LINEAR PROGRAMMING AND CIRCUIT IMBALANCES

László Végh

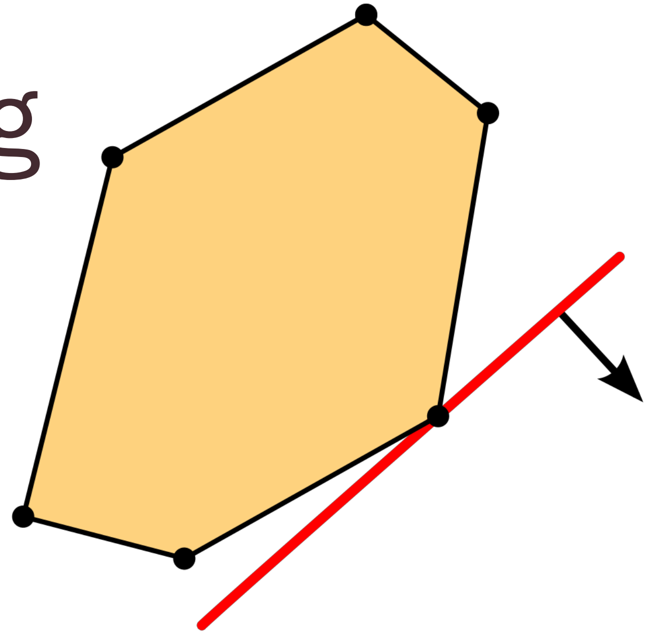
THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

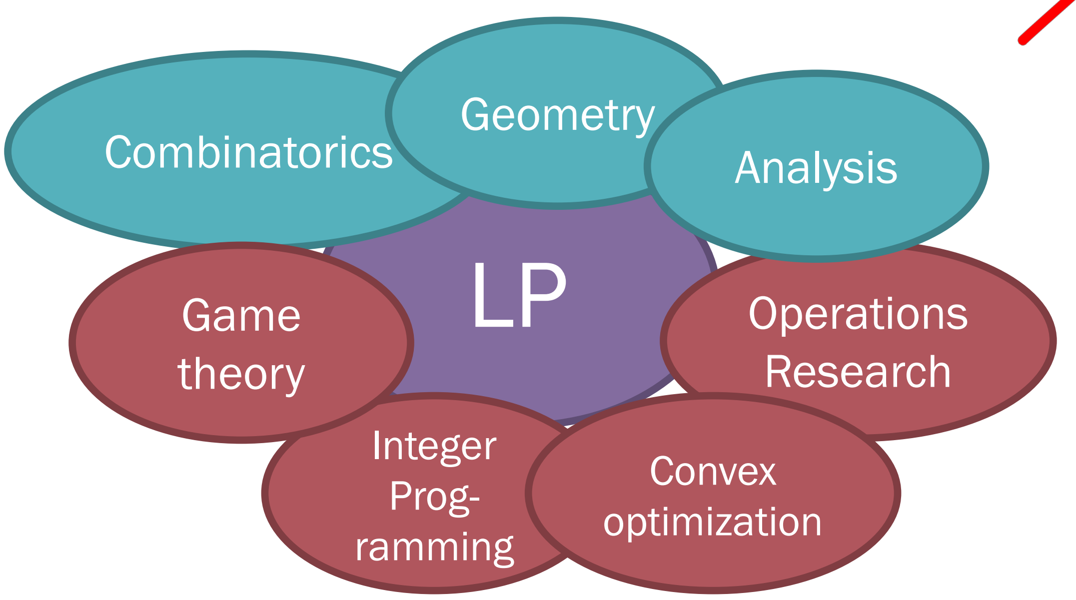
IPCO Summer School
Georgia Tech, May 2021

Slides available at
https://personal.lse.ac.uk/veghl/ipco

Linear programming

$$\min c^\top x$$
$$Ax = b$$
$$x \geq 0$$





Facets of linear programming

Discrete

- Basic solutions
- Polyhedral combinatorics
- Exact solution

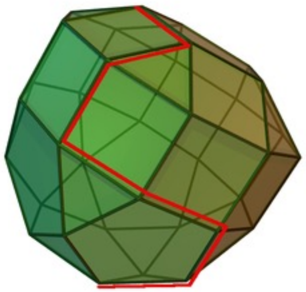
Continuous

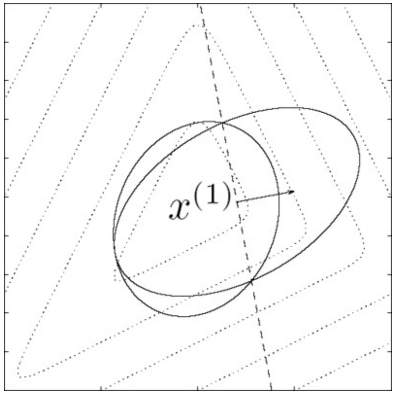
- Continuous solutions
- Convex program
- Approximate solution

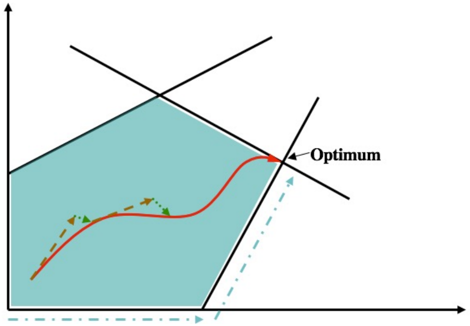
Linear programming algorithms

$$\min c^\top x$$
$$Ax = b$$
$$x \geq 0$$

- n variables, m constraints
- *L:* total bit-complexity of the rational input (A, b, c)
- Simplex method: Dantzig, 1947
- Weakly polynomial algorithms: *poly(L)* running time
  - Ellipsoid method: Khachiyan, 1979
  - Interior point method: Karmarkar, 1984







Weakly vs strongly polynomial algorithms for LP

$$\min c^\top x$$
$$Ax = b$$
$$x \geq 0$$

- n variables, m constraints, total encoding *L.*
- Strongly polynomial algorithm:
  - poly(n, m) elementary arithmetic operations $(+, -, \times, \div, \geq)$, independent of *L.*
  - PSPACE: The bit-length of numbers during the algorithm remain polynomially bounded in the size of the input.
  - Can also be defined in the real model of computation

Is there a strongly polynomial algorithm for Linear Programming?

Smale's 9th question

Strongly polynomial algorithms for some classes of Linear Programs

- Systems of linear inequalities with at most two nonzero variables per inequality: Megiddo ’83
- Network flow problems
  - Maximum flow: Edmonds-Karp-Dinitz ’70-72, …
  - Min-cost flow: Tardos ’85, Fujishige ’86, Goldberg-Tarjan ’89, Orlin ’93, …
  - Generalized flow: V ’17, Olver-V ’20
- Discounted Markov Decision Processes: Ye ’05, Ye ’11, …

Dependence on the constraint matrix only

$$\min c^\top x,\ Ax = b\ \ x \geq 0$$

- Algorithms with running time dependent only on A, but not on b and c.
- Combinatorial LP's: integer matrix $A \in \mathbb{Z}^{m \times n}$.

  $$\Delta_A = \max\{|\det(B)|: B \text{ submatrix of } A\}$$

  Tardos '86: poly$(n, m, \log \Delta_A)$ *black box* LP algorithm
- Layered-least-squares (LLS) Interior Point Method
  Vavasis-Ye '96: poly$(n, m, \log \bar{\chi}_A)$ LP algorithm in the real model of computation
  $\bar{\chi}_A$: condition number
- Dadush-Huiberts-Natura-V '20: poly$(n, m, \log \bar{\chi}_A^*)$
  $\bar{\chi}_A^*$: optimized version of $\bar{\chi}_A$

Outline of the lectures

1. Tardos's algorithm for min-cost flows
2. The circuit imbalance measure κ_A and the condition measure $\bar{\chi}_A$
3. Solving LPs: from approximate to exact
4. Optimizing circuit imbalances
5. Interior point methods: basic concepts
6. Layered-least-squares interior point methods

- Dadush-Huiberts-Natura-V '20: *A scaling-invariant algorithm for linear programming whose running time depends only on the constraint matrix*
- Dadush-Natura-V '20: *Revisiting Tardos's framework for linear programming: Faster exact solutions using approximate solvers*

Part 1

Tardos's algorithm for min-cost flows

circuits, proximity, and variable fixing

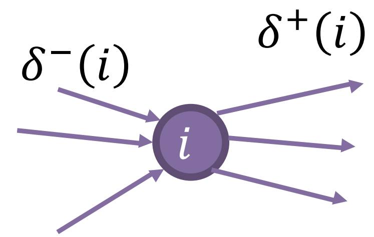
The minimum-cost flow problem

- Directed graph $G = (V, E)$, node demands $b: V \to \mathbb{R}$ with $b(V) = 0$, costs $c: E \to \mathbb{R}$.

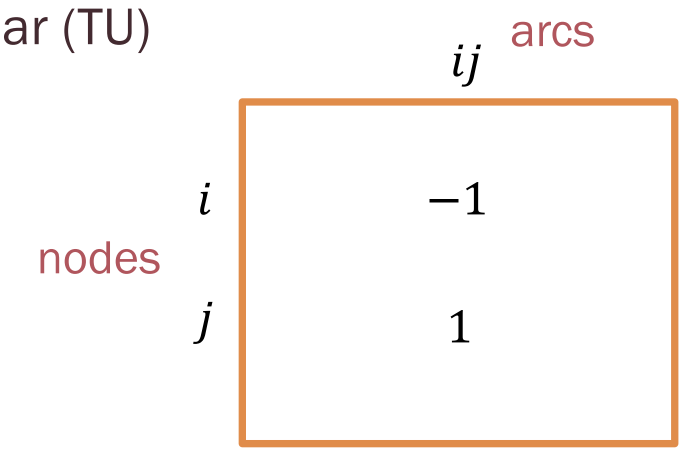
$$\min c^\top x$$

$$\text{s.t.} \sum_{ji \in \delta^-(i)} x_{ji} - \sum_{ij \in \delta^+(i)} x_{ij} = b_i \quad \forall i \in V$$

$$x \geq 0$$

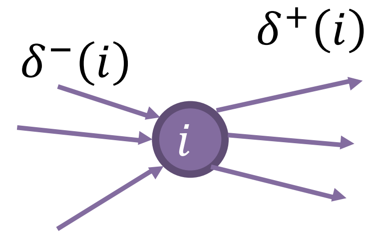


- Form with arc capacities can be reduced to this form.
- Constraint matrix is totally unimodular (TU)



The minimum-cost flow problem: optimality

- Directed graph $G = (V, E)$, node demands $b: V \to \mathbb{R}$ with $b(V) = 0$, costs $c: E \to \mathbb{R}$.



$$\min c^\top x$$
$$\text{s.t.} \sum_{(j,i) \in \delta^-(i)} x_{ji} - \sum_{(i,j) \in \delta^+(i)} x_{ij} = b_i \quad \forall i \in V$$
$$x \geq 0$$

- Dual program:

$$\max b^\top \pi$$
$$\text{s.t. } \pi_j - \pi_i \leq c_{ij} \quad \forall ij \in E$$

