A strongly polynomial algorithm for linear programs with at most two non-zero entries per row or column

Daniel Dadush

CWI Utrecht University

STACS 2025

Joint work with

Zhuan Khye Koh — Boston University

Bento Natura — Columbia

Neil Olver — The London School of Economics and Political Science

László A. Végh — Universität Bonn

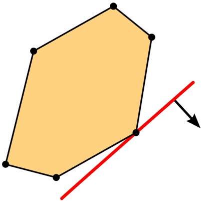
Talk Overview

1. Linear Program (LP)
   - Polynomial vs Strongly Polynomial Algorithms
2. LPs with ≤ 2 variables per Inequality
3. Minimum Cost Generalized Flow
4. A Strongly Polynomial Interior Point Method

Linear Program (LP)

Primal:

$$\min \; c^\top x$$
$$\text{s.t.} \; Ax = b$$
$$x \geq \mathbf{0}$$



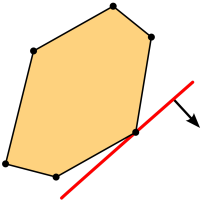
Dual:

$$\max \; b^\top y$$
$$\text{s.t.} \; A^\top y \leq c$$

Linear Program (LP)

Primal:

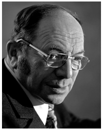
$$\min \; c^\top x \quad \text{s.t. } Ax = b, \; x \geq \mathbf{0}$$

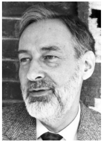


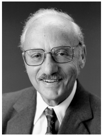
Dual:

$$\max \; b^\top y \quad \text{s.t. } A^\top y \leq c$$

- Introduced by [Kantorovich '39] [Hitchcock '41] [Koopmans '42] [Dantzig '47].







LP Algorithms

Input: $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L.

LP Algorithms

Input: $A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L.

Def: A polynomial algorithm runs in $\mathrm{poly}(m, n, L)$ time.

LP Algorithms

Input: $A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L.

Def: A polynomial algorithm runs in $\mathrm{poly}(m, n, L)$ time.

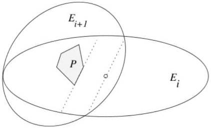
- Polynomial algorithms for LP:

LP Algorithms

Input: $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L.

Def: A polynomial algorithm runs in $\mathrm{poly}(m, n, L)$ time.

- Polynomial algorithms for LP:
  - Ellipsoid method [Khachiyan '79]

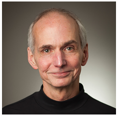


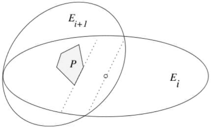
LP Algorithms

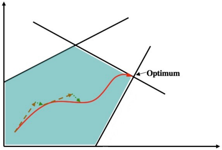
Input: $A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^n, c \in \mathbb{R}^m$. Total bit length L.

Def: A polynomial algorithm runs in $\mathrm{poly}(m, n, L)$ time.

- Polynomial algorithms for LP:
  - Ellipsoid method [Khachiyan '79]
  - Interior point method [Karmarkar '84] [Renegar '88]





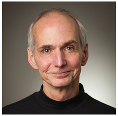


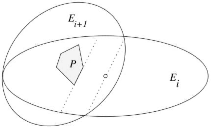
LP Algorithms

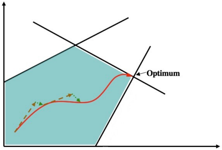
Input: $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L.

Def: A polynomial algorithm runs in $\mathrm{poly}(m, n, L)$ time.

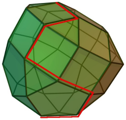
- Polynomial algorithms for LP:
  - Ellipsoid method [Khachiyan '79]
  - Interior point method [Karmarkar '84] [Renegar '88]







- Simplex method [Dantzig '47]

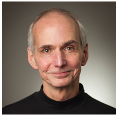


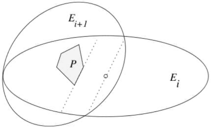
LP Algorithms

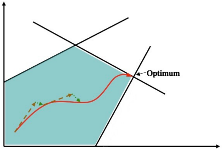
Input: $A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^n, c \in \mathbb{R}^m$. Total bit length L.

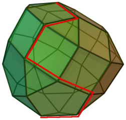
Def: A polynomial algorithm runs in $\mathrm{poly}(m, n, L)$ time.

- Polynomial algorithms for LP:
  - Ellipsoid method [Khachiyan '79]
  - Interior point method [Karmarkar '84] [Renegar '88]









- Simplex method [Dantzig '47]
  - Not known to be polynomial, but efficient in practice.

Strongly Polynomial

Input: $A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L.

Strongly Polynomial

Input: $A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L.

Def: An algorithm is strongly polynomial if it uses

1. $\mathrm{poly}(m, n)$ elementary arithmetic operations $(+, -, \times, \div, <?)$, and

Strongly Polynomial

Input: $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L.

Def: An algorithm is strongly polynomial if it uses

1. $\mathrm{poly}(m, n)$ elementary arithmetic operations $(+, -, \times, \div, <?)$, and
2. $\mathrm{poly}(m, n, L)$ space.

Strongly Polynomial

Input: $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L.

Def: An algorithm is strongly polynomial if it uses

1. $\mathrm{poly}(m, n)$ elementary arithmetic operations $(+, -, \times, \div, <?)$, and
2. $\mathrm{poly}(m, n, L)$ space.

Smale's 9th Problem [Megiddo '83]

Is there a strongly polynomial algorithm for linear programming?

The Zoo of LP Subclasses

General LP $\equiv$ LP with ≤ 3 variables per inequality

Strongly polynomial (as of 2023)

Combinatorial LP:

- Shortest path
- Bipartite matching
- Maximum flow
- Minimum cost flow

- LP feasibility with ≤ 2 variables per inequality
- Discounted MDP
- Maximum generalized flow

- Undiscounted MDP
- LP with ≤ 2 variables per inequality

The Zoo of LP Subclasses

General LP $\equiv$ LP with ≤ 3 variables per inequality

Strongly polynomial (as of 2024)

Combinatorial LP:

- Shortest path
- Bipartite matching
- Maximum flow
- Minimum cost flow

- LP feasibility with ≤ 2 variables per inequality
- Discounted MDP
- Maximum generalized flow
- LP with ≤ 2 variables per inequality

- Undiscounted MDP

2 Variables-per-Inequality LP

• [Hochbaum '04] Any 2-variables-per-inequality (2VPI) LP can be reduced to the following monotone form:

$$\begin{aligned} &\max \ b^\top y \\ &\text{s.t.} \ \gamma_e y_j - y_i \le c_e \qquad \forall e = (i,j), \end{aligned}$$

where the edges come from a directed multigraph $G = (V, E)$, and $\gamma_e > 0$ is the *gain factor* of the edge e.

Minimum Cost Generalized Flow

- The dual LP of a monotone 2VPI system is:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & \sum_{e \in \delta^{\text{in}}(v)} \gamma_e x_e - \sum_{e \in \delta^{\text{out}}(v)} x_e = b_v \quad \forall v \in V \\ & x \geq \mathbf{0} \end{aligned}$$

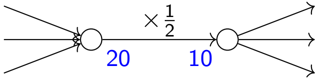
Interpretation: for directed multigraph $G = (V, E)$, $|V| = n$,$|E| = m$, node demands $b \in \mathbb{R}^V$, arc costs $c \in \mathbb{R}^E$ and gain factors $\gamma \in \mathbb{R}^E_{>0}$, find a **minimum cost generalized flow** satisfying all node demands.

Minimum Cost Generalized Flow

- The dual LP of a monotone 2VPI system is:

$$\begin{aligned} \min\ & c^\top x \\ \text{s.t.}\ & \sum_{e\in\delta^{\text{in}}(v)} \gamma_e x_e - \sum_{e\in\delta^{\text{out}}(v)} x_e = b_v \quad \forall v \in V \\ & x \geq \mathbf{0} \end{aligned}$$

Interpretation: for directed multigraph $G = (V, E)$, $|V| = n$,$|E| = m$, node demands $b \in \mathbb{R}^V$, arc costs $c \in \mathbb{R}^E$ and gain factors $\gamma \in \mathbb{R}^E_{>0}$, find a **minimum cost generalized flow** satisfying all node demands.



