '20
- Randomized $O((mn + m^3)\log^{O(1)}(n)\log(n/\varepsilon))$ van den Brand, Lee, Sidford, Song '20

Strongly Polynomial Algorithms for LP

Network flow problems

- Maximum flow: Edmonds–Karp–Dinitz '70-72
- Min-cost flow: Tardos '85

Special classes of LP

- Feasibility of 2-variable-per-inequality systems: Megiddo '83
- Discounted Markov Decision Processes: Ye '05, Ye '11
- Maximum generalized flow problem: V. '17, Olver– V. '20
- ...

Dependence on the constraint matrix only

$$\min\langle c, x\rangle \; Ax = b, \; x \geq 0$$

Running time dependent **only** on constraint matrix A, but **not** on b and c.

General LP

- 'Combinatorial LPs'
  If A integral and $\Delta_A := \max\{|\det(B)| : \; B \text{ is a square submatrix of } A\}$, then LP solvable in $\mathrm{poly}(m, n, \log \Delta_A)$ arithmetic operations: Tardos '86
- 'Layered-least-squares (LLS) Interior Point Method'
  LP solvable in $O(n^{3.5} \log \bar{\chi}_A)$ linear system solves: Vavasis–Ye '96
- 'Scaling invariant Layered-least-squares (LLS) Interior Point Method'
  LP solvable in $O(n^{2.5} \log n \log \bar{\chi}^*_A)$ linear system solves: Dadush–Huiberts–N.–Végh '20

Dependence on the constraint matrix only

$$\min\langle c, x\rangle \; Ax = b, \; x \geq 0$$

Running time dependent **only** on constraint matrix A, but **not** on b and c.

'Layered-least-squares (LLS) Interior Point Method'
LP solvable in $O(n^{3.5} \log \bar{\chi}_A)$ linear system solves: Vavasis–Ye '96

Condition number $\bar{\chi}_A$

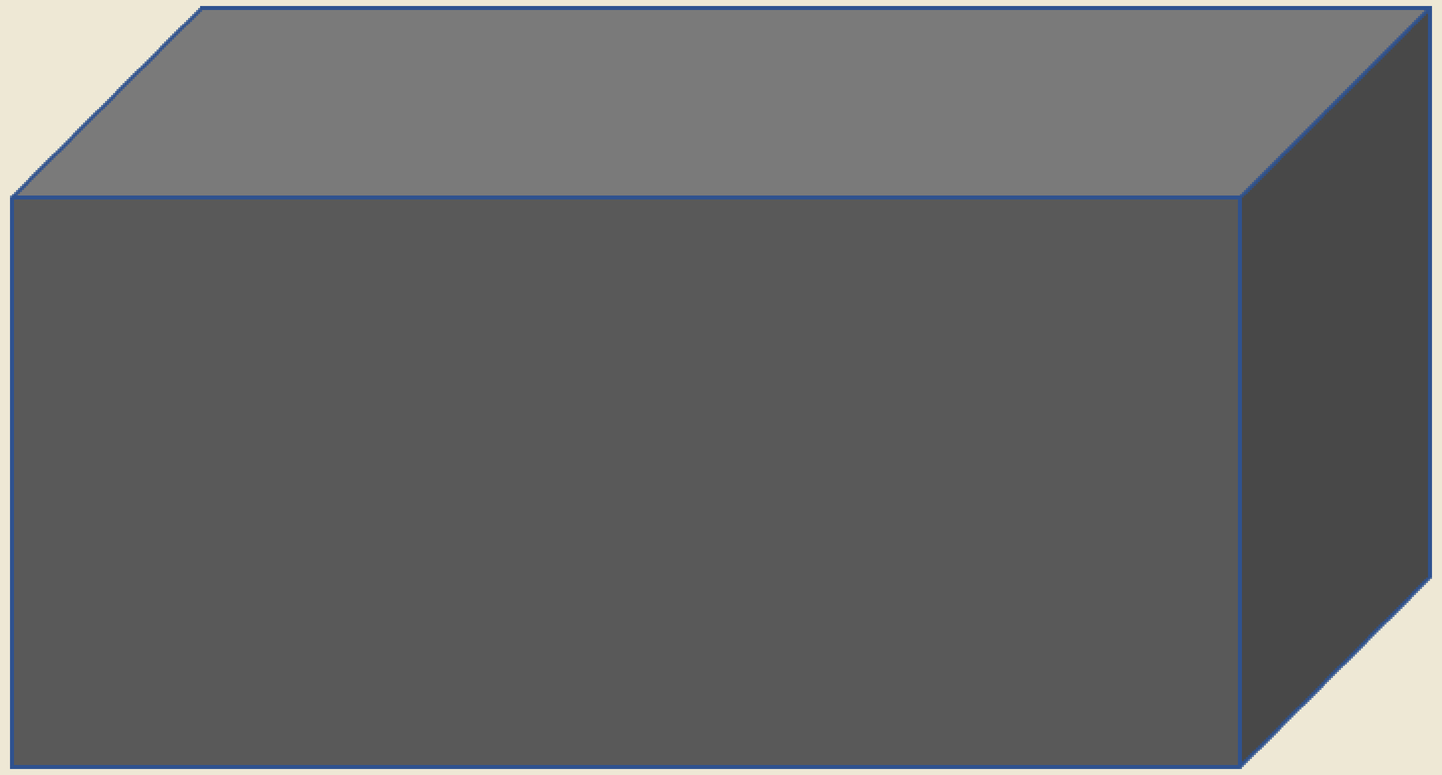
- $\bar{\chi}_A = O(2^{L_A})$.
- Depends **only** on the subspace $\ker(A)$.
- NP-hard to approximate within a factor $2^{\mathrm{poly}(\mathrm{rank}(A))}$: Tunçel '99