- Optimality: $f_{ij} > 0 \quad \Longrightarrow \quad \pi_j - \pi_i = c_{ij}$

Dual solutions: potentials

- Dual program: max cost feasible potential

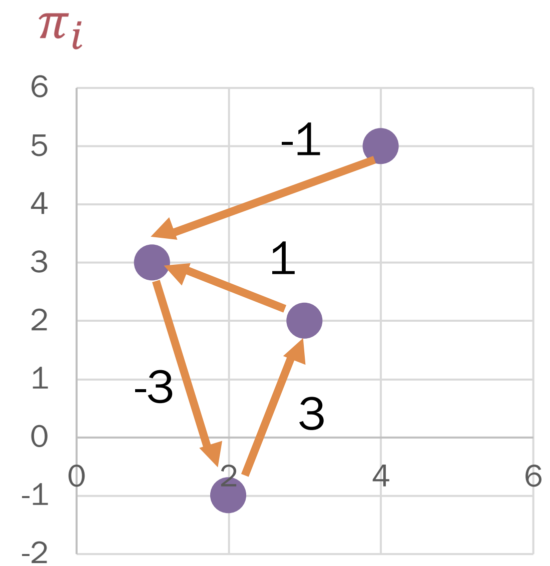
$$\max b^\top \pi$$
$$\text{s.t. } \pi_j - \pi_i \le c_{ij} \quad \forall ij \in E$$

- Residual cost:

$$c_{ij}^{\pi} = c_{ij} + \pi_i - \pi_j \ge 0$$

- Residual graph:

$$E_f = E \cup \{(j,i): f_{ij} > 0\}$$
$$c_{ji} = -c_{ij}$$



LEMMA: The primal feasible f is optimal $\Leftrightarrow$
$\exists \pi: c_{ij}^{\pi} \ge 0$ for all $(i,j) \in E$ and $c_{ij}^{\pi} = 0$ if $f_{ij} > 0$ $\Leftrightarrow$
$\exists \pi: c_{ij}^{\pi} \ge 0$ for all $(i,j) \in E_f$

Variable fixing by proximity

- If for some $(i,j) \in E$ we can show that $f^*_{ij} = 0$ in every optimal solution, then we can remove (i,j) from the graph.
- Overall goal: in strongly polynomial number of steps, guarantee that we can infer this for at least one arc.

PROXIMITY THEOREM: Let $\tilde{\pi}$ be the optimal dual potential for costs $\tilde{c}$, and f^* an optimal primal solution for the original costs c. Then,

$$c_{ij}^{\tilde{\pi}} > |V| \cdot \|c - \tilde{c}\|_\infty \quad \Rightarrow \quad f^*_{ij} = 0$$

Circulations and cycle decompositions

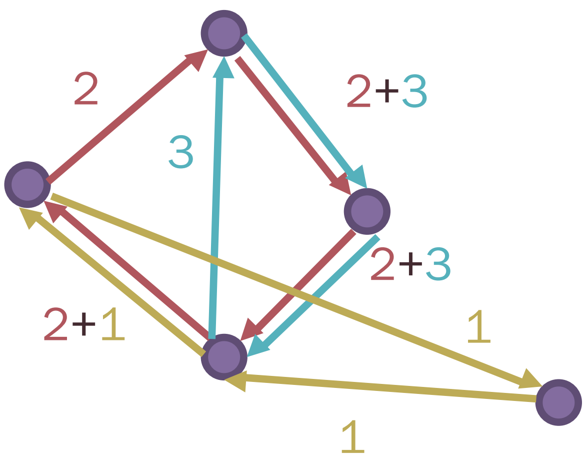
- For the node-arc incidence matrix A, $\ker(A) \subseteq \mathbb{R}^E$ is the set of circulations:

  in-flow=out-flow

- LEMMA: every circulation $f \geq 0$ can be decomposed as

$$f = \sum_i \lambda_i \chi_{C_i}, \qquad \lambda_i \geq 0$$

  for directed cycles C_i



Circulations and cycle decompositions

- **LEMMA:** Let f and f' be two feasible flows for the same demand vector b. Then, we can write

$$f' = f + \sum_i \lambda_i \chi_{C_i}, \qquad \lambda_i \geq 0$$

for sign-consistent directed cycles C_i in $\overleftrightarrow{E}$:

  - If $f'_{ij} > f_{ij}$ then cycles may only contain ij but not ji.
  - If $f_{ij} > f'_{ij}$ then cycles may only contain ji but not ij.
  - If $f_{ij} = f'_{ij}$ then no cycle contains ij or ji.

Every cycle is moving from f towards f'.

PROXIMITY THEOREM: Let $\tilde{\pi}$ be the optimal dual potential for costs $\tilde{c}$, and f^* an optimal primal solution for the original costs c. Then,

$$c_{ij}^{\tilde{\pi}} > |V| \cdot \|c - \tilde{c}\|_\infty \quad \Rightarrow \quad f_{ij}^* = 0$$

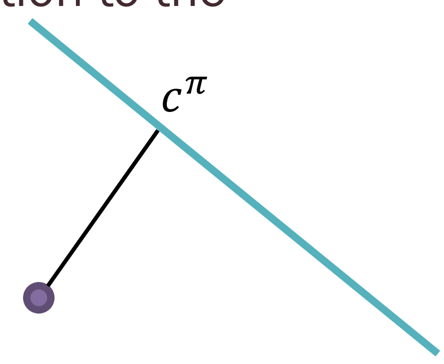
PROOF:

Rounding the costs

- Rescale c such that $||c||_\infty = |V|\sqrt{|E|}$
- Round costs as $\tilde{c}_{ij} = \lfloor c_{ij} \rfloor$
- For $\tilde{c}$ we can find optimal primal and dual solutions in strongly polynomial time, e.g. the Out-of-Kilter method by Ford and Fulkerson 1962.
- For the optimal dual $\tilde{\pi}$, fix all arcs to 0 that have
$$c_{ij}^{\tilde{\pi}} > |V| > |V| \cdot ||c - \tilde{c}||_\infty$$
- **QUESTION:** Why would such an arc exist?

Minimum-norm projections

- Residual cost:
$$c_{ij}^{\pi} = c_{ij} + \pi_i - \pi_j \geq 0$$
- The cost vectors
$$U = \{c^{\pi} : \pi \in \mathbb{R}^V\} \subset \mathbb{R}^E$$
form an affine subspace.
- For any feasible flow f and any residual cost c^{π},
$$(c^{\pi})^{\top} f = c^{\top} f + b^{\top} \pi$$
- Solving the problem for c and c^{π} is equivalent.
- If $0 \in U$, i.e. $\exists \pi : \mathrm{c}^{\pi} \equiv 0$, then every feasible flow is optimal
- IDEA: Replace the input c by the min norm projection to the affine subspace U:
$$c^{\pi} = \arg\min_{\pi \in \mathbb{R}^V} \|c^{\pi}\|_2$$



Rounding the costs

- Assume c is chosen as a min norm projection:
$$\|\boldsymbol{c}^{\boldsymbol{\pi}}\|_2 \geq \|\boldsymbol{c}\|_2 \;\; \forall \boldsymbol{\pi} \in \mathbb{R}^{\boldsymbol{V}}$$
- Rescale c such that $\|c\|_\infty = |V|\sqrt{|E|}$
- Round costs as $\tilde{c}_{ij} = \lfloor c_{ij} \rfloor$
- For the optimal dual $\tilde{\pi}$, fix all arcs to 0 that have
$$c_{ij}^{\tilde{\pi}} > |V| > |V| \cdot \|c - \tilde{c}\|_\infty$$
- LEMMA: There exist at least one such arc.
PROOF:
$$\left\|c^{\tilde{\pi}}\right\|_\infty \geq \frac{\left\|c^{\tilde{\pi}}\right\|_2}{\sqrt{|E|}} \geq \frac{\|c\|_2}{\sqrt{|E|}} \geq \frac{\|c\|_\infty}{\sqrt{|E|}} = |V|$$
Also note that
$$c_{ij}^{\tilde{\pi}} \geq \tilde{c}_{ij}^{\tilde{\pi}} \geq 0$$

Summary of Tardos's algorithm

- Variable fixing based on proximity that can be shown by cycle decomposition.
- Replace the input cost by an equivalent min-cost projection
- Round to small integer costs $\tilde{c}$
- Find optimal dual $\tilde{\pi}$ for $\tilde{c}$ with simple classical method
- Identify a variable $f_{ij}^* = 0$ as one where $c_{ij}^{\tilde{\pi}}$ is large and remove all such arcs.
- Iterate

Outline of the lectures

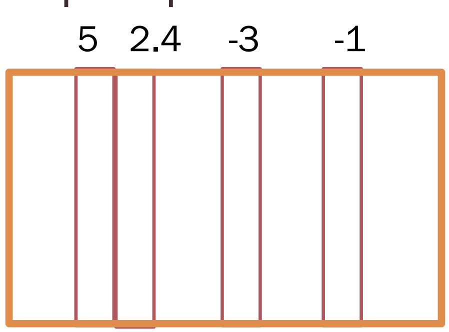
1. Tardos's algorithm for min-cost flows
2. The circuit imbalance measure κ_A and the condition measure $\bar{\chi}_A$
3. Solving LPs: from approximate to exact
4. Optimizing circuit imbalances
5. Interior point methods: basic concepts
6. Layered-least-squares interior point methods

Part 2
The circuit imbalance measure κ_A and the condition measure $\bar{\chi}_A$

The circuit imbalance measure

- The matrix $A \in \mathbb{R}^{m\times n}$ defines a linear matroid on $[n] = \{1,2,\dots,n\}$: a set $I \subseteq [n]$ is independent if the columns $\{a_i : i \in I\}$ are linearly independent.
- $C \subseteq [n]$ is a circuit if $\{a_i : i \in C\}$ is a linearly dependent set minimal for containment.
- For a circuit C, there exists a vector $g^C \in \mathbb{R}^C$ unique up to a scalar multiplier such that

$$\sum_{i\in C} g_i^C a_i = 0$$



- $\mathcal{C}_A$: set of all circuits.
- The circuit imbalance measure is defined as

$$\kappa_A = \max\left\{\frac{|g_j^C|}{|g_i^C|} : C \in \mathcal{C}_A, i,j \in C\right\}$$

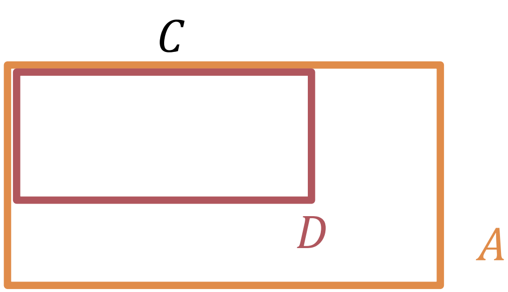
Properties of κ_A

$$\kappa_A = \max\left\{\frac{|g_j^C|}{|g_i^C|} : C \in \mathcal{C}_A, i, j \in C\right\}$$

- This measure depends only on the linear subspace $W = \ker(A)$: if $\ker(A) = \ker(B)$ then $\kappa_A = \kappa_B$
- We will use $\kappa_W = \kappa_A$ for $W = \ker(A)$

Connection to subdeterminants:

- For an integer matrix $A \in \mathbb{Z}^{m\times n}$,
$$\Delta_A = \max\{|\det(B)| : B \text{ submatrix of } A\}$$
- For a circuit $C \in \mathcal{C}_A$, with $|C| = t$ let $D = A_{J,C} \in \mathbb{R}^{(t-1)\times t}$ be a submatrix with linearly independent rows.