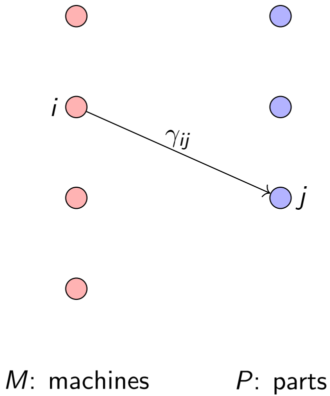
Models leaky pipes, currency exchange etc.

Example: Production with Different Machines

• Variant of a problem proposed by Kantorovich in his 1939 paper introducing Linear Programming.

Example: Production with Different Machines

• Variant of a problem proposed by Kantorovich in his 1939 paper introducing Linear Programming.

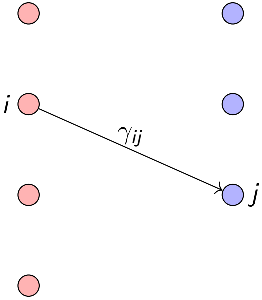


• Machine i can produce γ_{ij} units of part j in one day at cost c_{ij}.

• Daily demand d_j for part j.

Example: Production with Different Machines

• Variant of a problem proposed by Kantorovich in his 1939 paper introducing Linear Programming.



M: machines P: parts

• Machine i can produce γ_{ij} units of part j in one day at cost c_{ij}.

• Daily demand d_j for part j.

$$
\begin{aligned}
\min \quad & \sum_{i \in M, j \in P} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j \in P} x_{ij} \leq 1 \qquad \forall i \in M \\
& \sum_{i \in M} \gamma_{ij} x_{ij} \geq d_j \quad \forall j \in P \\
& x \geq \mathbf{0}
\end{aligned}
$$

Prior Work

Prior Work

- Algorithms for for two-variable-per-inequality feasibility:
  - Polynomial [Aspvall, Shiloach '80]
  - Strongly polynomial [Megiddo '83] [Cohen, Megiddo '94] [Hochbaum, Naor '94] [Karczmarz '22]

Prior Work

- Algorithms for for two-variable-per-inequality feasibility:
  - Polynomial [Aspvall, Shiloach '80]
  - Strongly polynomial [Megiddo '83] [Cohen, Megiddo '94] [Hochbaum, Naor '94] [Karczmarz '22]
- Algorithms for generalized flow feasibility:
  - Polynomial [Goldberg, Plotkin, Tardos '91]
  - Strongly polynomial [Végh '13] [Olver, Végh '20]

Prior Work

- Algorithms for for two-variable-per-inequality feasibility:
  - Polynomial [Aspvall, Shiloach '80]
  - Strongly polynomial [Megiddo '83] [Cohen, Megiddo '94] [Hochbaum, Naor '94] [Karczmarz '22]
- Algorithms for generalized flow feasibility:
  - Polynomial [Goldberg, Plotkin, Tardos '91]
  - Strongly polynomial [Végh '13] [Olver, Végh '20]
- Algorithms for minimum cost generalized flow:
  - Polynomial [Wayne '02]

Main Result

Theorem [D, Koh, Natura, Olver, Végh '24]

There is a strongly polynomial algorithm for the minimum cost generalized flow problem, and consequently, for LPs with at most 2 variables per inequality or 2 variables per column.

Main Result

Theorem [D, Koh, Natura, Olver, Végh '24]

There is a strongly polynomial algorithm for the minimum cost generalized flow problem, and consequently, for LPs with at most 2 variables per inequality or 2 variables per column.

- The algorithm is based on the interior point method by [Allamigeon, D, Loho, Natura, Végh '22].

Main Result

Theorem [D, Koh, Natura, Olver, Végh '24]

There is a strongly polynomial algorithm for the minimum cost generalized flow problem, and consequently, for LPs with at most 2 variables per inequality or 2 variables per column.

- The algorithm is based on the interior point method by [Allamigeon, D, Loho, Natura, Végh '22].
- What we'll need for this talk:
  1. Interior point method
  2. Straight line complexity

Central Path

Central Path

- For each $\mu > 0$, there exists a unique optimal solution $x^{\mathrm{cp}}(\mu)$ to

$$\begin{aligned} \min \ & c^\top x - \mu \sum_{i=1}^{n} \log(x_i) \\ \text{s.t.} \ & Ax = b, x \in \mathbb{R}^m_{\geq 0}. \end{aligned}$$

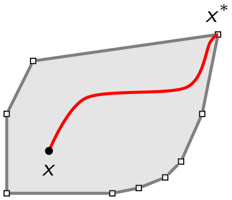
Central Path

- For each $\mu > 0$, there exists a unique optimal solution $x^{\mathrm{cp}}(\mu)$ to

$$\begin{aligned} \min \ & c^\top x - \mu \sum_{i=1}^{n} \log(x_i) \\ \text{s. t.} \ & Ax = b, x \in \mathbb{R}^m_{\geq 0}. \end{aligned}$$

Def: The central path is the curve

$$\{x^{\mathrm{cp}}(\mu) : \mu > 0\}.$$



Central Path

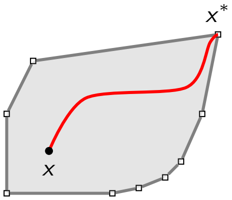
• For each $\mu > 0$, there exists a unique optimal solution $x^{\mathrm{cp}}(\mu)$ to

$$\begin{aligned} &\min \; c^\top x - \mu \sum_{i=1}^{n} \log(x_i) \\ &\text{s. t. } \; Ax = b, x \in \mathbb{R}^m_{\geq 0}. \end{aligned}$$

Def: The central path is the curve

$$\{x^{\mathrm{cp}}(\mu) : \mu > 0\}.$$

• As $\mu \to 0$, $x^{\mathrm{cp}}(\mu)$ converges to an optimal solution x^* of the LP.



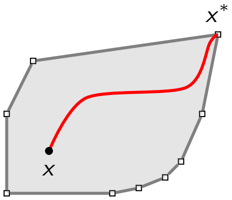
Central Path

• For each $\mu > 0$, there exists a unique optimal solution $x^{\mathrm{cp}}(\mu)$ to

$$\begin{aligned} &\min\ c^\top x - \mu \sum_{i=1}^{n} \log(x_i) \\ &\text{s. t.}\ \ Ax = b, x \in \mathbb{R}^m_{\geq 0}. \end{aligned}$$

Def: The central path is the curve

$$\{x^{\mathrm{cp}}(\mu) : \mu > 0\}.$$



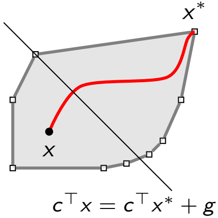
• As $\mu \to 0$, $x^{\mathrm{cp}}(\mu)$ converges to an optimal solution x^* of the LP.

• Interior Point Method (IPM): Walk down the central path with geometrically decreasing μ.

Alternate View of the Central Path

- Let us reparameterize x^{cp} by the optimality gap:

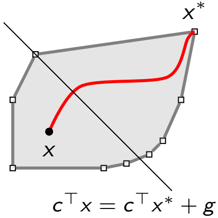
$$c^\top x^{\mathrm{cp}}(g) = c^\top x^* + g \qquad \forall g \geq 0.$$



Alternate View of the Central Path

- Let us reparameterize x^{cp} by the optimality gap:

$$c^\top x^{\mathrm{cp}}(g) = c^\top x^* + g \qquad \forall g \geq 0.$$

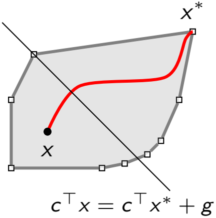


Intuition: Each coordinate of $x^{\mathrm{cp}}(g)$ approximately as big as possible subject to gap bound g.