Can an exact LP algorithm also be fast?

Layered Least Squares IPMs

- Require computationally expensive special step directions
- Extending them to weighted central paths seems difficult

Black box approach



- Use fast approximate solver in black box manner
- Learn information about the support of the optimal solution
- Relies on proximity results on LP solutions

Tardos's framework: variable fixing

$$\min\langle c, x\rangle \; Ax = b,\; x \geq 0, A \in \mathbb{Z}^{m\times n}$$

Running time dependent **only** on constraint matrix A, but **not** on b and c.
Key idea for the first strongly polynomial algorithm for minimum cost flows.

Proximity

Use **exact** solvers to find optimal solution x to ε-rounded (perturbed) problems. Proximity yields that an optimal solution x^* to the original problem is within $\mathrm{poly}(n)\Delta_A \cdot \varepsilon$ of the rounded problem.

Variable Fixing

If the proximity is better than $\|x\|_\infty$, then we learn $x_i^* > 0$ for a variable and so the corresponding slack variable is $s_i^* = 0$.
$\rightsquigarrow$ **delete** variable and **recurse** on smaller problem.

Our contributions: Dadush–N.–Végh '20

Generalizing Tardos' result to real matrices

We give a blackbox algorithm that can handle any **real** matrix $A \in \mathbb{R}^{m\times n}$ and dependence $\log \bar{\chi}_A$ instead of $\log \Delta_A$.

Usage of approximate solvers

We only require any **approximate** LP solver, and can directly leverage the fast approximate LP algorithms. $O(mn^{\omega+1+o(1)} \log \bar{\chi}_A)$ exact deterministic LP algorithm using van den Brand '20.

Certificates for infeasibility and large condition numbers

If primal or dual linear programming are infeasible we provide a Farkas certificate. In case that the condition number is larger than our guess, we are able to provide a certificate.

$\bar{\chi}$ is hard to estimate. Iterative guesses $M \to \max\{M^2, \text{certified lower bound at failure}\}$.

The mysterious $\bar{\chi}_A$

through a matroidal lens

The condition number $\bar{\chi}_A$

Definition.

$$\bar{\chi}_A := \sup\left\{\|A^\top\left(ADA^\top\right)^{-1}AD\| \,:\, D \in \mathbf{D}\right\}$$

- Introduced by Dikin '67, Stewart '89, Todd '90, ...
- Bounds norm of oblique projections.
- Depends only on the subspace $\ker(A)$.
- Plays key role in certain interior point methods.

The circuit imbalance measure

...the "combinatorial" sister of $\bar{\chi}_A$

Definition. A **circuit** of A is a minimal linearly dependent subset of columns $C \subseteq [n]$. Let $\mathcal{C}$ denote the set of all circuits.

Definition. The **circuit imbalance measure** of A is

$$\kappa_A := \max\left\{ \left|\frac{g_j}{g_i}\right| : Ag = 0,\ \mathrm{supp}(g) \in \mathcal{C}, i, j \in \mathrm{supp}(g), \right\}$$

Lemma. *If A is a TU-matrix, then $\kappa_A = 1$. More generally, if A is integer, then $\kappa_A \leq \Delta_A$.*

Proof. For a TU-matrix, $Ax = 0, -1 \leq x \leq 1, x_J = 0$ is an integral polytope for all $J \subseteq n$. The second part follows by Cramer's rule.