$D^{(i)} \in \mathbb{R}^{(t-1)\times(t-1)}$ remove the i-th column from D. By Cramer's rule

$$g^C = \left(\det\left(D^{(1)}\right), \det\left(D^{(2)}\right), \dots, \det\left(D^{(t)}\right)\right)$$

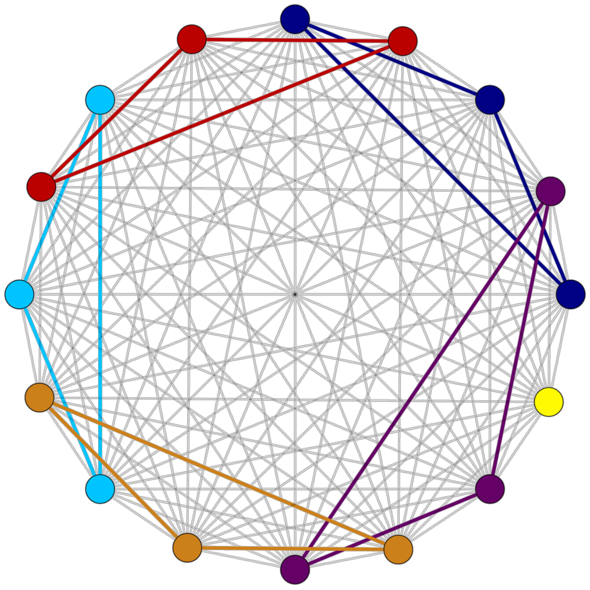
Properties of κ_A

- **LEMMA:** For an integer matrix $A \in \mathbb{Z}^{m\times n}$,
$$\kappa_A \leq \Delta_A$$
For a totally unimodular matrix A, $\kappa_A = 1$

- **EXERCISE:**

  i. If A is the node-edge incidence matrix of an undirected graph, then $\kappa_A \in \{1,2\}$

  ii. For the incidence matrix of a complete undirected graph on n nodes,
$$\Delta_A \geq 2^{\left\lfloor \frac{n}{3} \right\rfloor}$$



Circuit imbalance and TU matrices

THEOREM (Cederbaum, 1958): If $A \in \mathbb{Z}^{m\times n}$ is a TU-matrix, then $\kappa_A = 1$. Conversely, if $\kappa_W = 1$ for a linear subspace $W \subset \mathbb{R}^n$ then there exists a TU-matrix A such that $W = \ker(A)$.

PROOF (Ekbatani & Natura):

Duality of circuit imbalances

THEOREM: For every linear subspace $W \subset \mathbb{R}^n$, we have

$$\kappa_W = \kappa_{W^\perp}$$

Circuits in optimization

- Appear in various LP algorithms directly or indirectly
- IPCO summer school 2020: Laura Sanità's lectures discussed *circuit augmentation* algorithms and *circuit diameter*
- Integer programming: κ has a natural integer variant that is related to Graver bases
- …

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

$$\bar{\chi}_A = \sup\{\|A^\top (ADA^\top)^{-1} AD\| : D \text{ is positive diagonal matrix}\}$$

- Measures the norm of *oblique* projections
- Introduced by Dikin 1967, Stewart 1989, Todd 1990
- **THEOREM** (Vavasis-Ye 1996): There exists a $\text{poly}(n, m, \log \bar{\chi}_A)$ LP algorithm for $\min c^\top x$, $Ax = b, x \geq 0$, $A \in \mathbb{R}^{m\times n}$
- **LEMMA**

i. If A is an integer matrix with bit encoding length L, then $\bar{\chi}_A \leq 2^{O(L)}$

ii. $\bar{\chi}_A = \max\{\|B^{-1}A\| : B \text{ nonsingular } m\times m \text{ submatrix of } A\}$

iii. $\bar{\chi}_A$ *only depends on the subspace* $W = \ker(A)$

iv. $\bar{\chi}_W = \bar{\chi}_{W^\perp}$

The lifting operator

- For a linear subspace $W \subset \mathbb{R}^n$ and index set $I \subseteq [n]$, we let

  $$\pi_I : \mathbb{R}^n \to \mathbb{R}^I$$

  denote the coordinate projection, and

  $$\pi_I(W) = \{x_I : x \in W\}$$

- The lifting operator $L_I^W : \mathbb{R}^I \to \mathbb{R}^n$ is defined as

  $$L_I^W(z) = \arg\min\{\|x\|_2 : x \in W, x_I = z\}$$

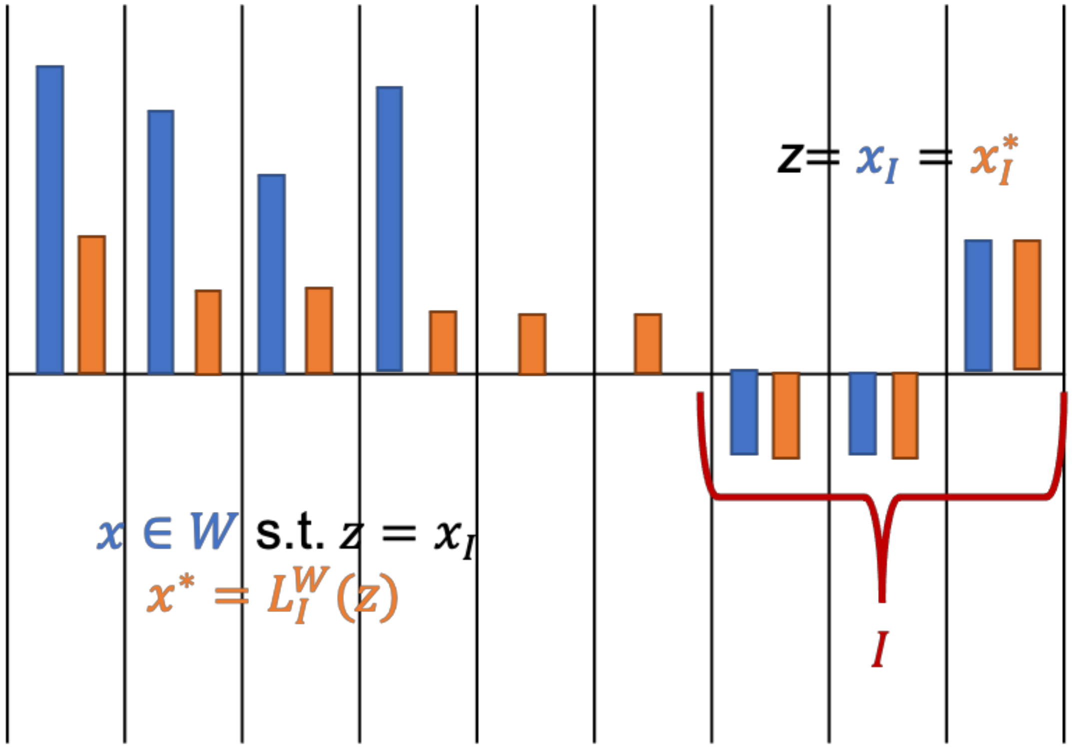
- This is a linear operator; we can efficiently compute a projection matrix $H \in \mathbb{R}^{n \times I}$ such that $L_I^W(z) = Hz$.

- LEMMA:

$$\bar{\chi}_A = \max_{I \subseteq [n]} \|L_I^W\| = \max\left\{\frac{\|L_I^W(z)\|_2}{\|z\|_2} : I \subseteq [n], z \in \pi_I(W) \setminus \{0\}\right\}$$

The lifting operator

$$L_I^W(z) = \arg\min\{\|x\|_2 : x \in W, x_I = z\}$$

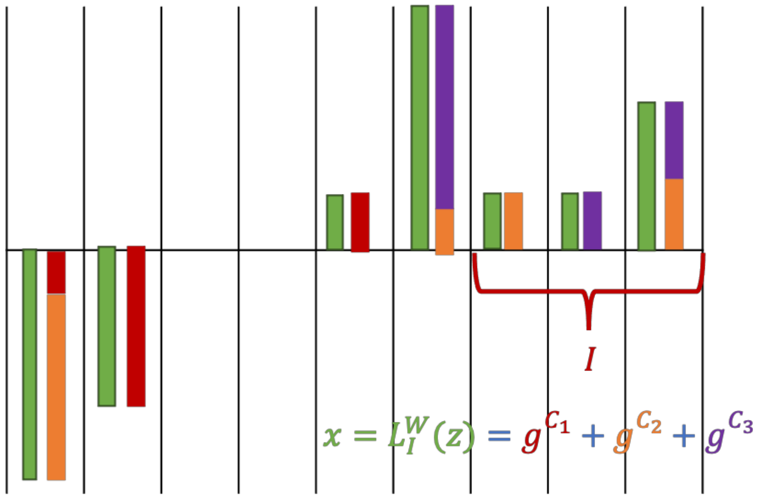


The lifting operator

LEMMA:

$$\kappa_A = \max\left\{\frac{\|L_I^W(z)\|_\infty}{\|z\|_1} : I \subseteq [n], z \in \pi_I(W) \setminus \{0\}\right\}$$

PROOF:



The condition numbers κ_A and $\bar{\chi}_A$

THEOREM: For every matrix $A \in \mathbb{R}^{m\times n}$, $n \geq 2$

$$\sqrt{1+\kappa_A^2} \leq \bar{\chi}_A \leq n\kappa_A$$

Approximability of $\boldsymbol{\kappa_A}$ and $\boldsymbol{\bar{\chi}_A}$:

LEMMA (Tunçel 1999): It is NP-hard to approximate $\boldsymbol{\bar{\chi}_A}$ by a factor better than $2^{\mathrm{poly}(\mathrm{rank}(A))}$

Outline of the lectures

1. Tardos's algorithm for min-cost flows
2. The circuit imbalance measure κ_A and the condition measure $\bar{\chi}_A$
3. Solving LPs: from approximate to exact
4. Optimizing circuit imbalances
5. Interior point methods: basic concepts
6. Layered-least-squares interior point methods

Part 3
Solving LPs:
from approximate to exact

Fast approximate LP algorithms

$$\min c^\top x$$
$$Ax = b$$
$$x \geq 0$$

- ε-approximate solution:
  - Approximately feasible: $\|Ax - b\| \leq \varepsilon(\|A\|_F R + \|b\|)$
  - Approximately optimal: $c^\top x \leq \mathrm{OPT} + \varepsilon \|c\| R$
- Finding an approximate solution with $\log\left(\frac{1}{\varepsilon}\right)$ running time dependence implies a weakly polynomial exact algorithm.

Fast approximate LP algorithms

$$\min c^\top x \quad Ax = b \quad x \geq 0$$

- n variables, m equality constraints, Randomized vs. Deterministic
- Significant recent progress:
  - R $O\left((\mathrm{nnz}(A) + m^2)\sqrt{m}\log^{O(1)}(n)\log\left(\frac{n}{\varepsilon}\right)\right)$ Lee–Sidford '13–'19
  - R $O\left(n^\omega \log^{O(1)}(n)\log\left(\frac{n}{\varepsilon}\right)\right)$ Cohen, Lee, Song '19
  - D $O\left(n^\omega \log^2(n)\log\left(\frac{n}{\varepsilon}\right)\right)$ van den Brand '20
  - R $O\left((mn + m^3)\log^{O(1)}(n)\log\left(\frac{n}{\varepsilon}\right)\right)$ van den Brand, Lee, Sidford, Song '20
  - R $O\left((mn + m^{2.5})\log^{O(1)}(n)\log\left(\frac{n}{\varepsilon}\right)\right)$
    van den Brand, Lee, Liu, Saranurak, Sidford, Song, Wang '21

Some important techniques:

- weighted and stochastic central paths
- fast approximate linear algebra
- efficient data structures

Fast exact LP algorithms with κ_A dependence

$$\min c^\top x$$
$$Ax = b$$
$$x \geq 0$$

- n variables, m equality constraints

THEOREM (Dadush, Natura, V. '20) There exists a poly$(n, m, \log \kappa_A)$ algorithm for solving LP exactly.