Alternate View of the Central Path

- Let us reparameterize x^{cp} by the optimality gap:

$$c^\top x^{\text{cp}}(g) = c^\top x^* + g \qquad \forall g \geq 0.$$



Intuition: Each coordinate of $x^{\text{cp}}(g)$ approximately as big as possible subject to gap bound g.

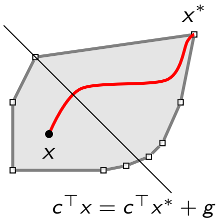
Def: The max central path is the curve $\{x^{\text{mcp}}(g) : g \geq 0\}$, given by

$$\begin{aligned} x_i^{\text{mcp}}(g) := \max\ & x_i \\ \text{s.t.}\ & x \text{ feasible} \\ & \text{optimality gap} \leq g. \end{aligned}$$

Alternate View of the Central Path

- Let us reparameterize x^{cp} by the optimality gap:

$$c^\top x^{\mathrm{cp}}(g) = c^\top x^* + g \qquad \forall g \geq 0.$$



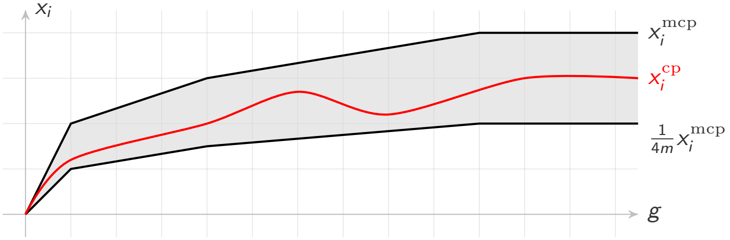
Intuition: Each coordinate of $x^{\mathrm{cp}}(g)$ approximately as big as possible subject to gap bound g.

Def: The max central path is the curve $\{x^{\mathrm{mcp}}(g) : g \geq 0\}$, given by

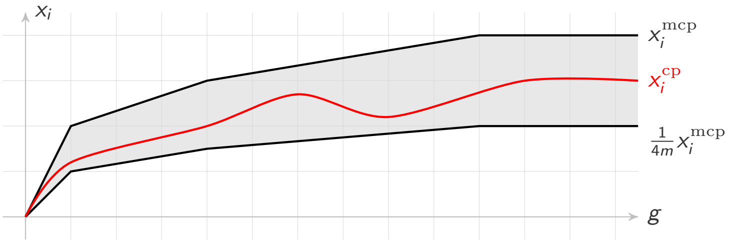
$$\begin{aligned} x_i^{\mathrm{mcp}}(g) := \max\ & x_i \\ \text{s.t. }\ & x \text{ feasible} \\ & \text{optimality gap} \leq g. \end{aligned}$$

Thm: $\frac{1}{2m} x^{\mathrm{mcp}} \leq x^{\mathrm{cp}} \leq x^{\mathrm{mcp}}$.

Straight Line Complexity



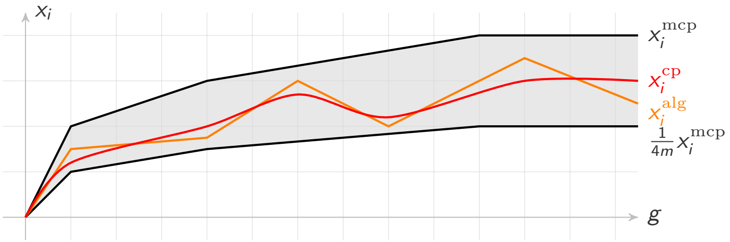
Straight Line Complexity



- IPM generates a piecewise-affine curve x^{alg} near the central path

$$\frac{1}{2} x^{\mathrm{cp}} \leq x^{\mathrm{alg}} \leq 2 x^{\mathrm{cp}} \Rightarrow \frac{1}{4m} x^{\mathrm{mcp}} \leq x^{\mathrm{alg}} \leq x^{\mathrm{mcp}}.$$

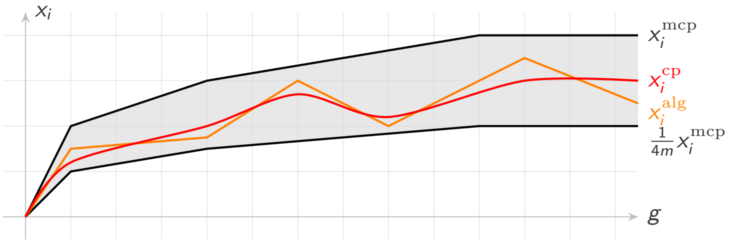
Straight Line Complexity



- IPM generates a piecewise-affine curve x^{alg} near the central path

$$\frac{1}{2}x^{\mathrm{cp}} \leq x^{\mathrm{alg}} \leq 2x^{\mathrm{cp}} \Rightarrow \frac{1}{4m}x^{\mathrm{mcp}} \leq x^{\mathrm{alg}} \leq x^{\mathrm{mcp}}.$$

Straight Line Complexity



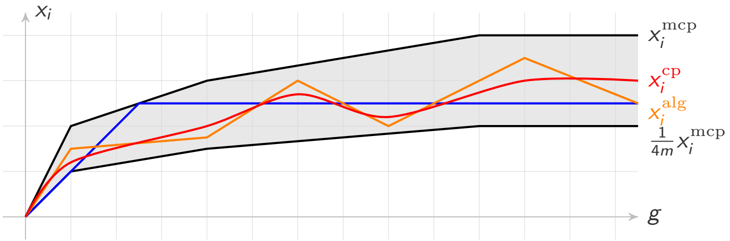
- IPM generates a piecewise-affine curve x^{alg} near the central path

$$\frac{1}{2} x^{\mathrm{cp}} \leq x^{\mathrm{alg}} \leq 2x^{\mathrm{cp}} \Rightarrow \frac{1}{4m} x^{\mathrm{mcp}} \leq x^{\mathrm{alg}} \leq x^{\mathrm{mcp}}.$$

Def: The straight line complexity of x_i^{mcp}, $\mathrm{SLC}_\theta(x_i^{\mathrm{mcp}})$, is the minimum number of pieces of a continuous piecewise-affine function h such that

$$\theta x_i^{\mathrm{mcp}} \leq h \leq x_i^{\mathrm{mcp}}.$$

Straight Line Complexity



- IPM generates a piecewise-affine curve x^{alg} near the central path

$$\frac{1}{2}x^{\mathrm{cp}} \leq x^{\mathrm{alg}} \leq 2x^{\mathrm{cp}} \Rightarrow \frac{1}{4m}x^{\mathrm{mcp}} \leq x^{\mathrm{alg}} \leq x^{\mathrm{mcp}}.$$

Def: The straight line complexity of x_i^{mcp}, $\mathrm{SLC}_\theta(x_i^{\mathrm{mcp}})$, is the minimum number of pieces of a continuous piecewise-affine function h such that

$$\theta x_i^{\mathrm{mcp}} \leq h \leq x_i^{\mathrm{mcp}}.$$

Straight Line Complexity

- # iterations required by any IPM is at least

$$\max_{i\in[m]} \mathsf{SLC}_{\frac{1}{4m}}(x_i^{\mathrm{mcp}}).$$

Straight Line Complexity

- # iterations required by any IPM is at least

$$\max_{i\in[m]} \mathrm{SLC}_{\frac{1}{4m}}(x_i^{\mathrm{mcp}}).$$

Theorem [Allamigeon, D, Loho, Natura, Végh '22]

There is an interior point method which solves LP in

$$O\left(\min_{\theta\in(0,1]} \sqrt{m}\log\left(\frac{m}{\theta}\right)\sum_{i=1}^{m} \mathrm{SLC}_\theta(x_i^{\mathrm{mcp}})\right)$$

iterations.