Theorem. *[DHNV20]* $\sqrt{1+\kappa_A^2} \leq \bar{\chi}_A \leq n\kappa_A$. *Thus,* $\log(n+\kappa_A) = \Theta(\log(n+\bar{\chi}_A))$.

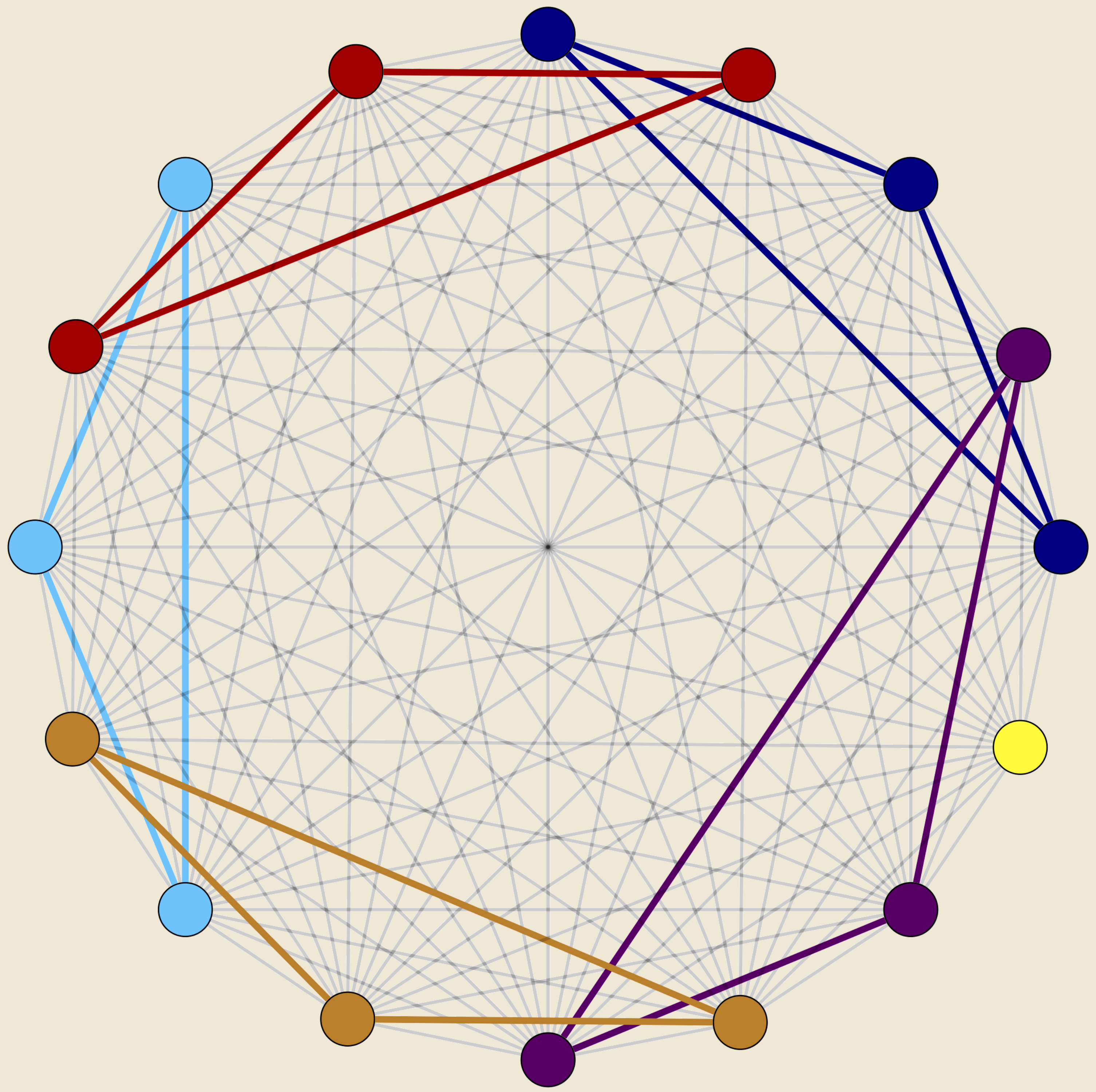
Δ vs κ

- In general $\kappa_A \leq n\Delta_A$.
- For complete undirected graph:

$$\kappa_A = 2, \text{ but } \Delta_A \geq 2^{\lfloor n/3 \rfloor}$$

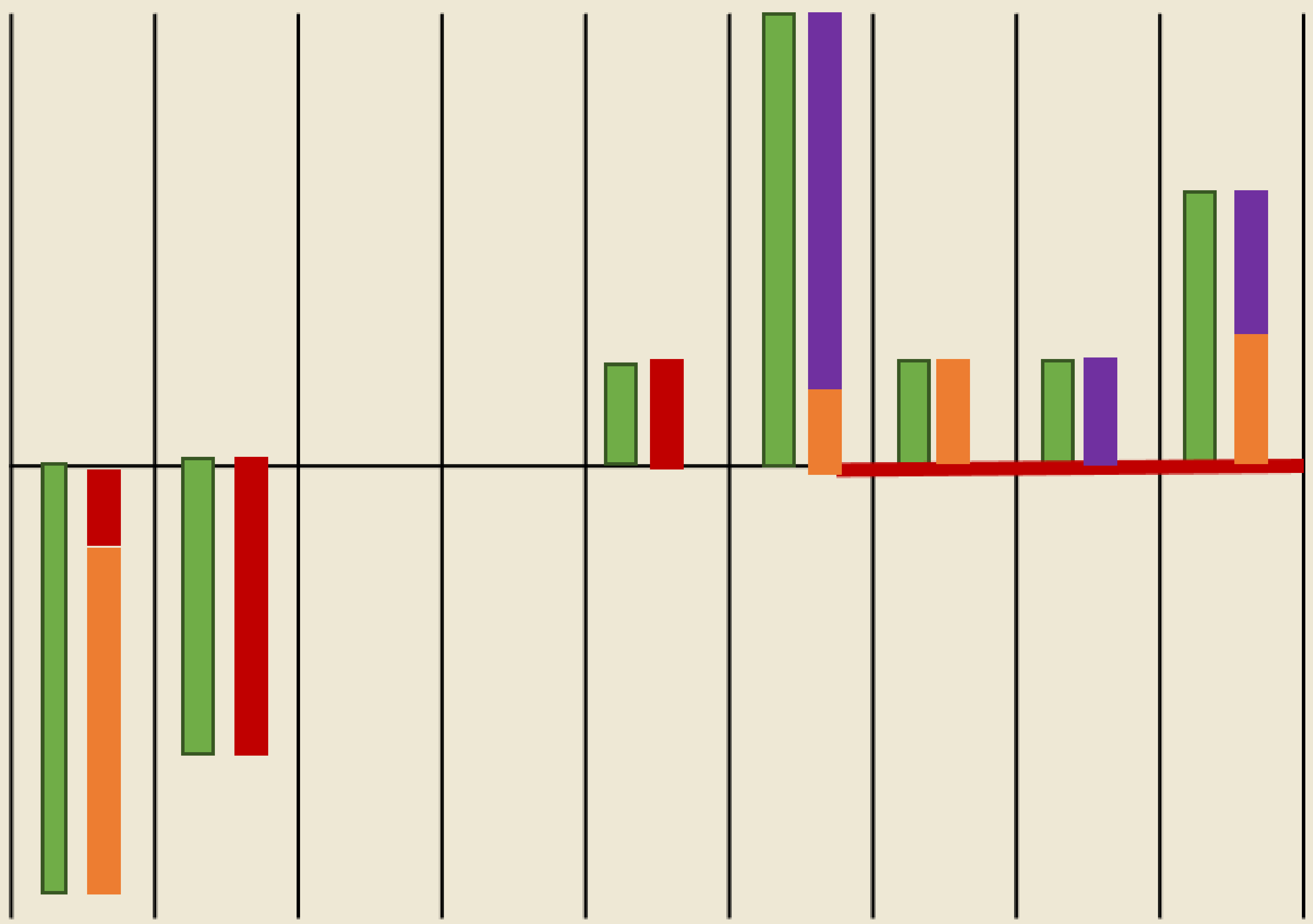
as

$$\det \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = 2$$



Near-optimal rescaling

Proximity theorems for κ_A



Linear Programming in subspace view

...a change of perspective

In standard form for $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n$,

$$\begin{aligned} \min \ \langle c, x \rangle \quad & \quad \max \ \langle y, b \rangle \\ Ax = b \quad & \quad A^\top y + s = c \\ x \geq 0 \quad & \quad s \geq 0 \end{aligned}$$