- Feasibility: m calls to an approximate solver
- Optimization: mn calls to an approximate solver

with $\varepsilon = 1/(\text{poly}(n, \kappa_A))$. Using van den Brand '20, this gives a deterministic exact $O(mn^{\omega+1} \log^2(n) \log(\kappa_A + n))$ time LP optimization algorithm

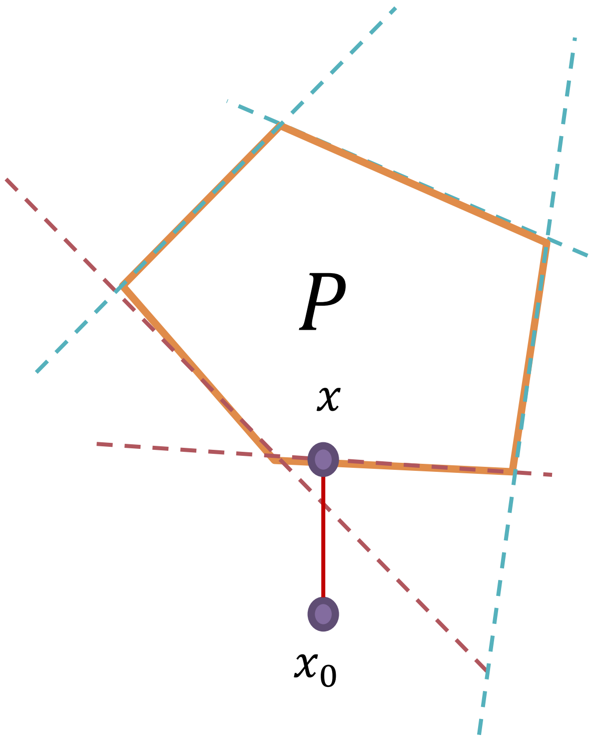
- Generalization of Tardos '86 for real constraint matrices and with directly working with approximate solvers.
- Main difference: arguments in Tardos '86 heavily rely on integrality assumptions

Hoffman's proximity theorem

Polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, point $x_0 \notin P$, norms $\|.\|_\alpha$, $\|.\|_\beta$

THEOREM (Hoffman, 1952): There exists a constant $H_{\alpha,\beta}(A)$ such that

$$\exists x \in P\colon \ \|x - x_0\|_\alpha \leq H_{\alpha,\beta}(A)\|(Ax_0 - b)^+\|_\beta$$



Alan J. Hoffman
1924-2021

LP in subspace form

- Matrix form: $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, \mathrm{c} \in \mathbb{R}^n$

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

- Subspace form: $\mathrm{W} = \ker(A),\ d \in \mathbb{R}^n$ s.t. $Ad = b$

$$\begin{aligned} \min c^\top x \\ x \in W + d \\ x \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \max\ d^\top (c - s) \\ s \in W^\perp + c \\ s \geq 0 \end{aligned}$$

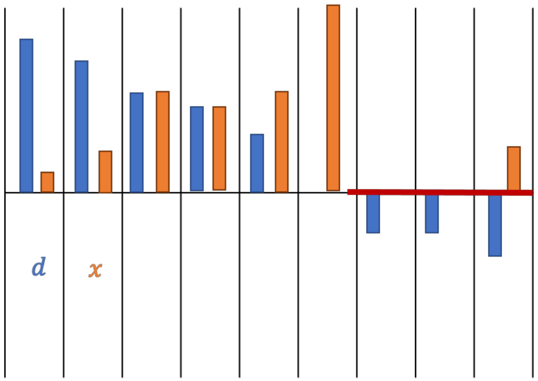
Proximity theorem with κ_A

THEOREM: For $A \in \mathbb{R}^{m\times n}, d \in \mathbb{R}^n$, consider the system

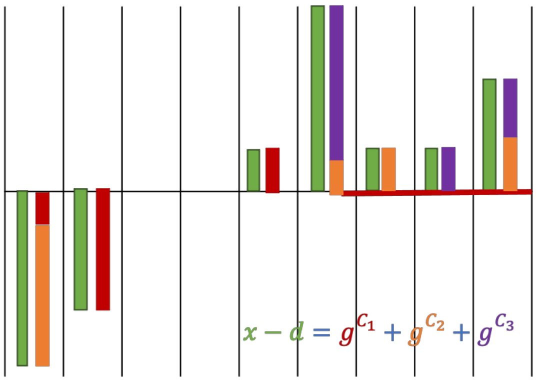
$$\mathrm{x} \in W + d, x \geq 0.$$

There exists a feasible solution x such that

$$\|x - d\|_\infty \leq \kappa_W \|d^-\|_1$$



PROOF:



Linear feasibility algorithm

Linear feasibility problem

$$x \in W + d, \qquad x \geq 0.$$

- Recursive algorithm using a **stronger** problem formulation:

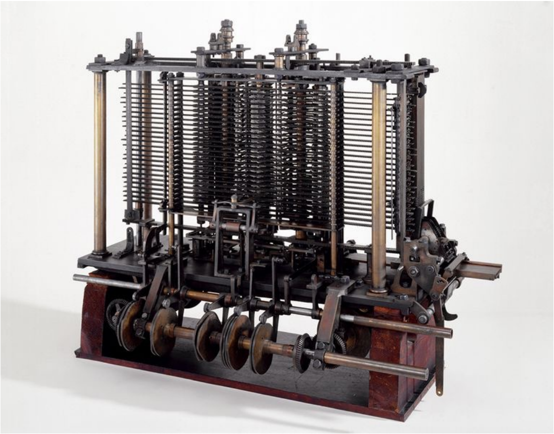
$$x \in W + d, \qquad x \geq 0.$$
$$\|x - d\|_\infty \leq C' \kappa_W^2 \|d^-\|_1$$

- Black box oracle for $\varepsilon = 1/(\mathrm{poly}(n, \kappa_A))$

$$x \in W + d$$

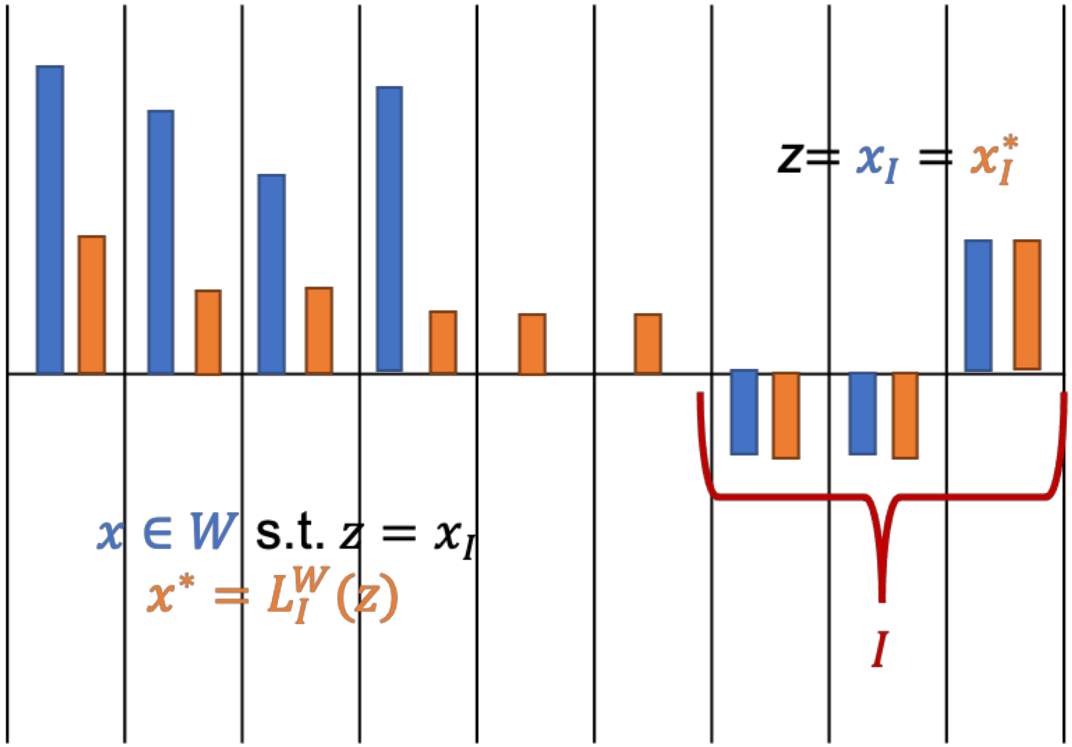
proximity $\|x - d\|_\infty \leq C \kappa_W \|d^-\|_1$

error $\|x^-\|_\infty \leq \varepsilon \|d^-\|_1$



The lifting operator

$$L_I^W(z) = \arg\min\{\|x\|_2 : x \in W, x_I = z\}$$



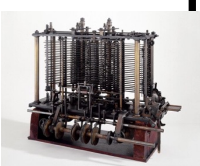
The linear feasibility algorithm

Problem $\mathcal{F}(W, d)$

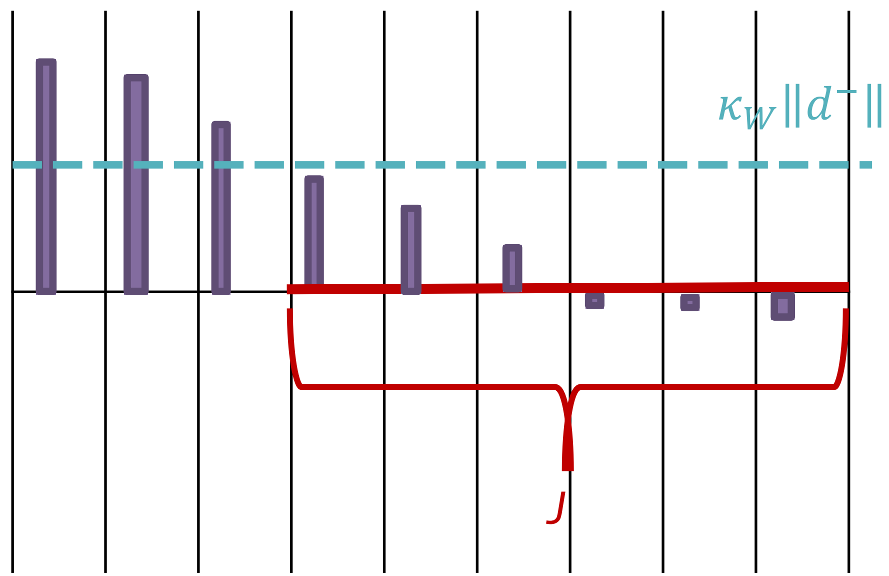
$$x \in W + d$$
$$\|x - d\|_\infty \le C' \kappa_W^2 \|d^-\|_1$$
$$x \ge 0$$

1. Call the black box solver to find a solution z for $\varepsilon = 1/(\kappa_W n)^4$

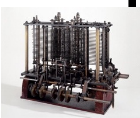
$$z \in W + d$$
$$\|z - d\|_\infty \le C \kappa_W \|d^-\|_1$$
$$\|z^-\|_\infty \le \varepsilon \|d^-\|_1$$



2. Set $J = \{i \in [n] : z_i < \kappa_W \|d^-\|_1\}$; assume $J \ne [n]$.
3. Recursively obtain $\tilde{x} \in \mathbb{R}_+^J$ from $\mathcal{F}(\pi_J(W), z_J)$
4. Return $x = z + L_J^W(\tilde{x} - z_J)$