Main Result

Theorem [D, Koh, Natura, Olver, Végh '23]

For the minimum-cost generalized flow problem on $G = (V, E)$ with n nodes and m arcs,

$$\mathsf{SLC}_{\frac{1}{m}}(x_e^{\mathrm{mcp}}) = O(mn) \qquad\qquad \forall e \in E.$$

Main Result

Theorem [D, Koh, Natura, Olver, Végh '23]

For the minimum-cost generalized flow problem on $G = (V, E)$ with n nodes and m arcs,

$$\mathsf{SLC}_{\frac{1}{m}}(x_e^{\mathrm{mcp}}) = O(mn) \qquad \forall e \in E.$$

Theorem [D, Koh, Natura, Olver, Végh '24]

There is a strongly polynomial algorithm for the minimum cost generalized flow problem.

Main Result

Theorem [D, Koh, Natura, Olver, Végh '23]

For the minimum-cost generalized flow problem on $G = (V, E)$ with n nodes and m arcs,

$$\mathrm{SLC}_{\frac{1}{m}}(x_e^{\mathrm{mcp}}) = O(mn) \qquad \forall e \in E.$$

- Key ingredient: **Circuits**

Theorem [D, Koh, Natura, Olver, Végh '24]

There is a strongly polynomial algorithm for the minimum cost generalized flow problem.

Circuits

Def: Let $W = \ker(A)$. A circuit is any vector $f \in W \setminus \{\mathbf{0}\}$ such that $\nexists h \in W \setminus \{\mathbf{0}\}$ with $\mathrm{supp}(h) \subsetneq \mathrm{supp}(f)$.

Circuits

Def: Let $W = \ker(A)$. A circuit is any vector $f \in W \setminus \{\mathbf{0}\}$ such that $\nexists h \in W \setminus \{\mathbf{0}\}$ with $\mathrm{supp}(h) \subsetneq \mathrm{supp}(f)$.

Example: Network flow movement subspace is

$$Ax = 0 \qquad \iff \qquad \sum_{e \in \delta^{\mathrm{in}}(v)} x_e - \sum_{e \in \delta^{\mathrm{out}}(v)} x_e = 0 \quad \forall v \in V$$

Circuits

Def: Let $W = \ker(A)$. A circuit is any vector $f \in W \setminus \{\mathbf{0}\}$ such that $\nexists h \in W \setminus \{\mathbf{0}\}$ with $\text{supp}(h) \subsetneq \text{supp}(f)$.

Example: Network flow movement subspace is

$$Ax = 0 \qquad \iff \qquad \sum_{e \in \delta^{\text{in}}(v)} x_e - \sum_{e \in \delta^{\text{out}}(v)} x_e = 0 \quad \forall v \in V$$

- $\ker(A)$ = set of circulations.

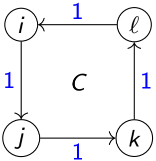
Circuits

Def: Let $W = \ker(A)$. A circuit is any vector $f \in W \setminus \{\mathbf{0}\}$ such that $\nexists h \in W \setminus \{\mathbf{0}\}$ with $\text{supp}(h) \subsetneq \text{supp}(f)$.

Example: Network flow movement subspace is

$$Ax = 0 \qquad \Longleftrightarrow \qquad \sum_{e \in \delta^{\text{in}}(v)} x_e - \sum_{e \in \delta^{\text{out}}(v)} x_e = 0 \quad \forall v \in V$$

- $\ker(A)$ = set of circulations. Circuits correspond to directed cycles.



	1	1	1	1
i	-1			1
j	1	-1		
k		1	-1	
ℓ			1	-1

$f^C = \mathbf{0}$

Circuits of Generalized Flow

Circuits of Generalized Flow

- Generalized flow movement subspace is

$$Ax = 0 \qquad \iff \qquad \sum_{e \in \delta^{\mathrm{in}}(v)} \gamma_e x_e - \sum_{e \in \delta^{\mathrm{out}}(v)} x_e = 0 \quad \forall v \in V$$

Circuits of Generalized Flow

- Generalized flow movement subspace is

$$Ax = 0 \qquad \Longleftrightarrow \qquad \sum_{e \in \delta^{\mathrm{in}}(v)} \gamma_e x_e - \sum_{e \in \delta^{\mathrm{out}}(v)} x_e = 0 \quad \forall v \in V$$

- $\ker(A)$ = set of generalized circulations.

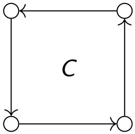
Circuits of Generalized Flow

- Generalized flow movement subspace is

$$Ax = 0 \quad \iff \quad \sum_{e \in \delta^{\text{in}}(v)} \gamma_e x_e - \sum_{e \in \delta^{\text{out}}(v)} x_e = 0 \quad \forall v \in V$$

- $\ker(A)$ = set of generalized circulations. 2 types of circuits:

Conservative cycle



$\prod_{e \in C} \gamma_e = 1$

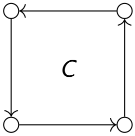
Circuits of Generalized Flow

- Generalized flow movement subspace is

$$Ax = 0 \quad \iff \quad \sum_{e \in \delta^{\text{in}}(v)} \gamma_e x_e - \sum_{e \in \delta^{\text{out}}(v)} x_e = 0 \quad \forall v \in V$$

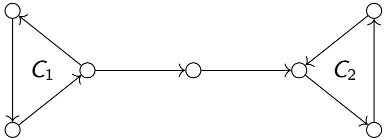
- $\ker(A)$ = set of generalized circulations. 2 types of circuits:

Conservative cycle



$\prod_{e \in C} \gamma_e = 1$

Bicycle



$\prod_{e \in C_1} \gamma_e > 1$

flow-generating

$\prod_{e \in C_2} \gamma_e < 1$

flow-absorbing

Upper Bounding the SLC

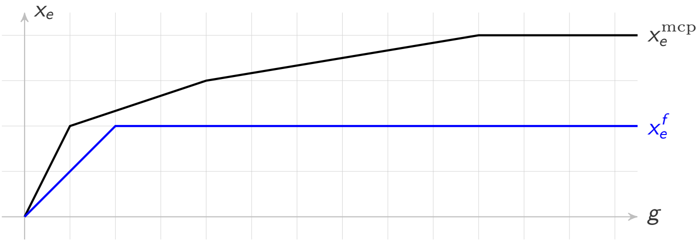
- $f \in \ker(A)$ with $c^\top f > 0$ induces a line segment in the feasible region:

$$x^f(g) := x^* + \frac{g}{c^\top f} f \qquad \text{as long as } x^* + \frac{g}{c^\top f} f \geq 0$$

Upper Bounding the SLC

- $f \in \ker(A)$ with $c^\top f > 0$ induces a line segment in the feasible region:

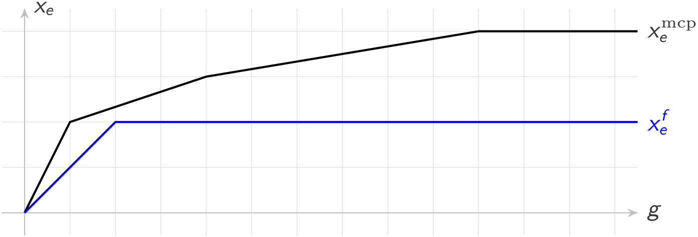
$$x^f(g) := x^* + \frac{g}{c^\top f} f \qquad \text{as long as } x^* + \frac{g}{c^\top f} f \geq 0$$



Upper Bounding the SLC

- $f \in \ker(A)$ with $c^\top f > 0$ induces a line segment in the feasible region:

$$x^f(g) := x^* + \frac{g}{c^\top f} f \qquad \text{as long as } x^* + \frac{g}{c^\top f} f \geq 0$$

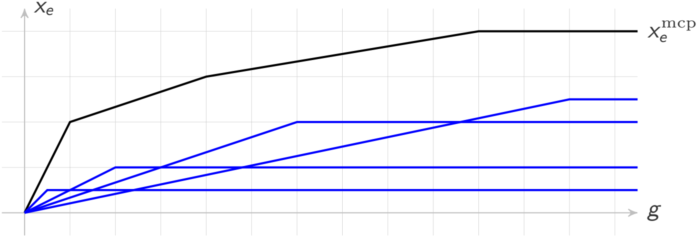


Fact: $\max\limits_{\text{circuit } C} x_e^{f^C} \geq \dfrac{x_e^{\mathrm{mcp}}}{m}$.