In subspace view for $W = \ker(A), d \in \mathbb{Q}^n$, s.t. $Ad = b$,

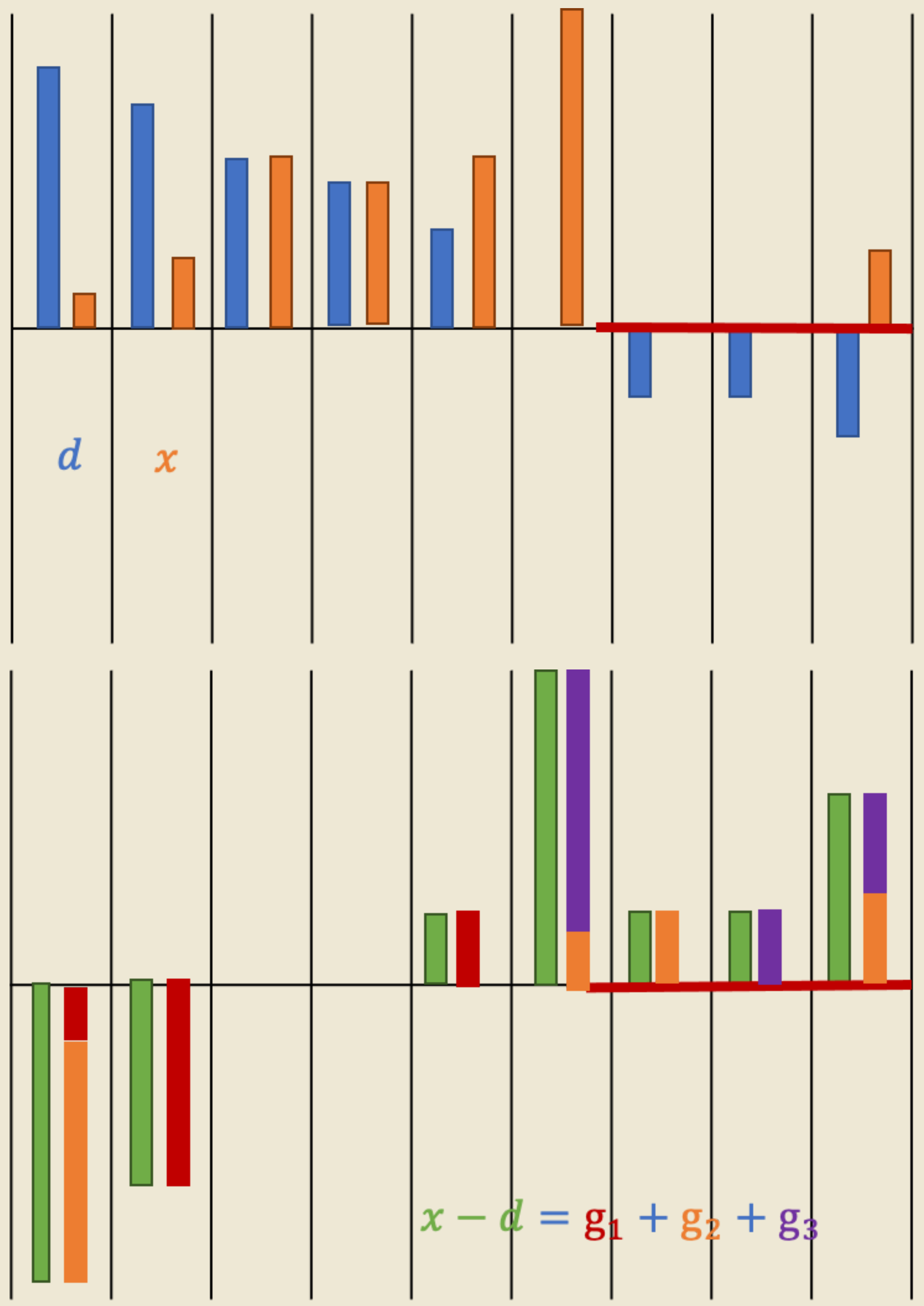
$$\begin{aligned} \min \ \langle c, x \rangle \quad & \quad \max \ \langle c - s, d \rangle \\ x \in W + d \quad & \quad s \in W^\perp + c \\ x \geq 0 \quad & \quad s \geq 0 \end{aligned}$$

Proximity theorem

Theorem. *Assume that the system $x \in W + d, x \geq 0$ is feasible. Then there exists a feasible solution such that $\|x - d\|_\infty \leq \kappa_W \|d^-\|_1$.*

Proof sketch.

- Take any feasible $x \in W + d, x \geq 0$. Thus, $x - d \in W$.
- We decompose $x - d = g_1 + g_2 + \ldots + g_t$ into sign-consistent circuits $g_i \in W$ by Carathéodory's theorem.
- Delete circuits that do not intersect $\mathrm{supp}(d^-)$.
- For all other circuits g and indices j, $|g_j| \leq \kappa_A |g_k|$ for some $k \in \mathrm{supp}(d^-)$.

