LINEAR PROGRAMMING
AND CIRCUIT IMBALANCES

László Végh

IPCO Summer School
Georgia Tech, May 2021

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

\[ \begin{align*}
\text{min } & \ c^T x \\
\text{Ax } & = \ b \\
x & \geq 0
\end{align*} \]
Facets of linear programming

Discrete
- Basic solutions
- Polyhedral combinatorics
- Exact solution

Continuous
- Continuous solutions
- Convex program
- Approximate solution
Linear programming algorithms

- $n$ 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

\[
\min c^\top x \\
Ax = b \\
x \geq 0
\]
Weakly vs strongly polynomial algorithms for LP

- $n$ variables, $m$ constraints, total encoding $L$.

- **Strongly polynomial algorithm:**
  - $\text{poly}(n, m)$ elementary arithmetic operations (+, −, ×, ÷, ≥), 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

\[
\min c^T x, \ Ax = b \quad x \geq 0
\]

- 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: \( \text{poly}(n, m, \log \Delta_A) \) \text{ black box } LP \text{ algorithm}

- Layered-least-squares (LLS) Interior Point Method
  Vavasis-Ye ’96: \( \text{poly}(n, m, \log \bar{\chi}_A) \) \text{ LP algorithm in the real model of computation}
  \( \bar{\chi}_A \): condition number

- Dadush-Huiberts-Natura-V ’20: \( \text{poly}(n, m, \log \bar{\chi}_A^*) \)
  \( \bar{\chi}_A^* \): optimized version of \( \bar{\chi}_A \)
Outline of the lectures

1. Tardos’s algorithm for min-cost flows
2. The circuit imbalance measure $\kappa_A$ and the condition measure $\bar{\kappa}_A$
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 $c: E \to \mathbb{R}$.

\[
\min \: c^T x
\]

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

\[
x \geq 0
\]

- 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 $c: E \to \mathbb{R}$.

\[
\begin{align*}
\text{min } & \quad c^T x \\
\text{s.t. } & \quad \sum_{(j,i) \in \delta^-(i)} x_{ji} - \sum_{(i,j) \in \delta^+(i)} x_{ij} = b_i \quad \forall i \in V \\
& \quad x \geq 0
\end{align*}
\]

- Dual program:

\[
\begin{align*}
\text{max } & \quad b^T \pi \\
\text{s.t. } & \quad \pi_j - \pi_i \leq c_{ij} \quad \forall ij \in E
\end{align*}
\]

- Optimality: $f_{ij} > 0 \quad \Rightarrow \quad \pi_j - \pi_i = c_{ij}$
Dual solutions: potentials

- **Dual program**: max cost feasible potential
  \[ \max b^T \pi \]
  s.t. \( \pi_j - \pi_i \leq c_{ij} \ \forall ij \in E \)

- **Residual cost**: 
  \[ c_{ij}^\pi = c_{ij} + \pi_i - \pi_j \geq 0 \]

- **Residual graph**: 
  \[ E_f = E \cup \{(j, i): f_{ij} > 0\} \]
  \[ c_{ji} = -c_{ij} \]

**LEMMA**: The primal feasible \( f \) is optimal \( \iff \exists \pi: c_{ij}^\pi \geq 0 \ \forall (i,j) \in E \) and \( c_{ij}^\pi = 0 \ \text{if} \ f_{ij} > 0 \ \iff \exists \pi: c_{ij}^\pi \geq 0 \ \forall (i,j) \in E_f \)
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,

\[
c_{ij} \tilde{\pi} > |V| \cdot \|c - \tilde{c}\|_\infty \Rightarrow f_{ij}^* = 0
\]
Circulations and cycle decompositions

- For the node-arc incidence matrix $A$, $\ker(A) \subseteq \mathbb{R}^E$ is the set of circulations:
  \[
  \text{in-flow} = \text{out-flow}
  \]

- **LEMMA:** every circulation $f \geq 0$ can be decomposed as
  \[
  f = \sum_{i} \lambda_i \chi_{C_i}, \quad \lambda_i \geq 0
  \]
  for directed cycles $C_i$
**LEMMA:** Let $f$ and $f'$ be two feasible flows for the same demand vector $b$. Then, we can write

$$f' = f + \sum_{i} \lambda_i \chi_{C_i}, \quad \lambda_i \geq 0$$

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

- 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 $f'$. 