Upper Bounding the SLC

- $f \in \ker(A)$ with $c^\top f > 0$ induces a line segment in the feasible region:

$$x^f(g) := x^* + \frac{g}{c^\top f} f \qquad \text{as long as } x^* + \frac{g}{c^\top f} f \geq 0$$

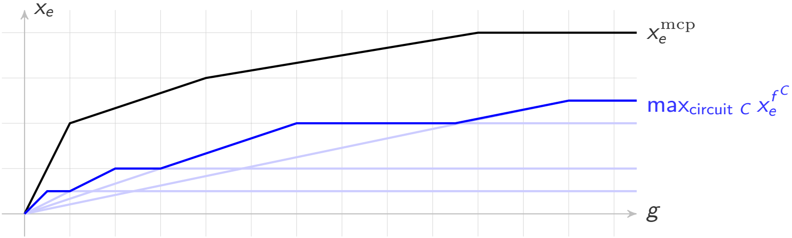


Fact: $\max_{\text{circuit } C} x_e^{f^C} \geq \frac{x_e^{\mathrm{mcp}}}{m}$.

Upper Bounding the SLC

- $f \in \ker(A)$ with $c^\top f > 0$ induces a line segment in the feasible region:

$$x^f(g) := x^* + \frac{g}{c^\top f} f \qquad \text{as long as } x^* + \frac{g}{c^\top f} f \geq 0$$

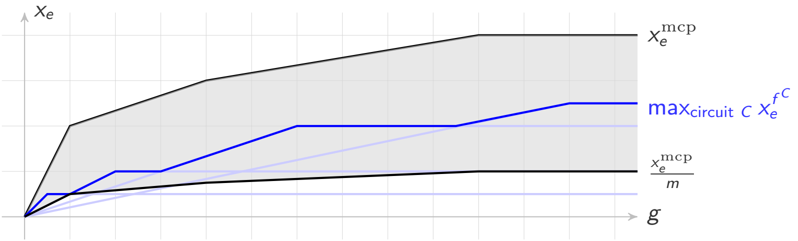


Fact: $\max_{\text{circuit } C} x_e^{f^C} \geq \dfrac{x_e^{\mathrm{mcp}}}{m}$.

Upper Bounding the SLC

- $f \in \ker(A)$ with $c^\top f > 0$ induces a line segment in the feasible region:

$$x^f(g) := x^* + \frac{g}{c^\top f} f \qquad \text{as long as } x^* + \frac{g}{c^\top f} f \geq 0$$

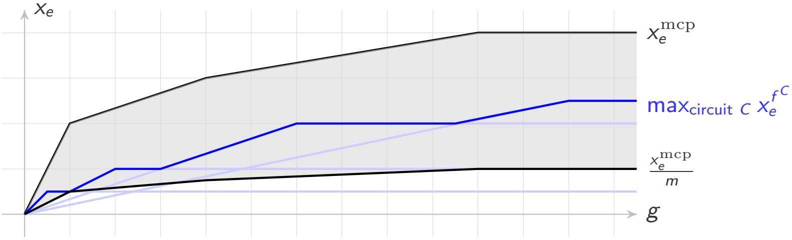


Fact: $\max\limits_{\text{circuit } C} x_e^{f^C} \geq \dfrac{x_e^{\mathrm{mcp}}}{m}$.

Upper Bounding the SLC

- $f \in \ker(A)$ with $c^\top f > 0$ induces a line segment in the feasible region:

$$x^f(g) := x^* + \frac{g}{c^\top f} f \qquad \text{as long as } x^* + \frac{g}{c^\top f} f \geq 0$$



Fact: $\max\limits_{\text{circuit } C} x_e^{f^C} \geq \dfrac{x_e^{\text{mcp}}}{m}$.

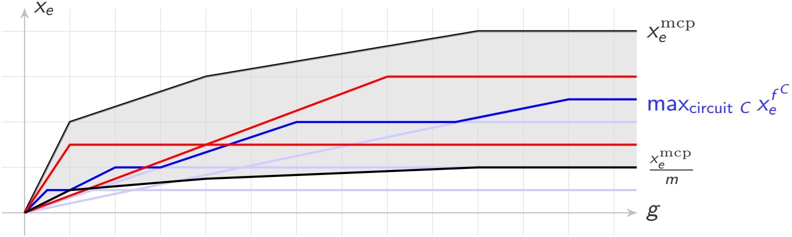
Strategy: Find $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

$$\implies \text{SLC}_{\frac{1}{m^3}}(x_e^{\text{mcp}}) \leq 2|S|.$$

Upper Bounding the SLC

- $f \in \ker(A)$ with $c^\top f > 0$ induces a line segment in the feasible region:

$$x^f(g) := x^* + \frac{g}{c^\top f} f \qquad \text{as long as } x^* + \frac{g}{c^\top f} f \geq 0$$



Fact: $\max\limits_{\text{circuit } C} x_e^{f^C} \geq \dfrac{x_e^{\mathrm{mcp}}}{m}$.

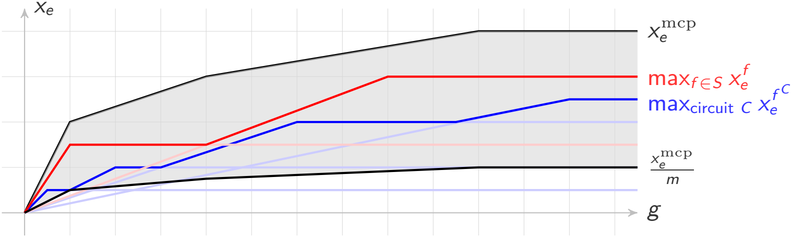
Strategy: Find $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

$$\implies \mathrm{SLC}_{\frac{1}{m^3}}(x_e^{\mathrm{mcp}}) \leq 2|S|.$$

Upper Bounding the SLC

- $f \in \ker(A)$ with $c^\top f > 0$ induces a line segment in the feasible region:

$$x^f(g) := x^* + \frac{g}{c^\top f} f \qquad \text{as long as } x^* + \frac{g}{c^\top f} f \geq 0$$



Fact: $\displaystyle\max_{\text{circuit } C} x_e^{f^C} \geq \frac{x_e^{\mathrm{mcp}}}{m}$.

Strategy: Find $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

$$\implies \mathrm{SLC}_{\frac{1}{m^3}}(x_e^{\mathrm{mcp}}) \leq 2|S|.$$

Dominating Circuits

Goal: Find small $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

Dominating Circuits

Goal: Find small $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

Recall: Circuits of generalized flow are conservative cycles and bicycles.

Dominating Circuits

Goal: Find small $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

Recall: Circuits of generalized flow are conservative cycles and bicycles.

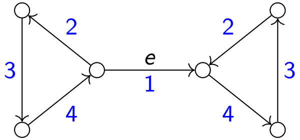
- Let C is a bicycle containing e.

Dominating Circuits

Goal: Find small $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

Recall: Circuits of generalized flow are conservative cycles and bicycles.

- Let C is a bicycle containing e.

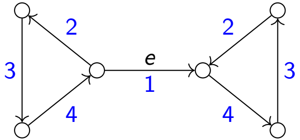


Dominating Circuits

Goal: Find small $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

Recall: Circuits of generalized flow are conservative cycles and bicycles.

- Let C is a bicycle containing e.



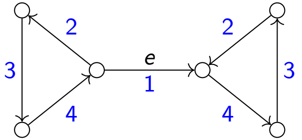
- Given $f \in \ker(A)$, what does $x_e^f \geq x_e^{f^C}$ mean?

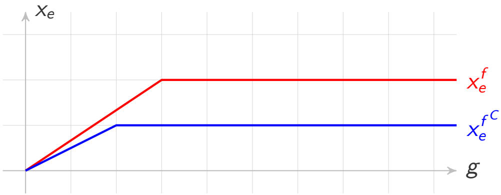
Dominating Circuits

Goal: Find small $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

Recall: Circuits of generalized flow are conservative cycles and bicycles.