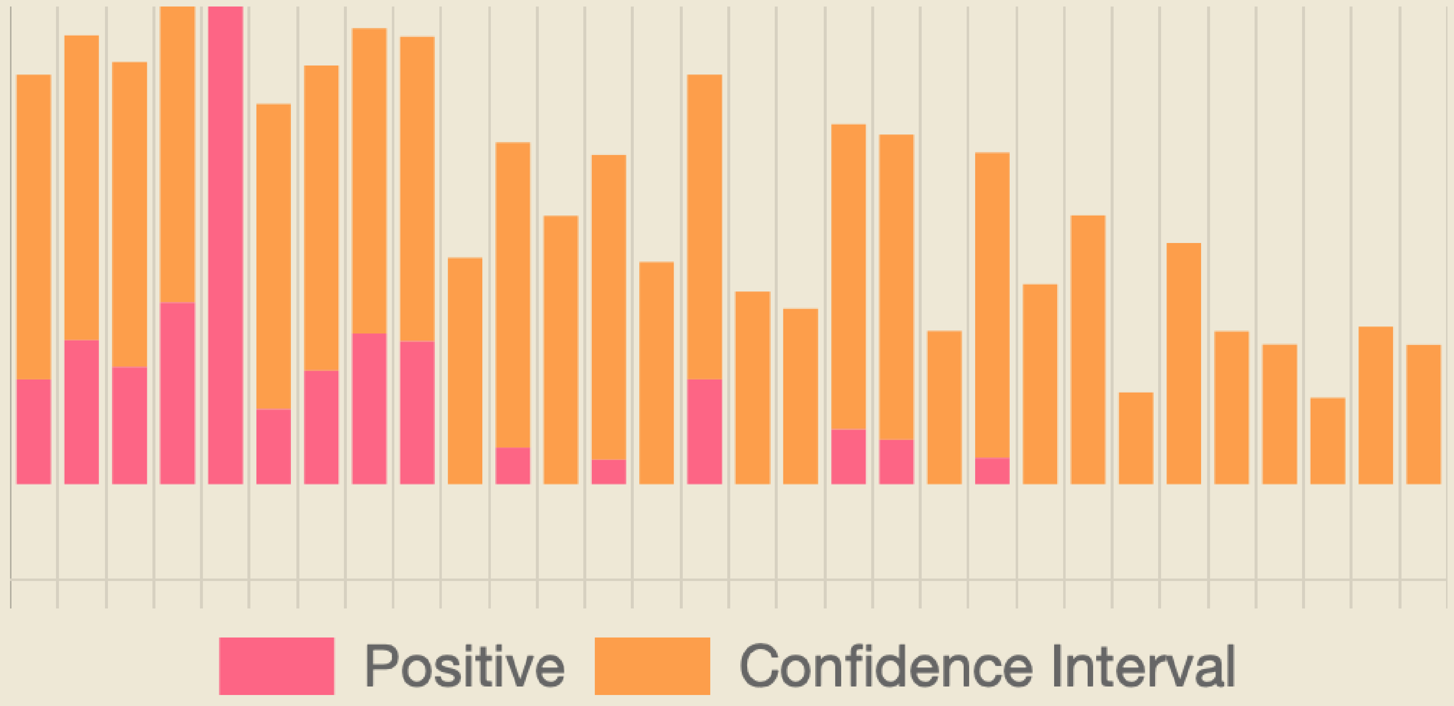


Variable fixing for feasibility

Theorem. *Assume that the system* $x \in W + d, x \geq 0$ *is feasible. Then there exists a feasible solution such that* $\|x - d\|_\infty \leq \kappa_W \|d^-\|_1$.

Recursive algorithm

- Use approximate solver to get near feasible $z \in W + d$ with $\|z^-\|_1$ "small".
- $I := \{i \in [n] : z_i > \kappa_W \|z^-\|_1\}$.
- $J := \{i \in [n] : z_i \leq \kappa_W \|z^-\|_1\}$.
- By proximity, there exists a feasible solution with $x_I > 0$.
- Recurse on the subspace $W' = \mathrm{proj}_J(W)$ with $d' = d_J$.
- If $W = \ker(A)$, then we obtain $W' = \ker(A')$ by eliminating the variables in I.



$$\left(\begin{array}{cc|c} 1 & 0 & \\ 0 & 1 & A' \\ 0 & 0 & \\ 0 & 0 & \end{array}\right)$$

Variable fixing for feasibility

Recursive algorithm

- Use approximate solver to get near feasible $z \in W + d$ with $\|z^-\|_1$ "small".
- $I := \{i \in [n] : z_i > \kappa_W \|z^-\|_1\}$.
- $J := \{i \in [n] : z_i \leq \kappa_W \|z^-\|_1\}$.
- Recurse on the subspace $W' = \text{proj}_J(W)$ with $d' = d_J$.