Circulations and cycle decompositions
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,

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

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 \( c^\pi \),
  \[ (c^\pi)^\top f = c^\top f + b^\top \pi \]
- Solving the problem for \( c \) and \( c^\pi \) is equivalent.
- If \( 0 \in U \), i.e. \( \exists \pi : c^\pi \equiv 0 \), 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:
  $$\|c^\pi\|_2 \geq \|c\|_2 \ \forall \pi \in \mathbb{R}^V$$

- 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
  $$c_{ij} > |V| > |V| \cdot \|c - \tilde{c}\|_\infty$$

**LEMMA:** There exist at least one such arc.

**PROOF:**

$$\|c^{\tilde{\pi}}\|_\infty \geq \frac{\|c^{\tilde{\pi}}\|_2}{\sqrt{|E|}} \geq \frac{\|c\|_2}{\sqrt{|E|}} \geq \frac{\|c\|_\infty}{\sqrt{|E|}} = |V|$$

Also note that

$$c_{ij} \geq \tilde{c}_{ij} \geq 0$$
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 $\kappa_A$ and the condition measure $\bar{\kappa}_A$
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 $\kappa_A$ 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 $[n] = \{1, 2, \ldots, n\}$: 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

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

- $C_A$ : set of all circuits.

- The circuit imbalance measure is defined as

$$\kappa_A = \max \left\{ \frac{|g^C_j|}{|g^C_i|} : C \in C_A, i, j \in C \right\}$$
Properties of $\kappa_A$

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

- 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 $C \in C_A$, 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

$$g^C = (\det(D^{(1)}), \det(D^{(2)}), \ldots, \det(D^{(t)}))$$
Properties of $\kappa_A$

- **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: $\kappa$ has a natural integer variant that is related to Graver bases
- ...
The condition number $\tilde{\chi}_A$

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

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

- **LEMMA**
  
  i. If $A$ is an integer matrix with bit encoding length $L$, then $\tilde{\chi}_A \leq 2^{O(L)}$
  
  ii. $\tilde{\chi}_A = \max\{\|B^{-1}A\|: B \text{ nonsingular } m \times m \text{ submatrix of } A\}$
  
  iii. $\tilde{\chi}_A$ only depends on the subspace $W = \ker(A)$
  
  iv. $\tilde{\chi}_W = \tilde{\chi}_{W^\perp}$
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 
  
  $L_I^W(z) = \text{arg min}\{\|x\|_2 : x \in W, x_I = z\}$

- 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

\[ L^W_I(z) = \arg \min \{ \|x\|_2 : x \in W, x_I = z \} \]
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:**

\[ x = L_I^W(z) = g_{C_1} + g_{C_2} + g_{C_3} \]
The condition numbers $\kappa_A$ 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 $\kappa_A$ and $\bar{\chi}_A$:

LEMMA (Tunçel 1999): It is NP-hard to approximate $\bar{\chi}_A$ by a factor better than $2^{\text{poly}(\text{rank}(A))}$
Recap from Lecture 1

- **Overall goal:** solving LPs exactly and “as strongly polynomially as possible”

- One can reduce the dependence to the constraint matrix only:
  - Tardos ’86: poly($n, m, \log \Delta_A$) *black box* LP algorithm
  - Vavasis-Ye ’96 Layered-least-squares Interior Point Method
    poly($n, m, \log \bar{\chi}_A$)

- The crucial parameter of the constraint matrix is the circuit imbalance measure, a nice geometric parameter associated with the subspace ker($A$)

*Updated slides available at*
https://personal.lse.ac.uk/veghl/ipco*
Recap from Lecture 1

- Tardos’s algorithm for min. cost generalized flows: circuits, proximity, and variable fixing

- **Circuit imbalance measure**: matrix $A \in \mathbb{R}^{m \times n}$
  - Circuit: a set $C \subseteq [n]$ if $\{a_i: i \in C\}$ is a linearly dependent set minimal for containment. $\exists g^C \in \mathbb{R}^C$ unique up to a scalar multiplication:
  $$\sum_{i \in C} g^C_i a_i = 0$$

- The circuit imbalance measure is defined as
  $$\kappa_A = \max \left\{ \frac{|g^C_j|}{|g^C_i|}: C \in C_A, i, j \in C \right\}$$

- Properties: $\text{TU} \Rightarrow \kappa_A = 1$; and $\kappa_A$ can be used to bound the lifting cost
Outline of the lectures

1. Tardos’s algorithm for min-cost flows
2. The circuit imbalance measure $\kappa_A$ and the condition measure $\bar{\kappa}_A$
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^T x
Ax = b
x \geq 0