- Let C is a bicycle containing e.





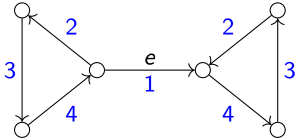
- Given $f \in \ker(A)$, what does $x_e^f \geq x_e^{f^C}$ mean?

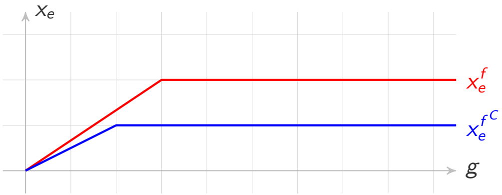
Dominating Circuits

Goal: Find small $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \frac{1}{m^2} \max_{\text{circuit } C} x_e^{f^C}$.

Recall: Circuits of generalized flow are conservative cycles and bicycles.

- Let C is a bicycle containing e.





- Given $f \in \ker(A)$, what does $x_e^f \geq x_e^{f^C}$ mean?
  - It is *cheaper* to send flow on e via f than f^C, and *more flow* can be sent on e via f than f^C.

Walk and Path Flows

Def: For s-t walk $W = (e_1, e_2, \ldots, e_k)$, the walk flow f^W sending 1 unit of flow into t is defined by

$$\gamma_{e_k} f^W_{e_k} = 1, \quad \gamma_{e_i} f^W_{e_i} = f^W_{e_{i+1}}, i \in [k-1].$$

Walk and Path Flows

Def: For s-t walk $W = (e_1, e_2, \ldots, e_k)$, the walk flow f^W sending 1 unit of flow into t is defined by

$$\gamma_{e_k} f^W_{e_k} = 1, \quad \gamma_{e_i} f^W_{e_i} = f^W_{e_{i+1}}, i \in [k-1].$$

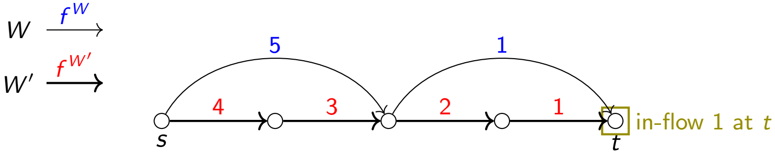
• By suitably replicating edges of E, will assume that edges of W are distinct. W is k-recurrent if it replicates original edges at most k times. A walk flow is a path flow it passes through each vertex at most once.

Walk and Path Flows

Def: For s-t walk $W = (e_1, e_2, \ldots, e_k)$, the walk flow f^W sending 1 unit of flow into t is defined by

$$\gamma_{e_k} f^W_{e_k} = 1, \quad \gamma_{e_i} f^W_{e_i} = f^W_{e_{i+1}}, i \in [k-1].$$

• By suitably replicating edges of E, will assume that edges of W are distinct. W is k-recurrent if it replicates original edges at most k times. A walk flow is a path flow it passes through each vertex at most once.



Dominating Paths by Walks

- Assumptions: edges have non-negative costs $c \in \mathbb{R}^E_{\geq 0}$,

edges have positive capacity $u \in \mathbb{R}^E_{>0}$ (allows us to assume $x^* = 0$).

Def: For s-t walks W, W', we say that W' dominates W if

$$\text{Cost Domination}: \quad \max_{e \in W'} f_e^{W'} c_e \leq \max_{e \in W} f_e^{W} c_e,$$

$$\text{Flow Domination}: \quad \max_{e \in W} \frac{f_e^{W'}}{u_e} \leq \max_{e \in W} \frac{f_e^{W}}{u_e}.$$

Dominating Paths by Walks

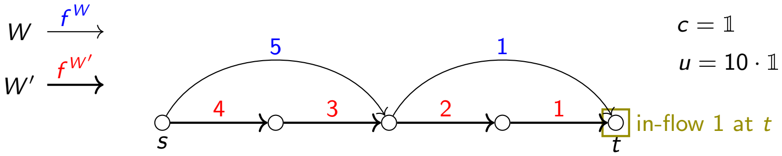
- Assumptions: edges have non-negative costs $c \in \mathbb{R}^E_{\geq 0}$,

edges have positive capacity $u \in \mathbb{R}^E_{>0}$ (allows us to assume $x^* = 0$).

Def: For s-t walks W, W', we say that W' dominates W if

$$\text{Cost Domination}: \quad \max_{e \in W'} f_e^{W'} c_e \leq \max_{e \in W} f_e^{W} c_e,$$

$$\text{Flow Domination}: \quad \max_{e \in W} \frac{f_e^{W'}}{u_e} \leq \max_{e \in W} \frac{f_e^{W}}{u_e}.$$



Dominating Paths by Walks

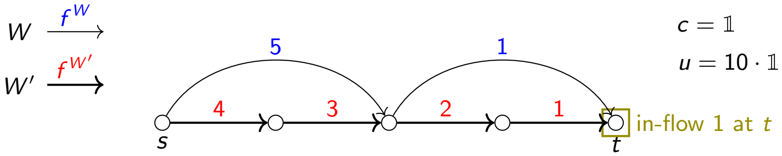
- Assumptions: edges have non-negative costs $c \in \mathbb{R}^E_{\geq 0}$,

edges have positive capacity $u \in \mathbb{R}^E_{>0}$ (allows us to assume $x^* = 0$).

Def: For s-t walks W, W', we say that W' dominates W if

$$\text{Cost Domination}: \quad \max_{e \in W'} f_e^{W'} c_e \leq \max_{e \in W} f_e^{W} c_e,$$

$$\text{Flow Domination}: \quad \max_{e \in W} \frac{f_e^{W'}}{u_e} \leq \max_{e \in W} \frac{f_e^{W}}{u_e}.$$



Core Part of Proof

Find a small set $\mathcal{W}$ of n-recurrent s-t walks such that every s-t path is dominated by some walk in $\mathcal{W}$.

Dominating Paths by Walks

Theorem [D, Koh, Natura, Olver, Végh '23]

For every $s, t \in V$, there exists a set $\mathcal{W}$ of $O(m^2)$ n-recurrent s-t walks such that every s-t path is dominated by some walk in $\mathcal{W}$.

Dominating Paths by Walks

Theorem [D, Koh, Natura, Olver, Végh '23]

For every $s, t \in V$, there exists a set $\mathcal{W}$ of $O(m^2)$ n-recurrent s-t walks such that every s-t path is dominated by some walk in $\mathcal{W}$.

- For every walk W, assign a signature (e_c, e_f) where

$$e_c := \arg\max_{e \in W} c_e f_e^W \qquad e_f := \arg\max_{e \in W} \frac{f_e^W}{u_e}.$$

We call e_c the cost bottleneck, and e_f the flow bottleneck of W.

Dominating Paths by Walks

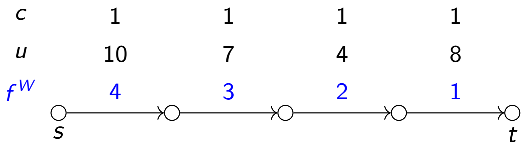
Theorem [D, Koh, Natura, Olver, Végh '23]

For every $s, t \in V$, there exists a set $\mathcal{W}$ of $O(m^2)$ n-recurrent s-t walks such that every s-t path is dominated by some walk in $\mathcal{W}$.

- For every walk W, assign a signature (e_c, e_f) where

$$e_c := \arg\max_{e \in W} c_e f_e^W \qquad e_f := \arg\max_{e \in W} \frac{f_e^W}{u_e}.$$

We call e_c the cost bottleneck, and e_f the flow bottleneck of W.



Dominating Paths by Walks

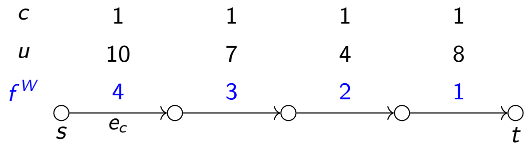
Theorem [D, Koh, Natura, Olver, Végh '23]

For every $s, t \in V$, there exists a set $\mathcal{W}$ of $O(m^2)$ n-recurrent s-t walks such that every s-t path is dominated by some walk in $\mathcal{W}$.