Questions

- How do we guarantee that $I \neq \emptyset$?
- How can we construct a feasible solution? Given $x' \in \text{proj}_J(W) + d_J, x' \geq 0$, how do we recover $x \in W + d, x \geq 0$?

The lifting operation

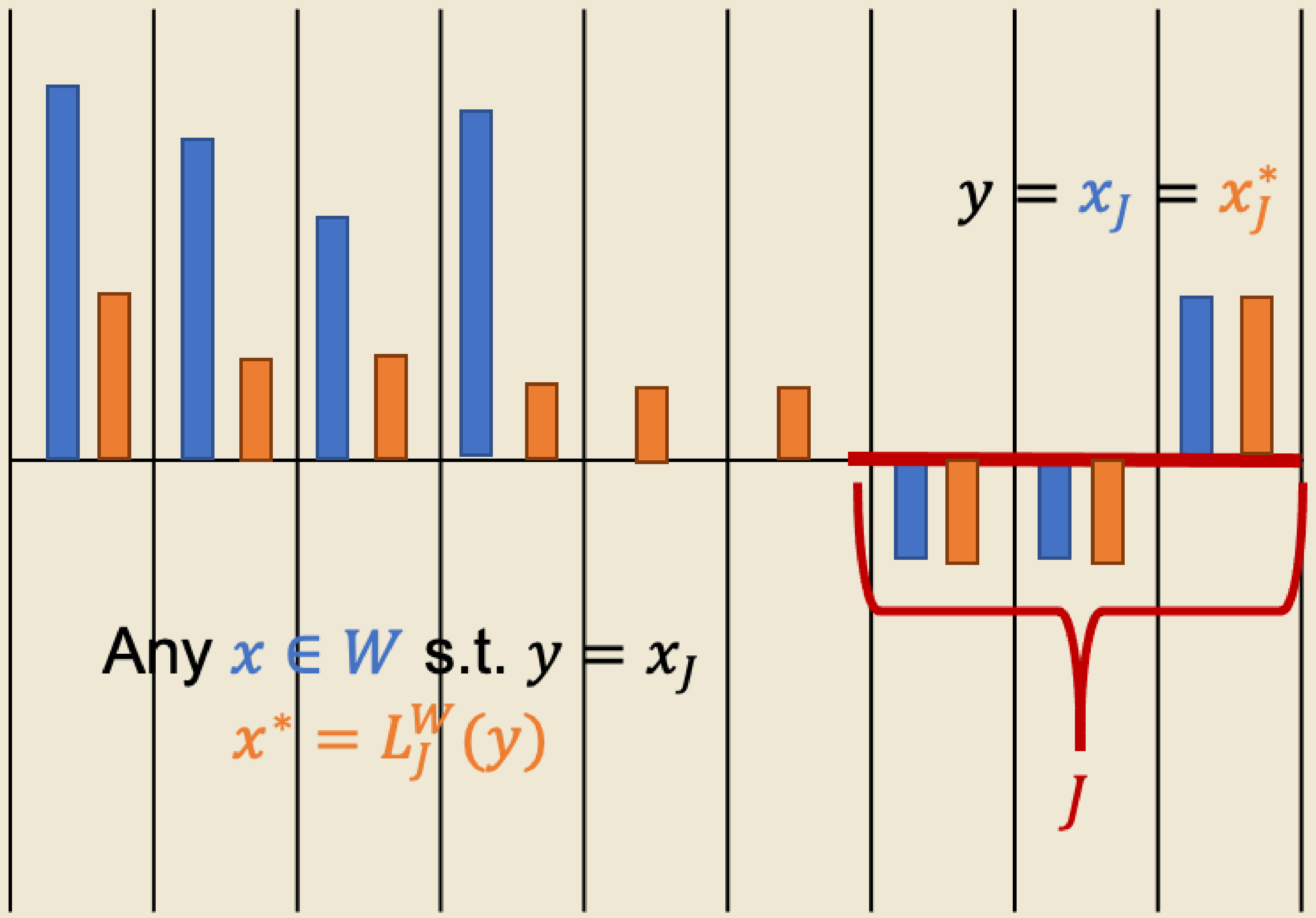
- $W \subseteq \mathbb{R}^n$ subspace, $J \subseteq [n]$
- $y \in \mathrm{proj}_J(W)$, i.e. $\exists x \in W, x_J = y$.
- The lifting of y to W is defined as

$$L_J^W(y) := \arg\min_x \{\|x\|_2 : x \in W, x_J = y\}$$

- Can be computed using a projection matrix.