1. Call the black box solver to find a solution z for $\varepsilon = 1/(\kappa_W n)^4$

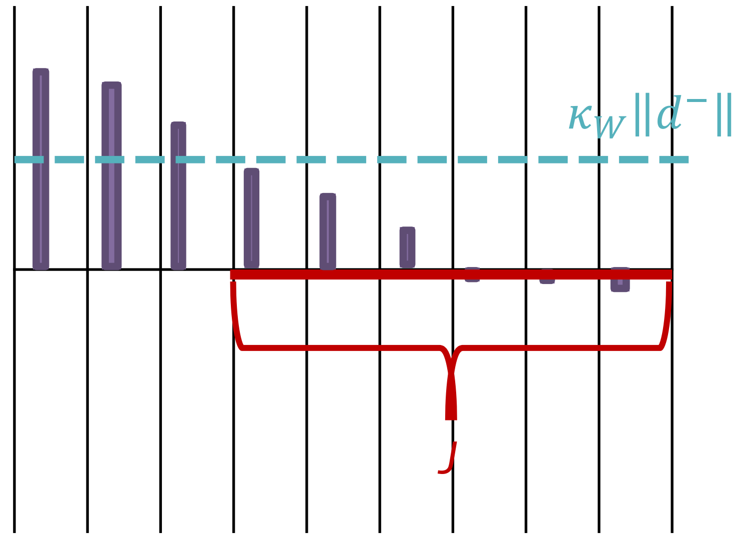


$$\boxed{\begin{gathered} z \in W + d \\ \|z - d\|_\infty \le C\kappa_W \|d^-\|_1 \\ \|z^-\|_\infty \le \varepsilon \|d^-\|_1 \end{gathered}}$$

2. Set $J = \{i \in [n]: z_i < \kappa_W \|d^-\|_1\}$; assume $J \ne [n]$.
3. Recursively obtain $\tilde{x} \in \mathbb{R}^J_+$ from $\mathcal{F}(\pi_J(W), z_J)$
4. Return $x = z + L_J^W(\tilde{x} - z_J)$

Problem $\mathcal{F}(W, d)$

$$\boxed{\begin{gathered} x \in W + d \\ \|x - d\|_\infty \le C' \kappa_W^2 \|d^-\|_1 \\ x \ge 0 \end{gathered}}$$



The linear feasibility algorithm

$$J = \{i \in [n] : z_i < \kappa_W \|d^-\|_1\};$$

- If $J = [n]$, then we replace d by its projection to $W^\perp$
- Bound n on the number of recursive calls; can be decreased to m
- $O(mn^{\omega+o(1)} \log(\kappa_W + n))$ feasibility algorithm using van den Brand '20.

Certification

- In case of infeasibility we return an exact Farkas certificate
- κ_W is hard to approximate within $2^{O(n)}$ Tunçel 1999
- We use an estimate M in the algorithm
- The algorithm may fail if $\left\|L_J^W(\tilde{x}-z_J)\right\|_\infty > M\left\|\tilde{x}-z_J\right\|_1$
- In this case, we restart with
$$\max\left\{M^2, \frac{\left\|L_J^W(\tilde{x}-z_J)\right\|_\infty}{\left\|\tilde{x}-z_J\right\|_1}\right\}$$
- Our estimate never overshoots κ_W by much, but can be significantly better.

Proximity for optimization

$$\begin{array}{c} \min c^\top x \\ x \in W + d \\ x \geq 0 \end{array} \qquad \begin{array}{c} \max\ d^\top (c - s) \\ s \in W^\perp + c \\ s \geq 0 \end{array}$$

THEOREM: Let $s \in W^\top + c, s \geq 0$ be a feasible dual solution, and assume the primal is also feasible. Then there exists a primal optimal $x^* \in W + d, x^* \geq 0$ such that

$$\|x^* - d\|_\infty \leq \kappa_W \left(\|d^-\|_1 + \left\|d_{\mathrm{supp}(s)}\right\|_1 \right).$$

Optimization algorithm

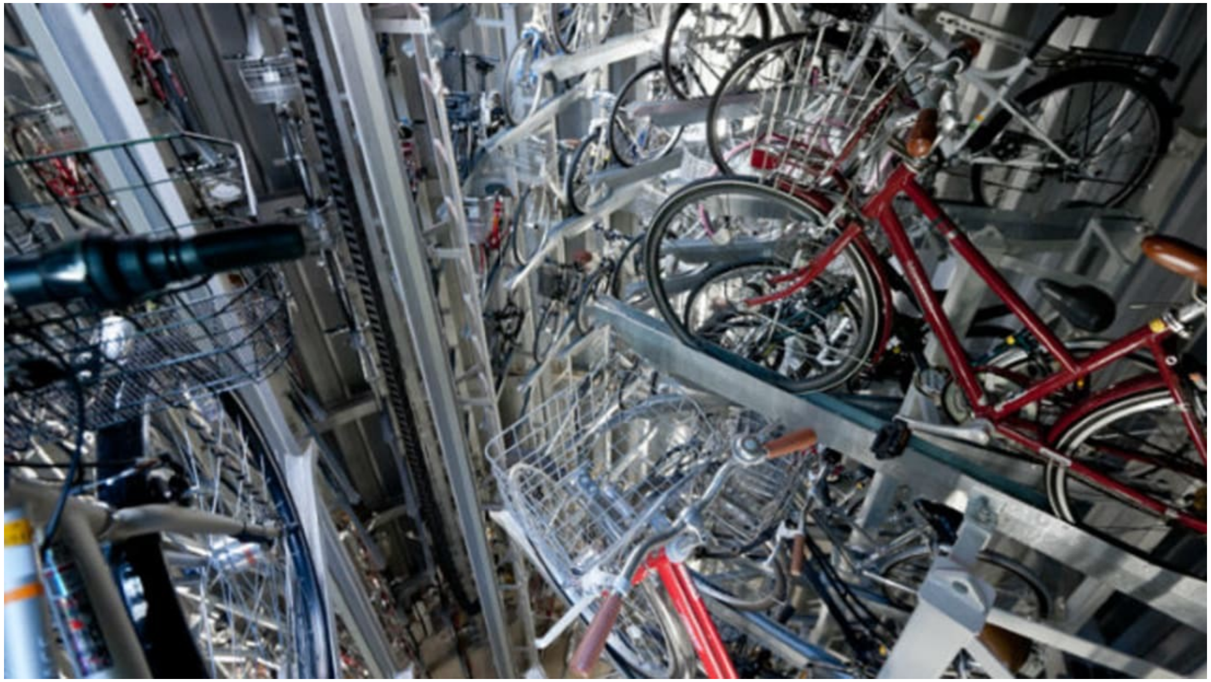
$$\begin{array}{ccc} \min c^\top x & \qquad & \max\ d^\top(c-s) \\ x \in W + d & & s \in W^\perp + c \\ x \geq 0 & & s \geq 0 \end{array}$$

- nm calls to the black box solver
- $\leq n$ **Outer Loops**, each comprising $\leq m$ **Inner Loops**
- Each Outer Loop finds $\tilde{d}$ with $\|d - \tilde{d}\|$ "small", and (x, s) primal and dual optimal solutions to
$$\min c^\top x \ \ s.t. x \in W + \tilde{d}, d \geq 0$$
- Using proximity, we can use this to conclude $x_I > 0$ for a certain variable set $I \subseteq n$ and recurse.

Outline of the lectures

1. Tardos's algorithm for min-cost flows
2. The circuit imbalance measure κ_A and the condition measure $\bar{\chi}_A$
3. Solving LPs: from approximate to exact
4. Optimizing circuit imbalances
5. Interior point methods: basic concepts
6. Layered-least-squares interior point methods

Part 4
Optimizing circuit imbalances



Diagonal rescaling of LP

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

Positive diagonal matrix $D \in \mathbb{R}^{n\times n}$

$$\begin{aligned} \min\ & (Dc)^\top x' \\ ADx' &= b \\ x' &\geq 0 \end{aligned} \qquad\qquad \begin{aligned} \max\ & b^\top y' \\ (AD)^\top y' + s' &= Dc \\ s' &\geq 0 \end{aligned}$$

Mapping between solutions:

$$x' = D^{-1}x, \qquad y' = y, \qquad s' = Ds$$

Diagonal rescaling of LP

Positive diagonal matrix $D \in \mathbb{R}^{n\times n}$

$$\begin{array}{c} \min\ (Dc)^\top x' \\ ADx' = b \\ x' \geq 0 \end{array} \qquad\qquad \begin{array}{c} \max b^\top y' \\ (AD)^\top y' + s' = Dc \\ s' \geq 0 \end{array}$$

Mapping between solutions:

$$x' = D^{-1}x, \qquad y' = y, \qquad s' = Ds$$

- Natural symmetry of LPs and many LP algorithms.
- The Central Path is invariant under diagonal scaling.
- Most “standard” interior point methods are invariant.

Dependence on the constraint matrix only

$$\min c^\top x,\ Ax = b\ \ x \geq 0$$

- Algorithms with running time dependent only on A, but not on b and c.
- Combinatorial LP's: integer matrix $A \in \mathbb{Z}^{m\times n}$.

  $$\Delta_A = \max\{|\det(B)|: B \text{ submatrix of } A\}$$

  Tardos '86: poly$(n, m, \log \Delta_A)$ LP algorithm ✗
- Layered-least-squares (LLS) Interior Point Method
  Vavasis-Ye '96: poly$(n, m, \log \bar{\chi}_A)$ LP algorithm in the real model of computation ✗
  $\bar{\chi}_A$: condition number
- Dadush-Huiberts-Natura-V '20: poly$(n, m, \log \bar{\chi}^*_A)$ ✓
  $\bar{\chi}^*_A$: optimized version of $\bar{\chi}_A$

Optimizing κ_A and $\bar{\chi}_A$ by rescaling

$\mathcal{D}$ = set of $n \times n$ positive diagonal matrices

$$\kappa_A^* = \inf\{\kappa_{AD} : D \in \mathcal{D}\}$$

$$\bar{\chi}_A^* = \inf\{\bar{\chi}_{AD} : D \in \mathcal{D}\}$$

- A scaling invariant algorithm with $\bar{\chi}_A$ dependence automatically yields $\bar{\chi}_A^*$ dependence.
- Recall $\sqrt{1+\kappa_A^2} \le \bar{\chi}_A \le n\kappa_A$.

THEOREM (Dadush-Huiberts-Natura-V '20): Given $A \in \mathbb{R}^{m \times n}$, in $O(n^2m^2 + n^3)$ time, one can

- approximate the value κ_A within a factor $(\kappa_A^*)^2$, and
- compute a rescaling $D \in \mathcal{D}$ satisfying $\kappa_{AD} \le (\kappa_A^*)^3$.