- For every walk W, assign a signature (e_c, e_f) where

$$e_c := \arg\max_{e \in W} c_e f_e^W \qquad e_f := \arg\max_{e \in W} \frac{f_e^W}{u_e}.$$

We call e_c the cost bottleneck, and e_f the flow bottleneck of W.



Dominating Paths by Walks

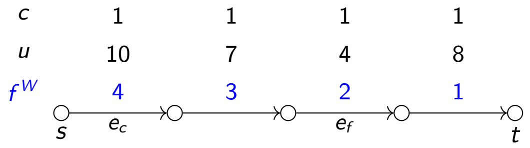
Theorem [D, Koh, Natura, Olver, Végh '23]

For every $s, t \in V$, there exists a set $\mathcal{W}$ of $O(m^2)$ n-recurrent s-t walks such that every s-t path is dominated by some walk in $\mathcal{W}$.

- For every walk W, assign a signature (e_c, e_f) where

$$e_c := \arg\max_{e \in W} c_e f_e^W \qquad\qquad e_f := \arg\max_{e \in W} \frac{f_e^W}{u_e}.$$

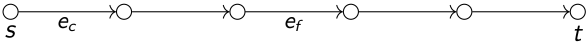
We call e_c the cost bottleneck, and e_f the flow bottleneck of W.



Path Patching

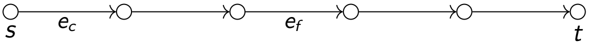
Path Patching

- Let P be an s-t path with signature (e_c, e_f).



Path Patching

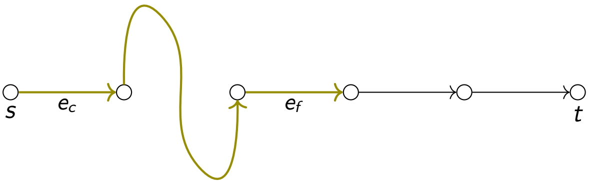
- Let P be an s-t path with signature (e_c, e_f).



Def: Let patch(P) be the walk obtained from P by replacing the e_c-e_f subpath with a max gain e_c-e_f path of signature (e_c, e_f).

Path Patching

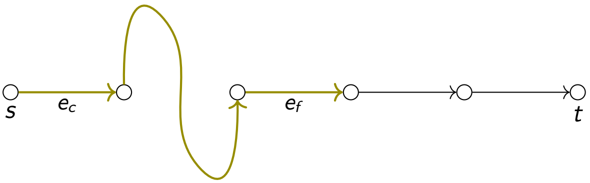
- Let P be an s-t path with signature (e_c, e_f).



Def: Let patch(P) be the walk obtained from P by replacing the e_c-e_f subpath with a max gain e_c-e_f path of signature (e_c, e_f).

Path Patching

- Let P be an s-t path with signature (e_c, e_f).



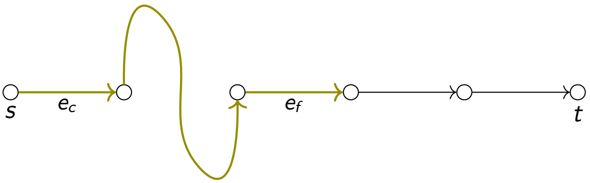
Def: Let patch(P) be the walk obtained from P by replacing the e_c-e_f subpath with a max gain e_c-e_f path of signature (e_c, e_f).

Patching Lemma:

1. patch(P) dominates P.

Path Patching

- Let P be an s-t path with signature (e_c, e_f).



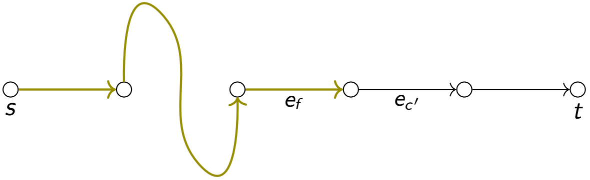
Def: Let patch(P) be the walk obtained from P by replacing the e_c-e_f subpath with a max gain e_c-e_f path of signature (e_c, e_f).

Patching Lemma:

1. patch(P) dominates P.
2. The signature of patch(P) is either (e_c, e_f) or (e'_c, e_f), where e'_c comes after e_f.

Path Patching

- Let P be an s-t path with signature (e_c, e_f).



Def: Let patch(P) be the walk obtained from P by replacing the e_c-e_f subpath with a max gain e_c-e_f path of signature (e_c, e_f).

Patching Lemma:

1. patch(P) dominates P.
2. The signature of patch(P) is either (e_c, e_f) or (e'_c, e_f), where e'_c comes after e_f.

Dominating Paths by Walks

Dominating Paths by Walks

- For an s-t path P, let $W_1, W_2, \ldots, W_k$ be the sequence of walks obtained by repeatedly patching until the signature stops changing, i.e.

$$W_1 = \text{patch}(P) \qquad\qquad W_i = \text{patch}(W_{i-1}) \quad \forall i \geq 2.$$

Dominating Paths by Walks

- For an s-t path P, let $W_1, W_2, \ldots, W_k$ be the sequence of walks obtained by repeatedly patching until the signature stops changing, i.e.

$$W_1 = \text{patch}(P) \qquad\qquad W_i = \text{patch}(W_{i-1}) \quad \forall i \geq 2.$$

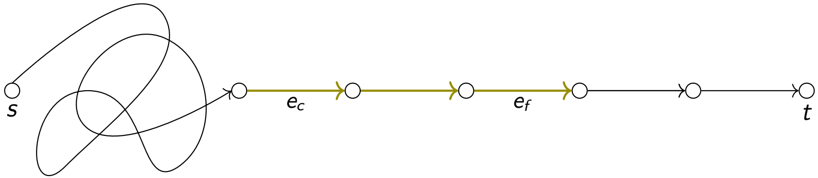
- By patching lemma, W_k dominates P and $k \leq n$.

Dominating Paths by Walks

- For an s-t path P, let $W_1, W_2, \ldots, W_k$ be the sequence of walks obtained by repeatedly patching until the signature stops changing, i.e.

$$W_1 = \text{patch}(P) \qquad W_i = \text{patch}(W_{i-1}) \quad \forall i \geq 2.$$

- By patching lemma, W_k dominates P and $k \leq n$.

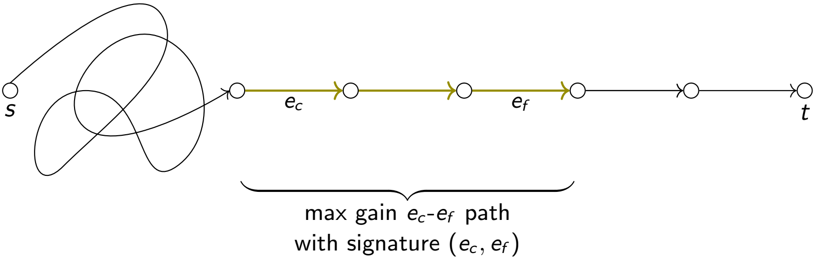


Dominating Paths by Walks

- For an s-t path P, let $W_1, W_2, \ldots, W_k$ be the sequence of walks obtained by repeatedly patching until the signature stops changing, i.e.

$$W_1 = \text{patch}(P) \qquad W_i = \text{patch}(W_{i-1}) \quad \forall i \geq 2.$$

- By patching lemma, W_k dominates P and $k \leq n$.

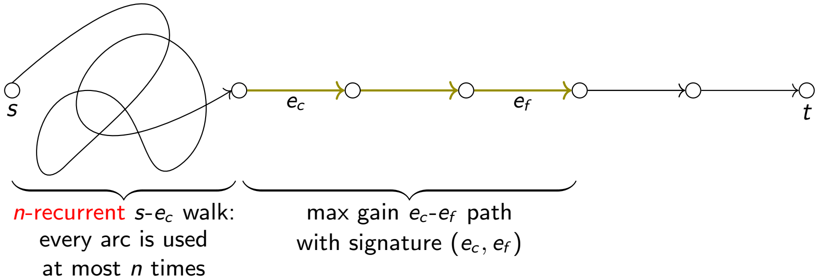


Dominating Paths by Walks

- For an s-t path P, let $W_1, W_2, \dots, W_k$ be the sequence of walks obtained by repeatedly patching until the signature stops changing, i.e.

$$W_1 = \text{patch}(P) \qquad W_i = \text{patch}(W_{i-1}) \quad \forall i \geq 2.$$

- By patching lemma, W_k dominates P and $k \leq n$.