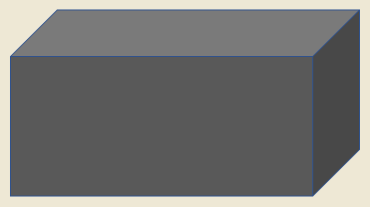
Lemma. $\|L_J^W(y)\|_\infty \leq \kappa_W \|y\|_1$.

Proof. A similar circuit decomposition argument.



The feasibility algorithm

Oracle(ε)



$$z \in W + d$$
$$\|z^-\|_\infty \le \varepsilon \|d^-\|_1$$
$$\|z - d\|_\infty \le C\kappa_W \|d^-\|_1$$

Feasibility(W, d)

$$x \in W + d, \quad x \ge 0$$
$$\|x - d\|_\infty \le C'\kappa_W^2 n \|d^-\|_1$$

Stronger system with proximity constraint useful for "pullback"

- Obtain z by applying the oracle with $\varepsilon = 1/(\kappa \cdot \mathrm{poly}(n))$
- $J := \{i \in [n] : \ z_i < \kappa_W \|z^-\|_1\}$.
- If $J = [n]$ then replace d by the projection d/W.
- Apply the recursive solver to $\mathrm{proj}_J(W)$ and z_J to obtain $\tilde{x} \in \mathrm{proj}_J(W) + z_J, \ \tilde{x} \ge 0$.
- Lift the solution back up to obtain

$$x := z + L_J^W(\tilde{x} - z_J) \ge 0.$$

- Non-negativity and proximity follows from proximity of the recursive solver!

The feasibility algorithm

- As described above, we need $\leq n$ calls to the oracle.
- Can be decreased to $\leq m$ calls (with a little more care.)
- This leads to an $O(mn^{\omega+o(1)}\log(\kappa_W + n))$ feasibility algorithm using van den Brand '20.

Estimating and certifying κ_W

- We maintain a guess M on κ_W.
- If $\|L_J^W(y)\|_\infty \leq M\|y\|_1$ for every lifting call, the algorithm succeeds.
- Otherwise, we can recover a circuit with imbalance $> M$, showing that $\kappa_W > M$.

Proximal optimal solutions

proximity works for optimization as well!

$$\begin{aligned} \min\ & \langle c, x\rangle \\ & x \in W + d \\ & x \geq 0 \end{aligned} \qquad \begin{aligned} \max\ & \langle c - s, d\rangle \\ & s \in W^\perp + c \\ & s \geq 0 \end{aligned}$$

Let $s \geq 0$, $s \in W^\perp + c$ be a feasible dual, but not necessarily optimal solution.

Theorem. *Assuming that the primal is feasible, there exists an* ***optimal*** *solution* $x \in W + d, x \geq 0$ *such that* $\|x - d\|_\infty \leq \kappa_W \big(\|d^-\|_1 + \|d_{\mathrm{supp}(s)}\|_1\big)$.

Optimization algorithm

$$\begin{array}{ll} \min \langle c, x\rangle & \max \langle c-s, d\rangle \\ \quad x \in W + d & \quad s \in W^\perp + c \\ \quad x \geq 0 & \quad s \geq 0 \end{array}$$

Theorem. *Assuming that the primal is feasible, there exists an **optimal** solution $x \in W + d, x \geq 0$ such that $\|x - d\|_\infty \leq \kappa_W(\|d^-\|_1 + \|d_{\mathrm{supp}(s)}\|_1)$.*

- Altogether nm calls to the black box solver.
- We have $\leq n$ **Outer Loops**, each comprising $\leq m$ **Inner Loops**
- Each **Outer Loop** finds $(x, s) \in (W + d) \times (W^\perp + c)$ with $\|x^-\|_1 + \|x_{\mathrm{supp}(s)}\|_1$ "small", and $s \geq 0$ feasible for the dual.
- Using proximity, we can use this to conclude $x_I > 0$ for a certain variable set $I \subseteq [n]$ and recurse.

Open questions

- Feasibility needs m calls—can we make it $\min\{m, n-m\}$ to have the same for primal and dual?
- Optimization takes mn calls—would fewer be enough?
- Can we get better for special cases, such as max flow or min-cost flow?
- Can we get faster (possibly non-deterministic) version of the constructive Hoffman algorithm?
- Can we extend the black box approach to problems with unbounded κ, such as generalized flows?
- κ - theory for more general convex programs e.g. **Convex Quadratic Programs** or **Semidefinite Programs (SDP)**
- κ - theory for **Integer Programming (IP)**