THEOREM (Tunçel 1999): It is NP-hard to approximate $\bar{\chi}_A$ (and thus κ_A) by a factor better than $2^{\mathrm{poly}(\mathrm{rank}(A))}$

Approximating κ_A^*

$\mathcal{D}$ = set of $n \times n$ positive diagonal matrices

$$\kappa_A^* = \inf\{\kappa_{AD} : D \in \mathcal{D}\}$$

- EXAMPLE: Let $\ker(A) = \mathrm{span}((0,1,1,\mathrm{M}),(1,0,\mathrm{M},1)\ )$

Pairwise circuit imbalances

- For a circuit C, there exists a vector $g^C \in \mathbb{R}^C$ unique up to a scalar multiplier such that

$$\sum_{i \in C} g_i^C a_i = 0$$

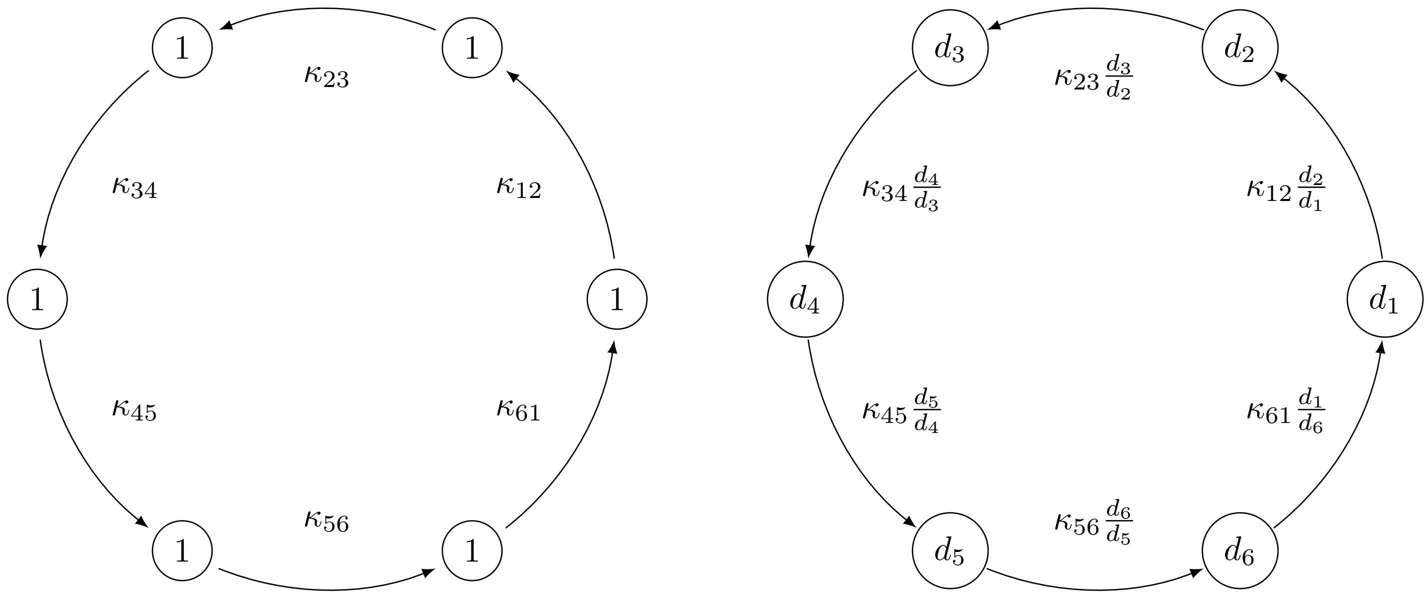
- $\mathcal{C}_A$: set of all circuits.
- For any $i, j \in [n]$,

$$\kappa_{ij} = \max\left\{\frac{\left|g_j^C\right|}{\left|g_i^C\right|} : C \in \mathcal{C}_A, \text{s.t.}\, i, j \in C\right\}$$

- The circuit imbalance measure is

$$\kappa_A = \max_{i,j \in [n]} \kappa_{ij}$$

Cycles are invariant under scaling



LEMMA For any directed cycle H on $\{1,2,\dots,n\}$

$$(\kappa_A^*)^{|H|} \geq \prod_{(i,j)\in H} \kappa_{ij}$$

Circuit imbalance min-max formula

THEOREM (Dadush-Huiberts-Natura-V '20):

$$\kappa_A^* = \max\left\{\left(\prod_{(i,j)\in H} \kappa_{ij}\right)^{1/|H|} : \ H \text{ directed cycle on } \{1,2,\dots,n\}\right\}$$

<u>PROOF:</u>

Circuit imbalance min-max formula

THEOREM (Dadush-Huiberts-Natura-V '20):

$$\kappa_A^* = \max\left\{\left(\prod_{(i,j)\in H} \kappa_{ij}\right)^{1/|H|} : H \text{ directed cycle on } \{1,2,\ldots,n\}\right\}$$

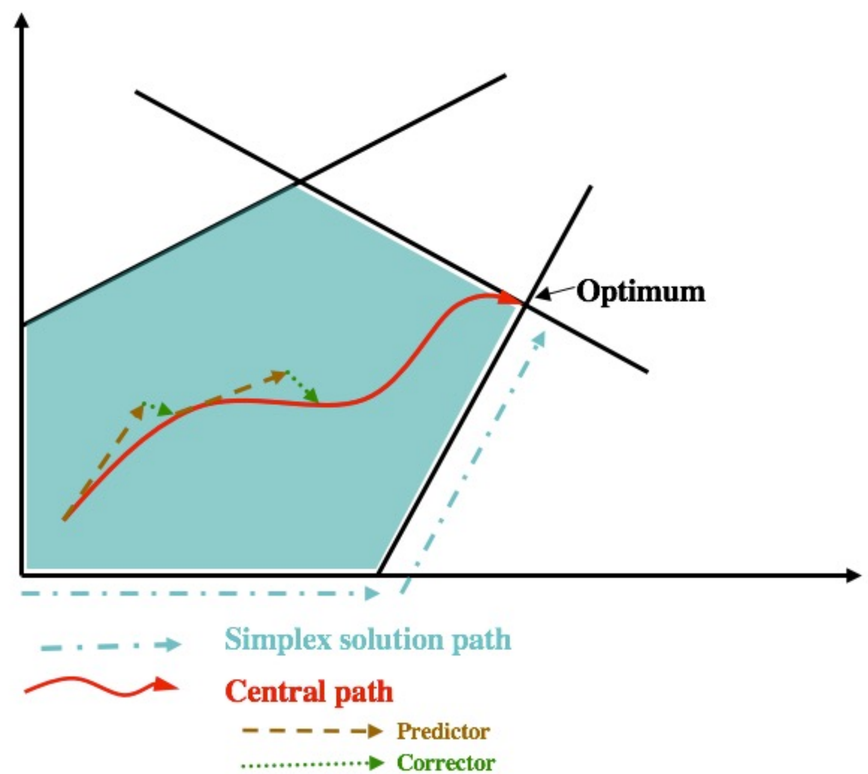
- BUT: Computing the κ_{ij} values is NP-complete…
- **LEMMA:** For any circuit $C \in \mathcal{C}_A$ s.t. $i, j \in C$,

$$\frac{|g_j^C|}{|g_i^C|} \geq \frac{\kappa_{ij}}{(\kappa_W^*)^2}$$

Outline of the lectures

1. Tardos's algorithm for min-cost flows
2. The circuit imbalance measure κ_A and the condition measure $\bar{\chi}_A$
3. Solving LPs: from approximate to exact
4. Optimizing circuit imbalances
5. Interior point methods: basic concepts
6. Layered-least-squares interior point methods

Part 5
Interior point methods: basic concepts



Primal and dual LP

- Matrix form: $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, \mathrm{c} \in \mathbb{R}^n$

$$\begin{array}{cc} \min c^\top x & \max b^\top y \\ Ax = b & A^\top y + s = c \\ x \geq 0 & \mathrm{s} \geq 0 \end{array}$$

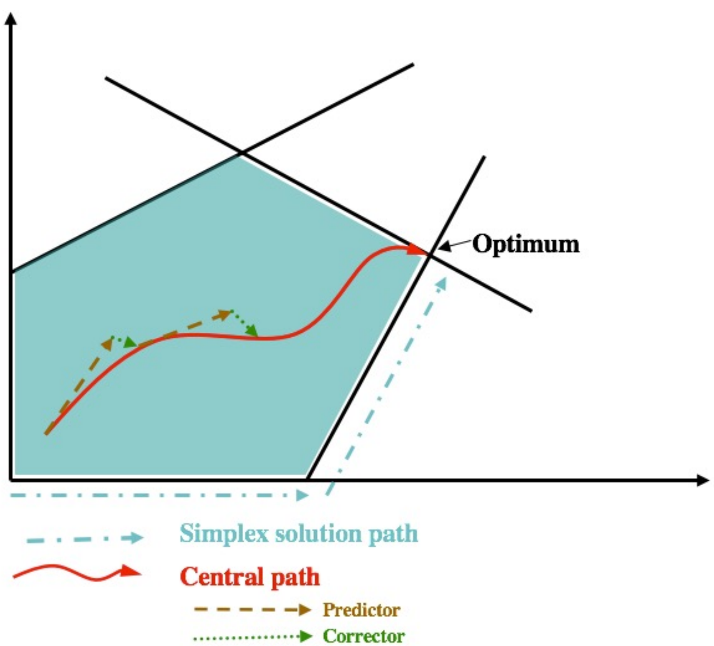
- Subspace form: $\mathrm{W} = \ker(A),\ d \in \mathbb{R}^n$ s.t. $Ad = b$

$$\begin{array}{cc} \min c^\top x & \max\ d^\top (c - s) \\ x \in W + d & s \in W^\top + c \\ x \geq 0 & s \geq 0 \end{array}$$

- Complementary slackness: Primal and dual solutions (x, s) are optimal if $x^\top s = 0$: for each $i \in [n]$, either $x_i = 0$ or $s_i = 0$.
- Optimality gap:

$$c^\top x - d^\top (c - s) = x^\top s.$$

The central path

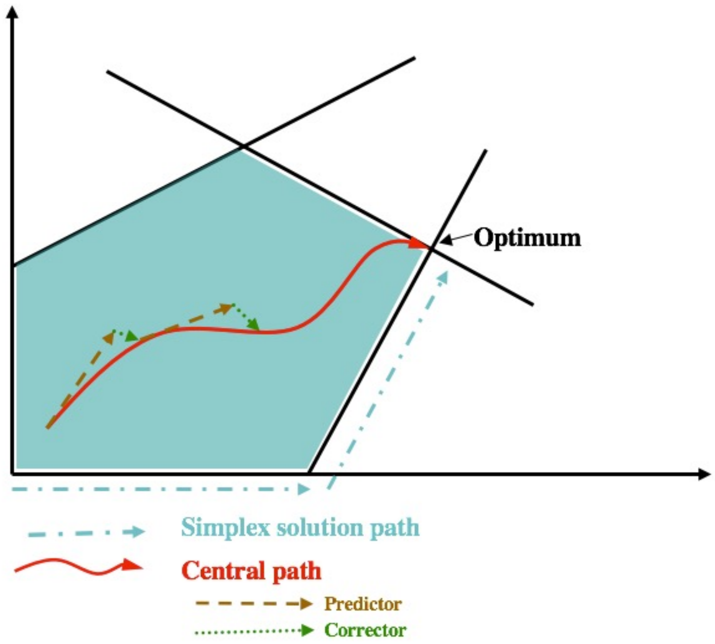


- For each $\mu > 0$, there exists a unique solution $w(\mu) = (x(\mu), y(\mu), s(\mu))$ such that
$$x(\mu)_i s(\mu)_i = \mu \quad \forall i \in [n]$$
the central path element for μ.
- The central path is the algebraic curve formed by $\{w(\mu): \mu > 0\}$
- For $\mu \to 0$, the central path converges to an optimal solution $w^* = (x^*, y^*, s^*)$.
- The optimality gap is $s(\mu)^\top x(\mu) = n\mu$.
- Interior point algorithms: walk down along the central path with μ decreasing geometrically.

The Mizuno-Todd-Ye Predictor-Corrector Algorithm

- Start from point $w_0 = (x_0, y_0, s_0)$ 'near' the central path at some $\mu_0 > 0$.
- Alternate between
  - Predictor steps: 'shoot down' the central path, decreasing μ by a factor at least $1 - \beta/n$.
    May move slightly 'farther' from the central path.
  - Corrector steps: do not change parameter μ, but move back 'closer' to the central path.

Within $O(n)$ iterations, μ decreases by a factor 2.