- \varepsilon\text{-approximate solution:}
  - Approximately feasible: \|Ax - b\| \leq \varepsilon(\|A\|_F R + \|b\|)
  - Approximately optimal: \( c^T x \leq \text{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

\[ \min c^T x \quad Ax = b \quad x \geq 0 \]

- \( n \) variables, \( m \) equality constraints, Randomized vs. Deterministic
- Significant recent progress:
  - \( R O \left( (\text{nnz}(A) + m^2)\sqrt{m} \log^{O(1)}(n) \log \left( \frac{n}{\varepsilon} \right) \right) \) 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 $\kappa_A$ dependence

- $n$ 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 $\| \cdot \|_\alpha, \| \cdot \|_\beta$

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

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

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

$$Ax = Ad, \quad x \geq 0.$$ 

If feasible, then there exists a feasible solution $x$ such that

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

**PROOF:**
Linear feasibility algorithm

Linear feasibility problem

\[ Ax = Ad, \quad x \geq 0. \]

- Recursive algorithm using a stronger problem formulation:

\[ Ax = Ad, \quad x \geq 0. \]
\[ \|x - d\|_\infty \leq C' \kappa_A^2 \|d^-\|_1 \]

- Variable fixing: conclude \( x_i > 0 \) and project out \( x_i \)

- Black box oracle for \( \epsilon = 1/(\text{poly}(n, \kappa_A)) \)

proximity
\[ Ax = Ad \]
\[ \|x - d\|_\infty \leq C \kappa_A \|d^-\|_1 \]

error
\[ \|x^-\|_\infty \leq \epsilon \|d^-\|_1 \]
The lifting operator

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

\[ W = \ker(A) \]

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

For every \( z \in \pi_I(W) \), \( x = L_I^W(z) \in W = \ker(A) \) s.t.

\[ x_I = z, \text{ and } \|x\|_\infty \leq \kappa_A \|z\|_1 \]
The linear feasibility algorithm

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

   $$Az = Ad$$
   $$\|z - d\|_\infty \leq C\kappa_A \|d^-\|_1$$
   $$\|z^-\|_\infty \leq \varepsilon \|d^-\|_1$$

2. Set $J = \{i \in [n]: z_i < \kappa_A \|d^-\|_1\}$; assume $J \neq [n]$.

3. Recursively obtain $\tilde{x} \in \mathbb{R}_+^J$ from $\mathcal{F}(\pi_J(\ker(A)), z_J)$

4. Return $x = z + L_J^W (\tilde{x} - z_J)$

Problem $\mathcal{F}(\ker(A), d)$

$$Ax = Ad$$
$$\|x - d\|_\infty \leq C'\kappa_A^2 \|d^-\|_1$$
$$x \geq 0$$
1. Call the black box solver to find a solution $z$ for $\epsilon = 1/(\kappa_A n)^4$

$$Az = Ad$$

$$\|z - d\|_\infty \leq C\kappa_A \|d^-\|_1$$

$$\|z^-\|_\infty \leq \epsilon \|d^-\|_1$$

2. Set $J = \{i \in [n]: z_i < \kappa_A \|d^-\|_1\}$; assume $J \neq [n]$.

3. Recursively obtain $\tilde{x} \in \mathbb{R}_+^J$ from $F(\pi_J(\ker(A)), z_J)$

4. Return $x = z + L^W_J (\tilde{x} - z_J)$ where $W = \ker(A)$
The linear feasibility algorithm

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

- If \( J = [n] \), then we replace \( d \) by its projection to \( W^\perp = \text{im}(A^T) \)
- 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.
- $\kappa_A$ is hard to approximate within $2^{O(n)}$ Tunçel 1999.
- We use an estimate $M$ in the algorithm.
- The algorithm may fail if $\|L_j^W(\tilde{x} - z_j)\|_{\infty} > M \|\tilde{x} - z_j\|_1$
- In this case, we restart with
  \[
  \max \left\{ M^2, \frac{\|L_j^W(\tilde{x} - z_j)\|_{\infty}}{\|\tilde{x} - z_j\|_1} \right\}
  \]
- Our estimate never overshoots $\kappa_A$ by much, but can be significantly better.
Proximity for optimization

\[
\begin{align*}
\min \quad & c^T x \\
\text{subject to} \quad & Ax = Ad \\
\quad & x \geq 0 \\
\max \quad & b^T y \\
\text{subject to} \quad & A^T y + s = c \\
\quad & s \geq 0
\end{align*}
\]

