

A Strongly Polynomial Algorithm for the Minimum Cost Generalized Flow Problem

Zhuan Khye (Cedric) Koh

Joint work with

Daniel Dadush, Bento Natura, Neil Olver, László A. Végh



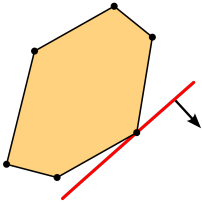
Talk Overview

- 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:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s. t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$



Dual:

$$\begin{aligned} \max \quad & b^\top y \\ \text{s. t.} \quad & A^\top y \leq c \end{aligned}$$

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

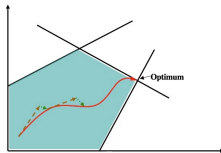
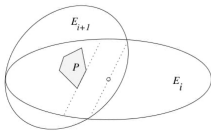


LP Algorithms

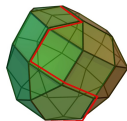
Input: $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. Total bit length L .

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

Def: An algorithm is **strongly polynomial** if it uses

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

Smale's 9th Problem [Megiddo '83]

Is there a **strongly polynomial** algorithm for linear programming?



LP Zoo: Strongly Polynomial View

General LP \equiv LP with ≤ 3 variables per inequality

Strongly polynomial (before 2024)

Combinatorial LP:
(small $\|A\|_\infty$)

- 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

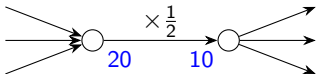
- Undiscounted MDP
- LP with ≤ 2 variables per inequality

2-Variables-per-Inequality LP

- Every 2-variables-per-inequality LP can be reduced to the **minimum cost generalized flow** problem [Hochbaum '04]:

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

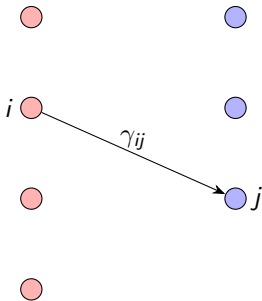
Interpretation: Given a directed graph $G = (V, E)$, node demands $b \in \mathbb{R}^V$, arc costs $c \in \mathbb{R}^E$ and gain factors $\gamma \in \mathbb{R}_{>0}^E$, find a **generalized flow** satisfying all node demands at minimum cost.



Models energy loss,
currency exchange etc.

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 .

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

Previous Algorithms for Generalized Flow

- Algorithms for **primal** feasibility:
 - ▶ Polynomial [Goldberg, Plotkin, Tardos '91]
 - ▶ Strongly polynomial [Végh '13] [Olver, Végh '20]
- Algorithms for **dual** feasibility:
 - ▶ Polynomial [Aspvall, Shiloach '80]
 - ▶ Strongly polynomial [Megiddo '83] [Cohen, Megiddo '94]
[Hochbaum, Naor '94] [Dadush, K, Natura, Végh '21]
[Karczmarz '22]
- Algorithms for optimization:
 - ▶ Polynomial [Wayne '02]

Main Result

Theorem [Dadush, K, Nataraj, 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.

- The algorithm is based on the interior point method (IPM) by [Allamigeon, Dadush, Loh, Nataraj, Végh '22].
- What we'll need for this result:
 - ① Interior point method
 - ② Straight line complexity

Central Path

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

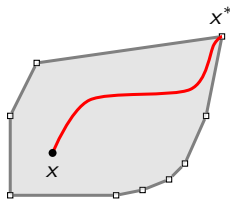
$$\min c^\top x - \mu \sum_{i=1}^n \log(x_i)$$

$$\text{s. t. } Ax = b.$$

Def: The **central path** is the curve

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

- As $\mu \rightarrow 0$, $x^{\text{cp}}(\mu)$ converges to an optimal solution x^* of the LP.
- Interior Point Method (IPM):** Walk down the central path with geometrically decreasing μ .



Max Central Path

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

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

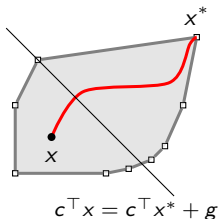
- For every $g \geq 0$ and $i \in [m]$, define

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

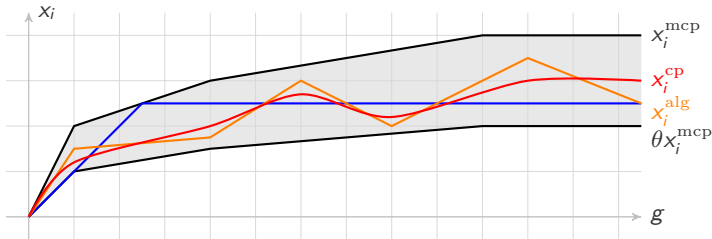
Def: The **max central path** is the curve $\{x^{\text{mcp}}(g) : g \geq 0\}$,

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

$$\frac{1}{2m} x^{\text{mcp}} \leq x^{\text{cp}} \leq x^{\text{mcp}}$$



Straight Line Complexity



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

$$\theta x^{\text{mcp}} \leq x^{\text{alg}} \leq x^{\text{mcp}}.$$

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

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

Straight Line Complexity

- # iterations required by any IPM is at least

$$\max_{i \in [m]} \text{SLC}_{\theta}(x_i^{\text{mcp}}).$$

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

There is an interior point method which solves LP in

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

iterations.

Main Result

Theorem [Dadush, K, Natura, Olver, Vég h '24]

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

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

- Key ingredient: **Circuits**

Theorem [Dadush, K, Natura, Olver, Vég h '24]

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

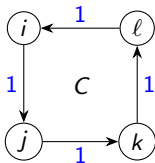
Circuits

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 $\text{supp}(h) \subsetneq \text{supp}(f)$.

Example: Network flow

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

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



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

$f^C = \mathbf{0}$

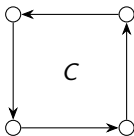
Circuits of Generalized Flow

- For generalized flow,

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

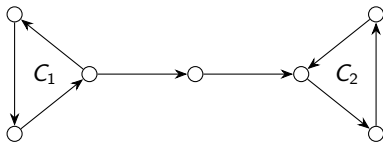
- $\ker(A)$ = set of generalized circulations. 2 types of circuits:

Conservative cycle



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

Bicycle



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

flow-generating

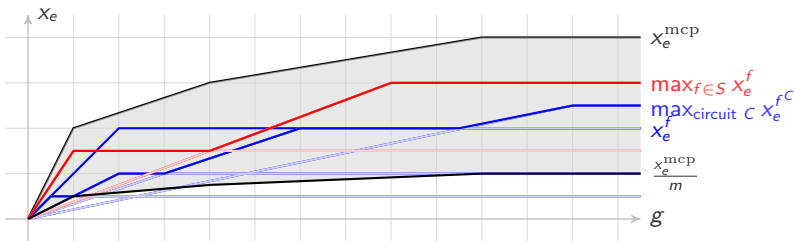
$$\prod_{e \in C_2} \gamma_e < 1$$

flow-absorbing

Upper Bounding the SLC

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

$$x^f(g) := x^* + \frac{g}{c^\top f}.$$



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

Key Idea: Find $S \subseteq \ker(A)$ such that $\max_{f \in S} x_e^f \geq \max_{\text{circuit } C} x_e^{f^C}$.

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

Conclusion & Future Directions

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

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

- Future directions:
 - ▶ Undiscounted MDP: strongly polynomial solvability/straight line complexity open.
 - ▶ Faster strongly polynomial algorithm for minimum cost generalized flow.

Thank you!