The predictor step

- Step direction $\Delta w = (\Delta x, \Delta y, \Delta s)$

$$\boxed{\begin{aligned} A\Delta x &= 0 \\ A^\top \Delta y + \Delta s &= 0 \\ s_i \Delta x_i + x_i \Delta s_i &= -x_i s_i \; \forall i \in [n] \end{aligned}}$$

- Pick the largest $\alpha \in [0,1]$ such that w' is still "close enough" to the central path
  $w' = w + \alpha\Delta w = (x + \alpha\Delta x, y + \alpha\Delta y, s + \alpha\Delta s)$

- Long step: $|\Delta x_i \Delta s_i|$ small for every $i \in [n]$
- New optimality gap is $(1 - \alpha)\mu$.

The predictor step – subspace view

$$\boxed{\begin{aligned} A\Delta x &= 0 \\ A^\top \Delta y + \Delta s &= 0 \\ s_i \Delta x_i + x_i \Delta s_i &= -x_i s_i \ \forall i \in [n] \end{aligned}}$$

- Assume the current point $w = (x, y, s)$ is on the central path. The steps can be found as minimum norm projections in the $({}^1\!/_x)$ and $({}^1\!/_s)$ rescaled norms

$$\Delta x = \arg\min \sum_{i=1}^{n} \left(\frac{x_i + \Delta x_i}{x_i}\right)^2 \ \text{s.t.}\ x \in W = \ker(A)$$

$$\Delta s = \arg\min \sum_{i=1}^{n} \left(\frac{s_i + \Delta s_i}{s_i}\right)^2 \ \text{s.t.}\ s \in W^\perp = \operatorname{im}(A^\top)$$

Some recent progress on interior point methods

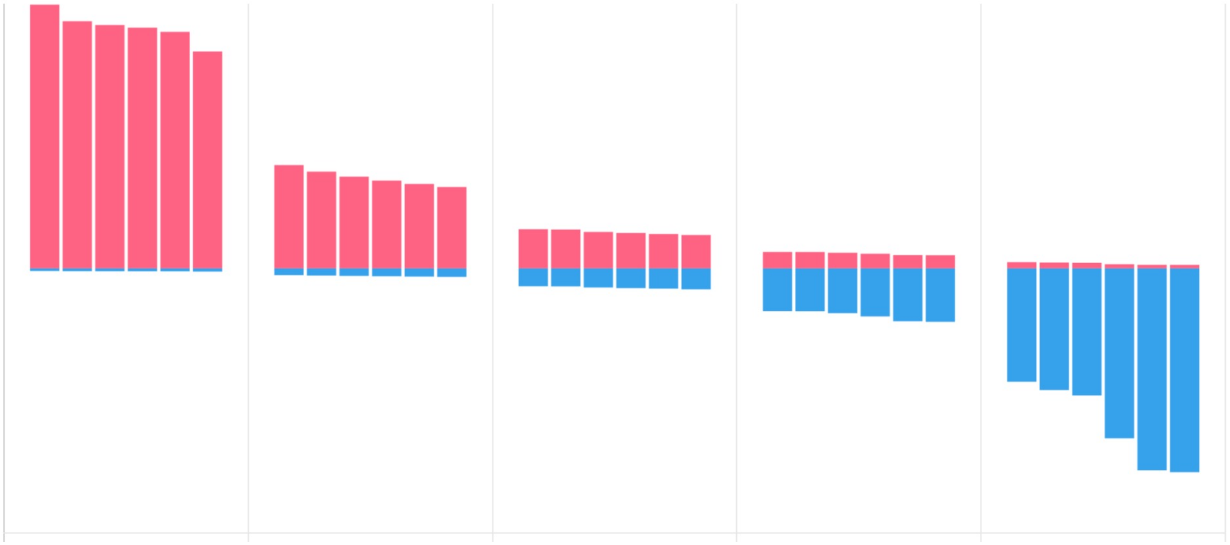
- Tremendous recent progress on fast approximate variants LS'14–'19, CLS'19,vdB'20,vdBLSS'20,vdBLLSSSW'21
- Fast approximate algorithms for combinatorial problems flows, matching and MDPs: DS'08, M'13, M'16, CMSV'17, AMV'20, vdBLNPTSSW'20, vdBLLSSSW'21

Outline of the lectures

1. Tardos's algorithm for min-cost flows
2. The circuit imbalance measure κ_A and the condition measure $\bar{\chi}_A$
3. Solving LPs: from approximate to exact
4. Optimizing circuit imbalances
5. Interior point methods: basic concepts
6. Layered-least-squares interior point methods

Part 6

Layered-least-squares interior point methods



Layered-least-squares (LLS) Interior Point Methods:
Dependence on the constraint matrix only

$$\bar{\chi}_A^* = \inf\{\bar{\chi}_{AD} : D \in \mathcal{D}\}$$

- Vavasis-Ye '96: $O(n^{3.5}\log(\bar{\chi}_A + n))$ iterations
- Monteiro-Tsuchiya '03 $O(n^{3.5}\log(\bar{\chi}_A^* + n) + n^2 \log\log 1/\varepsilon)$ iterations
- Lan-Monteiro-Tsuchiya '09 $O(n^{3.5}\log(\bar{\chi}_A^* + n))$ iterations, but the running time of the iterations depends on b and c
- Dadush-Huiberts-Natura-V '20: scaling invariant LLS method with $O(n^{2.5}\log(n)\log(\bar{\chi}_A^* + n))$ iterations

Near monotonicity of the central path

LEMMA For $w = (x, y, s)$ on the central path, and for any solution $w' = (x', y', s')$ s.t. $(x')^\top s' \leq x^\top s$, we have

$$\sum_{i=1}^{n} \frac{x_i'}{x_i} + \frac{s_i'}{s_i} \leq 2n$$

PROOF:

IPM learns gradually improved upper bounds on the optimal solution.

Variable fixing…—or not?

LEMMA After every iteration, there exists variables x_i and s_j such that

$$\frac{1}{O(n)} \leq \frac{x_i}{x_i^*}, \frac{s_j}{s_j^*} \leq O(n)$$

For the optimal (x^*, y^*, s^*). Thus, x_i and s_j have *"converged"* to their final values.

- PROOF: Can be shown using the form of the predictor step:

$$\Delta x = \arg\min \sum_{i=1}^{n} \left(\frac{x_i + \Delta x_i}{x_i}\right)^2 \quad \text{s.t. } x \in W$$

$$\Delta s = \arg\min \sum_{i=1}^{n} \left(\frac{s_i + \Delta s_i}{s_i}\right)^2 \quad \text{s.t. } s \in W^\perp$$

and bounds on the stepsize.

Variable fixing…—or not?

LEMMA After every iteration, there exists variables x_i and s_j such that

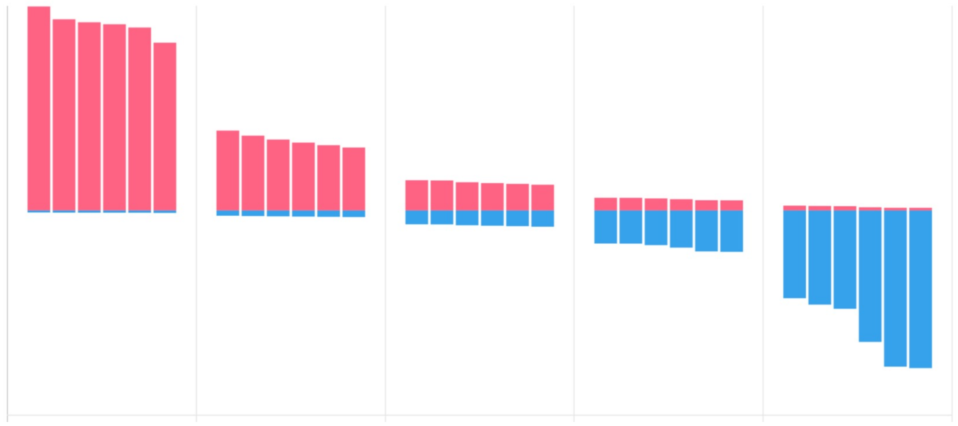
$$\frac{1}{O(n)} \le \frac{x_i}{x_i^*}, \frac{s_j}{s_j^*} \le O(n)$$

Thus, x_i and s_j have *"converged"* to their final values.

We cannot identify these indices, just show their existence

Layered least squares methods

- Instead of the standard predictor step, split the variables into layers.
- Variables on different layers “behave almost like separate LPs”
- Force new primal and dual variables that must have converged.



Recap: the lifting operator and κ_A

- For a linear subspace $W \subset \mathbb{R}^n$ and index set $I \subseteq [n]$, we let
$$\pi_I : \mathbb{R}^n \to \mathbb{R}^I$$
denote the coordinate projection, and
$$\pi_I(W) = \{x_I : x \in W\}$$

- The lifting operator $L_I^W : \mathbb{R}^I \to \mathbb{R}^n$ is defined as
$$L_I^W(z) = \arg\min\{\|x\|_2 : x \in W, x_I = z\}$$

- **LEMMA:** $\kappa_A = \max\left\{\frac{\|L_I^W(z)\|_\infty}{\|z\|_1} : z \in \pi_I(W)\right\}$

- For every $z \in \pi_I(W)$, $x = L_I^W(z) \in W$ s.t.
$$x_I = z, \text{and } \|x\|_\infty \le \kappa_A \|z\|_1$$

Motivating the layering idea: final rounding step in standard IPM

$$\min c^\top x \qquad\qquad \max b^\top y$$
$$Ax = b \qquad\qquad A^\top y + s = c$$
$$x \geq 0 \qquad\qquad s \geq 0$$

- Limit optimal solution (x^*, y^*, s^*), and optimal partition $[n] = B \cup N$ s.t. $B = \mathrm{supp}(x^*),\ N = \mathrm{supp}(s^*)$.
- Given (x, y, s) near central path with 'small enough' $\mu = s^\top x/n$ such that for every $i \in [n]$, either x_i or s_i very small.
- Assume that we can correctly guess
$$B = \{i: x_i > M\sqrt{\mu}\}, \qquad N = \{i: s_i > M\sqrt{\mu}\}$$

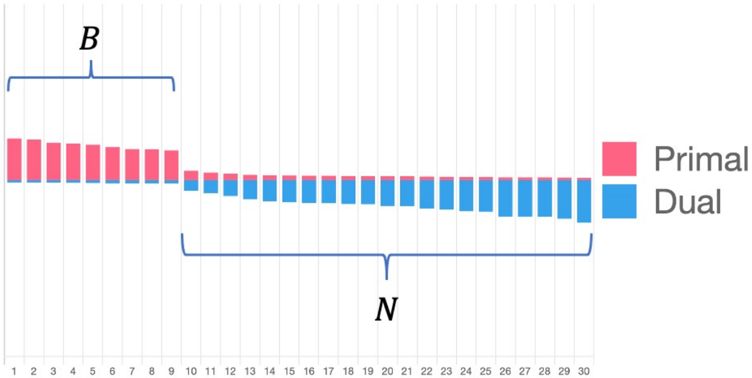
- Assume we have a partition B, N, we have

$$i \in B\colon\ x_i > M\sqrt{\mu}, \qquad s_i < \sqrt{\mu}/M$$
$$i \in N\colon x_i < \sqrt{\mu}/M, \qquad s_i > M\sqrt{\mu}$$

- Goal: move to $\bar{x} = x + \Delta x,\ \ \bar{y} = y + \Delta y, \bar{s} = s + \Delta s$
s.t. $\mathrm{supp}(\bar{x}) \subseteq B$, $\mathrm{supp}(\bar{s}) \subseteq N$. Then, $\bar{x}^\top \bar{s} = 0$: optimal solution.