**THEOREM:** Let \( A^T y + s = c, s \geq 0 \) be a feasible dual solution, and assume the primal is also feasible. Then there exists a primal optimal \( Ax^* = Ad, x^* \geq 0 \) such that

\[
\|x^* - d\|_\infty \leq \kappa_A \left( \|d^-\|_1 + \|d_{\text{supp}(s)}\|_1 \right).
\]
Optimization algorithm

\[
\begin{align*}
\min & \quad c^T x \\
\text{subject to} & \quad Ax = Ad, \quad x \geq 0
\end{align*}
\]

\[
\begin{align*}
\max & \quad b^T y \\
\text{subject to} & \quad A^T y + s = c, \quad s \geq 0
\end{align*}
\]

- \( nm \) 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
  \[
  \min c^T x \quad \text{s.t.} \quad Ax = A\tilde{d}, \quad d \geq 0
  \]
- 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 $\kappa_A$ and the condition measure $\bar{\kappa}_A$
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

\[ \begin{align*}
\text{min } & c^T x \\
A & x = b \\
x & \geq 0
\end{align*} \quad \begin{align*}
\text{max } & b^T y \\
A^T & y + s = c \\
s & \geq 0
\end{align*} \]

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

\[ \begin{align*}
\text{min } & (Dc)^T x' \\
AD & x' = b \\
x' & \geq 0
\end{align*} \quad \begin{align*}
\text{max } & b^T y' \\
(AD)^T & y' + s' = Dc \\
s' & \geq 0
\end{align*} \]

Mapping between solutions:

\[ x' = D^{-1} x, \quad y' = y, \quad s' = Ds \]
Diagonal rescaling of LP

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

$$\min (Dc)^\top x' \quad \max b^\top y'$$
$$ADx' = b \quad (AD)^\top y' + s' = Dc$$
$$x' \geq 0 \quad s' \geq 0$$

Mapping between solutions:

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

- 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

\[ \begin{array}{ll}
\min c^\top x, & Ax = b \quad x \geq 0 \\
\end{array} \]

- 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: \( \text{poly}(n, m, \log \Delta_A) \) LP algorithm
- Layered-least-squares (LLS) Interior Point Method
  Vavasis-Ye ’96: \( \text{poly}(n, m, \log \bar{\chi}_A) \) LP algorithm in the real model of computation
  \( \bar{\chi}_A \): condition number
- Dadush-Huiberts-Natura-V ’20: \( \text{poly}(n, m, \log \bar{\chi}_A^*) \)
  \( \bar{\chi}_A^* \): optimized version of \( \bar{\chi}_A \)
Optimizing $\kappa_A$ and $\bar{\chi}_A$ by rescaling

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

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

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

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

**THEOREM (Dadush-Huiberts-Natura-V '20):** Given $A \in \mathbb{R}^{m \times n}$, in $O(n^2 m^2 + n^3)$ time, one can
  - approximate the value $\kappa_A$ within a factor $(\kappa_A^*)^2$, and
  - compute a rescaling $D \in \mathcal{D}$ satisfying $\kappa_{AD} \leq (\kappa_A^*)^3$.

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

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

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

- **EXAMPLE:** Let $\ker(A) = \text{span}((0,1,1,M), (1,0,M,1))$
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 $i, j \in [n]$, \[ \kappa_{ij} = \max \left\{ \frac{|g_j^C|}{|g_i^C|} : C \in \mathcal{C}_A, \text{s.t. } i, j \in C \right\} \]
- 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 $\{1, 2, \ldots, n\}$

$$(k_A^*)^{|H|} \geq \prod_{(i,j) \in H} k_{i,j}$$
Circuit imbalance min-max formula

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

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

PROOF:
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 \( \kappa_{ij} \) values is NP-complete...
- **LEMMA:** For any circuit \( C \in C_A \) s.t. \( i, j \in C \),

\[
\frac{|g_j^C|}{|g_i^C|} \geq \frac{\kappa_{ij}}{\left(\kappa_W^*\right)^2}
\]
Outline of the lectures

1. Tardos’s algorithm for min-cost flows
2. The circuit imbalance measure $\kappa_A$ and the condition measure $\bar{\kappa}_A$
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