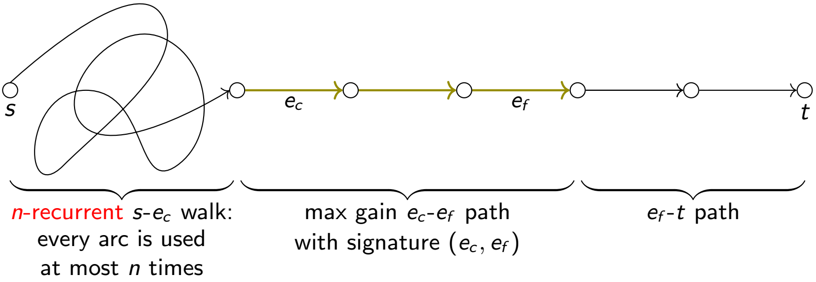


Dominating Paths by Walks

- For an s-t path P, let $W_1, W_2, \ldots, W_k$ be the sequence of walks obtained by repeatedly patching until the signature stops changing, i.e.

$$W_1 = \text{patch}(P) \qquad W_i = \text{patch}(W_{i-1}) \quad \forall i \geq 2.$$

- By patching lemma, W_k dominates P and $k \leq n$.



The Dominating Set of Walks $\mathcal{W}$

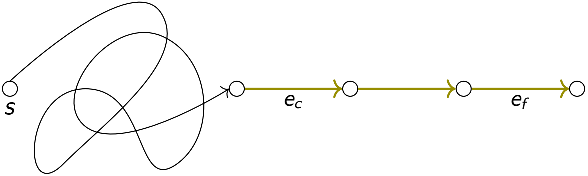
- For every signature (e_c, e_f),

The Dominating Set of Walks $\mathcal{W}$

- For every signature (e_c, e_f),
  1. Start with a max gain e_c-e_f path with signature (e_c, e_f).

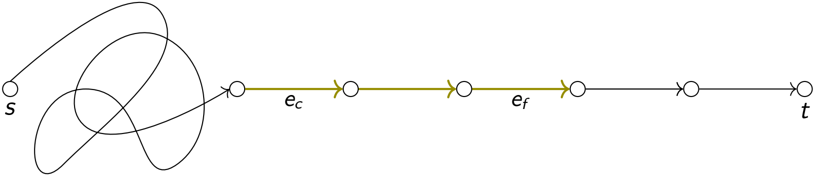
The Dominating Set of Walks $\mathcal{W}$

- For every signature (e_c, e_f),
  1. Start with a max gain e_c-e_f path with signature (e_c, e_f).
  2. Append a max gain n-recurrent s-e_c walk which preserves signature.



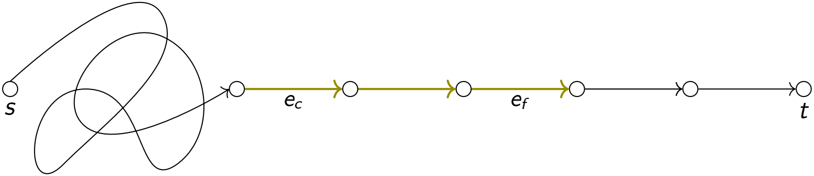
The Dominating Set of Walks $\mathcal{W}$

- For every signature (e_c, e_f),
  1. Start with a max gain e_c-e_f path with signature (e_c, e_f).
  2. Append a max gain n-recurrent s-e_c walk which preserves signature.
  3. Append a max gain e_f-t path which preserves signature.



The Dominating Set of Walks $\mathcal{W}$

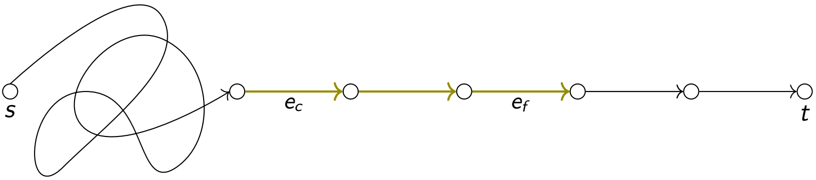
- For every signature (e_c, e_f),
  1. Start with a max gain e_c-e_f path with signature (e_c, e_f).
  2. Append a max gain n-recurrent s-e_c walk which preserves signature.
  3. Append a max gain e_f-t path which preserves signature.



- Analogous construction for the case where e_c comes after e_f.

The Dominating Set of Walks $\mathcal{W}$

- For every signature (e_c, e_f),
  1. Start with a max gain e_c-e_f path with signature (e_c, e_f).
  2. Append a max gain n-recurrent s-e_c walk which preserves signature.
  3. Append a max gain e_f-t path which preserves signature.



- Analogous construction for the case where e_c comes after e_f.
- $|\mathcal{W}| = O(m^2)$.

Conclusion

Conclusion

- SLC of minimum cost generalized flow is poly(m, n).

Conclusion

- SLC of minimum cost generalized flow is poly(m, n).
- Strongly polynomial algorithm for LPs with ≤ 2 variables per inequality.

Conclusion

- SLC of minimum cost generalized flow is poly(m, n).
- Strongly polynomial algorithm for LPs with ≤ 2 variables per inequality.
- [Allamigeon, Benchimol, Gaubert, Joswig '18] There exist LPs with

$$\mathsf{SLC}_\theta(x_i^{\mathrm{mcp}}) = 2^{\Omega(m)}.$$

Conclusion

- SLC of minimum cost generalized flow is poly(m, n).
- Strongly polynomial algorithm for LPs with ≤ 2 variables per inequality.
- [Allamigeon, Benchimol, Gaubert, Joswig '18] There exist LPs with

$$\mathsf{SLC}_\theta(x_i^{\mathrm{mcp}}) = 2^{\Omega(m)}.$$

- Future directions:
  - Develop a theory of SLC for LPs.

Conclusion

- SLC of minimum cost generalized flow is poly(m, n).
- Strongly polynomial algorithm for LPs with ≤ 2 variables per inequality.
- [Allamigeon, Benchimol, Gaubert, Joswig '18] There exist LPs with

$$\mathsf{SLC}_\theta(x_i^{\mathrm{mcp}}) = 2^{\Omega(m)}.$$

- Future directions:
  - Develop a theory of SLC for LPs.
  - Undiscounted MDP: strongly polynomial solvability/straight line complexity open.

Conclusion

- SLC of minimum cost generalized flow is poly(m, n).
- Strongly polynomial algorithm for LPs with ≤ 2 variables per inequality.
- [Allamigeon, Benchimol, Gaubert, Joswig '18] There exist LPs with

$$\mathrm{SLC}_\theta(x_i^{\mathrm{mcp}}) = 2^{\Omega(m)}.$$

- Future directions:
  - Develop a theory of SLC for LPs.
  - Undiscounted MDP: strongly polynomial solvability/straight line complexity open.
  - Faster strongly polynomial algorithm for minimum cost generalized flow.

Conclusion

- SLC of minimum cost generalized flow is poly(m, n).
- Strongly polynomial algorithm for LPs with ≤ 2 variables per inequality.
- [Allamigeon, Benchimol, Gaubert, Joswig '18] There exist LPs with

$$\mathsf{SLC}_\theta(x_i^{\mathrm{mcp}}) = 2^{\Omega(m)}.$$

- Future directions:
  - Develop a theory of SLC for LPs.
  - Undiscounted MDP: strongly polynomial solvability/straight line complexity open.
  - Faster strongly polynomial algorithm for minimum cost generalized flow.

Thank you!