- Choice:

$$\Delta x = -L_N^W(x_N), \qquad \Delta s = -L_B^W(s_B)$$

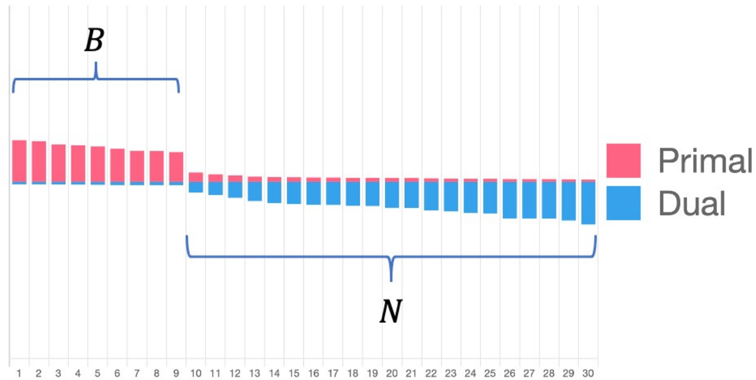


Layered-least-squares step

Assume we have a partition B, N, with

$i \in B$: $x_i > M\sqrt{\mu}$, $\quad s_i < \sqrt{\mu}/M$

$i \in N$: $x_i < \sqrt{\mu}/M$, $\quad s_i > M\sqrt{\mu}$



Standard primal predictor step:

$$\Delta x = \arg\min \sum_{i=1}^{n} \left(\frac{x_i + \Delta x_i}{x_i}\right)^2$$

$$\text{s.t. } \Delta x \in W$$

Vavasis-Ye LLS step with layers (B, N)*:*

$$\Delta x_N = \arg\min \sum_{i \in N} \left(\frac{x_i + \Delta x_i}{x_i}\right)^2$$

$$\text{s.t. } \Delta x \in W$$

$$\Delta x_B = \arg\min \sum_{i \in B} \left(\frac{x_i + \Delta x_i}{x_i}\right)^2$$

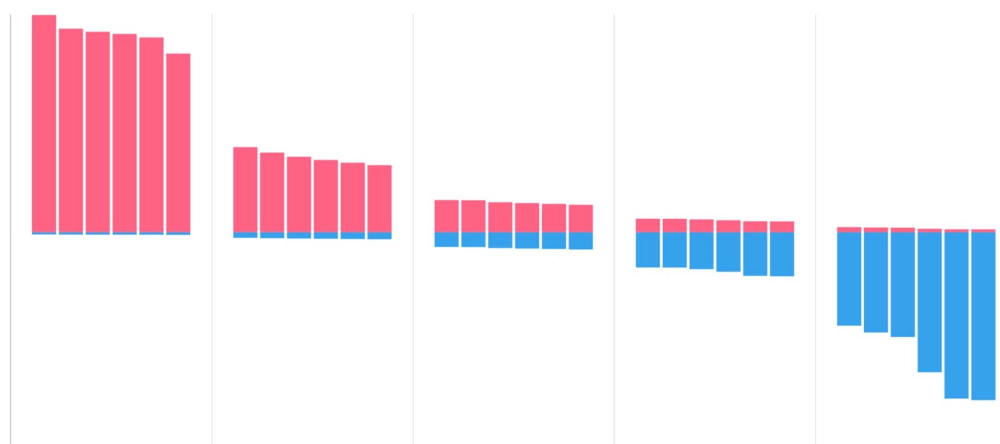
$$\text{s.t. } (\Delta x_B, \Delta x_N) \in W$$

Layered-least-squares step

Vavasis-Ye ‘96

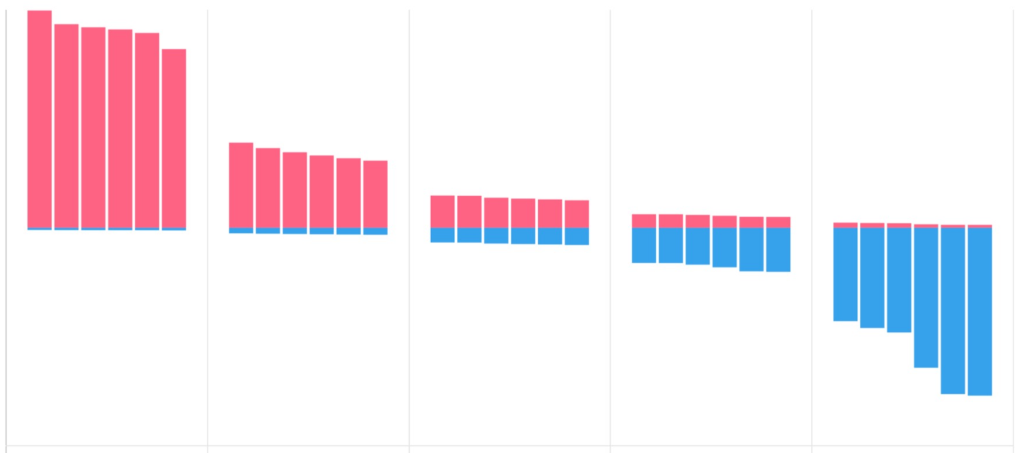
- Order variables decreasingly as $x_1 \geq x_2 \geq \cdots \geq x_n$
- Arrange variables into layers $(J_1, J_2, \ldots, J_t)$; start a new layer when
$$x_i > O(n^c)\, \bar{\chi}_A x_{i+1}$$
- Primal step direction by least squares problems from backwards, layer-by-layer
- Lifting costs from lower layers low
- Dual step in the opposite direction

Not scaling invariant!



Progress measure: crossover events
Vavasis-Ye'96

- **DEFINITION:** The variables x_i and x_j cross over between μ and μ', $\mu > \mu'$, if
  - $O(n^c)(\bar{\chi}_A)^n\, x_j(\mu) \geq x_i(\mu)$
  - $O(n^c)(\bar{\chi}_A)^n\, x_j(\mu'') < x_i(\mu'')$ for any $\mu'' \leq \mu'$
- **LEMMA:** In the Vavasis-Ye algorithm, a crossover event happens every $O(n^{1.5}\log(\bar{\chi}_A + n))$ iterations, totalling to $O(n^{3.5}\log(\bar{\chi}_A + n))$.

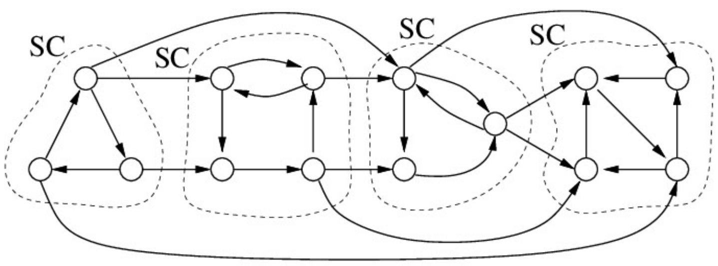


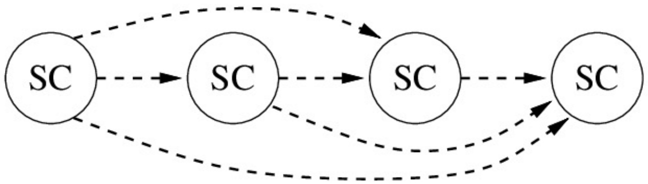
Scaling invariant layering DNHV'20

- Instead of the ratios x_i/x_j, we consider the rescaled circuit imbalance measures $\kappa_{ij}x_i/x_j$
- Layers: strongly connected components of the arcs

$$(i,j):\ \frac{\kappa_{ij}x_i}{x_j} > \frac{1}{poly(n)}$$

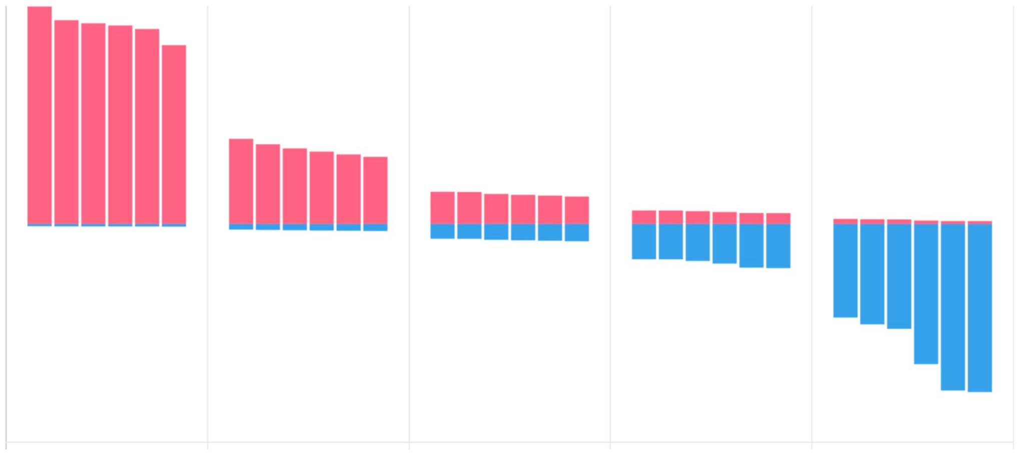
The κ_{ij} values are not known: increasingly improving estimates.





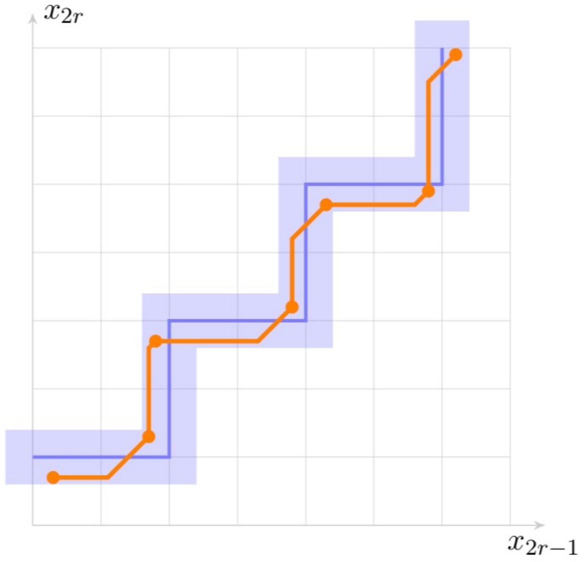
Scaling invariant crossover events
Vavasis-Ye'96

- **DEFINITION:** The variables x_i and x_j cross over between μ and μ', $\mu > \mu'$, if
  - $O(n^c)(\bar{\chi}_A)^n\, x_j(\mu) \geq \kappa_{ij} x_i(\mu)$
  - $O(n^c)(\bar{\chi}_A)^n\, x_j(\mu'') < \kappa_{ij} x_i(\mu'')$ for any $\mu'' \leq \mu'$
- Amortized analysis, resulting in improved $O(n^{2.5}\log(n)\log(\bar{\chi}_A + n))$ iteration bound.



Limitation of IPMs

- THEOREM (Allamigeon–Benchimol–Gaubert–Joswig '18): No standard path following method can be strongly polynomial.
- Proof using **tropical geometry**: studies the tropical limit of a family of parametrized linear programs.



Future directions

- Circuit imbalance measure: key parameter for strongly polynomial solvability.
- LP classes with existence of strongly polynomial algorithms open:
  - LPs with 2 nonzeros per column in the constraint matrix, equivalently: min cost generalized flows
  - Undiscounted Markov Decision Processes
- Extend the theory of circuit imbalances more generally, to convex programming and integer programming.

Thank you!

Postdoc position open

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

erc European Research Council

Application deadline: 5 June