- $A \in \mathbb{R}^{m \times n}, c, d \in \mathbb{R}^m$

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = Ad \\
\quad & \quad x \geq 0
\end{align*}
\]

\[
\begin{align*}
\max & \quad b^T y \\
\text{s.t.} & \quad A^T y + s = c \\
\quad & \quad s \geq 0
\end{align*}
\]

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

- Optimality gap:

\[
c^T x - b^T y = x^T s.
\]
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 $\mu$.
- 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 $s(\mu)^T x(\mu) = n\mu$.
- Interior point algorithms: walk down along the central path with $\mu$ 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 \(\mu\) by a factor at least \(1 - \beta/n\). May move slightly 'farther' from the central path.
  - **Corrector steps**: do not change parameter \(\mu\), but move back 'closer' to the central path.

Within \(O(n)\) iterations, \(\mu\) decreases by a factor 2.
The predictor step

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

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

- 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

*least squares view*

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

\[
\begin{align*}
A \Delta x &= 0 \\
A^T \Delta y + \Delta s &= 0 \\
s_i \Delta x_i + x_i \Delta s_i &= -x_i s_i \quad \forall i \in [n]
\end{align*}
\]

\[
\Delta x = \arg \min \sum_{i=1}^{n} \left( \frac{x_i + \Delta x_i}{x_i} \right)^2 \quad \text{s. t. } A \Delta x = 0
\]

\[
\Delta s = \arg \min \sum_{i=1}^{n} \left( \frac{s_i + \Delta s_i}{s_i} \right)^2 \quad \text{s. t. } A^T \Delta y + \Delta s = 0
\]
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 $\kappa_A$ and the condition measure $\bar{\kappa}_A$
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. $(x')^T s' \leq x^T s$, we have

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

*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:

$$\Delta x = \arg \min \sum_{i=1}^{n} \left( \frac{x_i + \Delta x_i}{x_i} \right)^2 \text{ s.t. } A\Delta x = 0$$

$$\Delta s = \arg \min \sum_{i=1}^{n} \left( \frac{s_i + \Delta s_i}{s_i} \right)^2 \text{ s.t. } A^T\Delta y + \Delta s = 0$$

and bounds on the stepsize.
Variable fixing...—or not?

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

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

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

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 $\kappa_A$

- 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 $L^W_I : \mathbb{R}^I \to \mathbb{R}^n$ is defined as
  \[ L^W_I(z) = \text{arg min}\{\|x\|_2 : x \in W, x_I = z\} \]

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

- For every $z \in \pi_I(W)$, $x = L^W_I(z) \in W = \ker(A)$ s.t.
  \[ x_I = z, \text{ and } \|x\|_\infty \leq \kappa_A \|z\|_1 \]
Motivating the layering idea:
final rounding step in standard IPM

\[
\begin{align*}
\min & \ c^T x \\
\text{subject to} & \ Ax = b \\
& \ x \geq 0 \\
\max & \ b^T y \\
\text{subject to} & \ A^T y + s = c \\
& \ s \geq 0
\end{align*}
\]

- Limit optimal solution \((x^*, y^*, s^*)\), and optimal partition \([n] = B \cup N\) s.t. \(B = \text{supp}(x^*), \ N = \text{supp}(s^*)\).

- Given \((x, y, s)\) near central path with ‘small enough’ \(\mu = s^T x/n\) such that for every \(i \in [n]\), either \(x_i\) or \(s_i\) very small.

- Assume that we can correctly guess \(B = \{i: x_i > M\sqrt{\mu}\}, \ N = \{i: s_i > M\sqrt{\mu}\}\)
Assume we have a partition $B, N$, we have
$$i \in B: x_i > M\sqrt{\mu}, \quad s_i < \sqrt{\mu}/M$$
$$i \in N: x_i < \sqrt{\mu}/M, \quad s_i > M\sqrt{\mu}$$

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

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

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

s. t. $A\Delta x = 0$

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

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

s. t. $A\Delta x = 0$

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

s. t. $A(\Delta x_B, \Delta x_N) = 0$
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 $x_i > O(n^c) \tilde{c} \chi A x_{i+1}$
- 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!
DEFINITION: The variables $x_i$ and $x_j$ cross over between $\mu$ and $\mu'$, $\mu > \mu'$, 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

\[
(i, j): \quad \frac{\kappa_{ij} x_i}{x_j} > \frac{1}{\text{poly}(n)}
\]

The \( \kappa_{ij} \) 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 \( \mu \) and \( \mu' \), \( \mu > \mu' \), 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

Application deadline: 5 June