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

Bento Natura

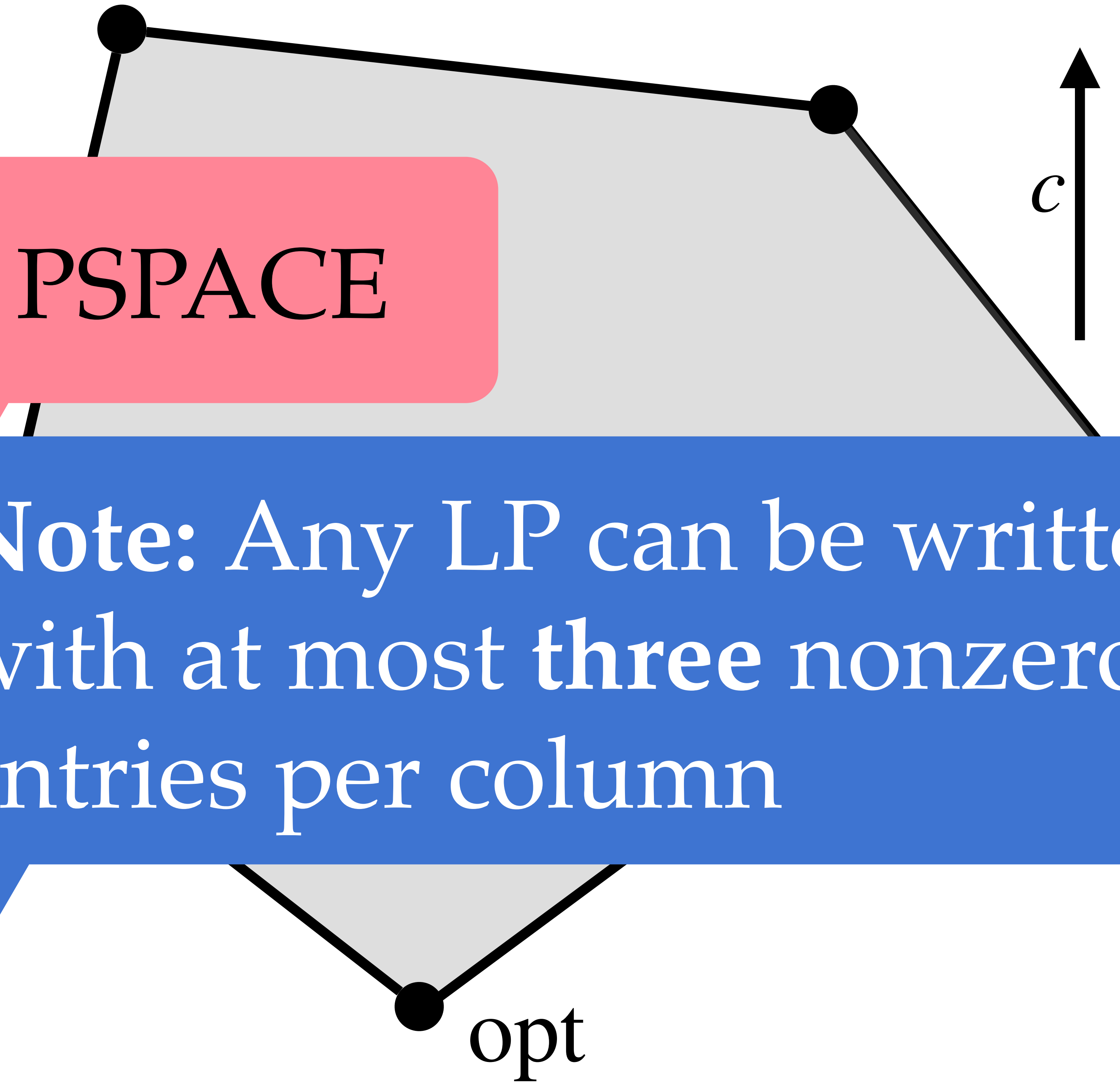
MIP Workshop 2024

Joint work with Daniel Dadush, Zhuan Khye Koh, Neil Olver, and László Végh

Linear Programming (LP)

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

LP can be solved in time $\text{poly}\big(m, \text{size}(A, b, c)\big)$
(Khachiyan '79 (Ellipsoid Method), Karmarkar '84
(Interior Point Methods),…



+ PSPACE

Note: Any LP can be written with at most **three** nonzero entries per column

Is there a ***strongly*** polynomial algorithm for LP?
…i.e. an algorithm with running time $\text{poly}(m)$…

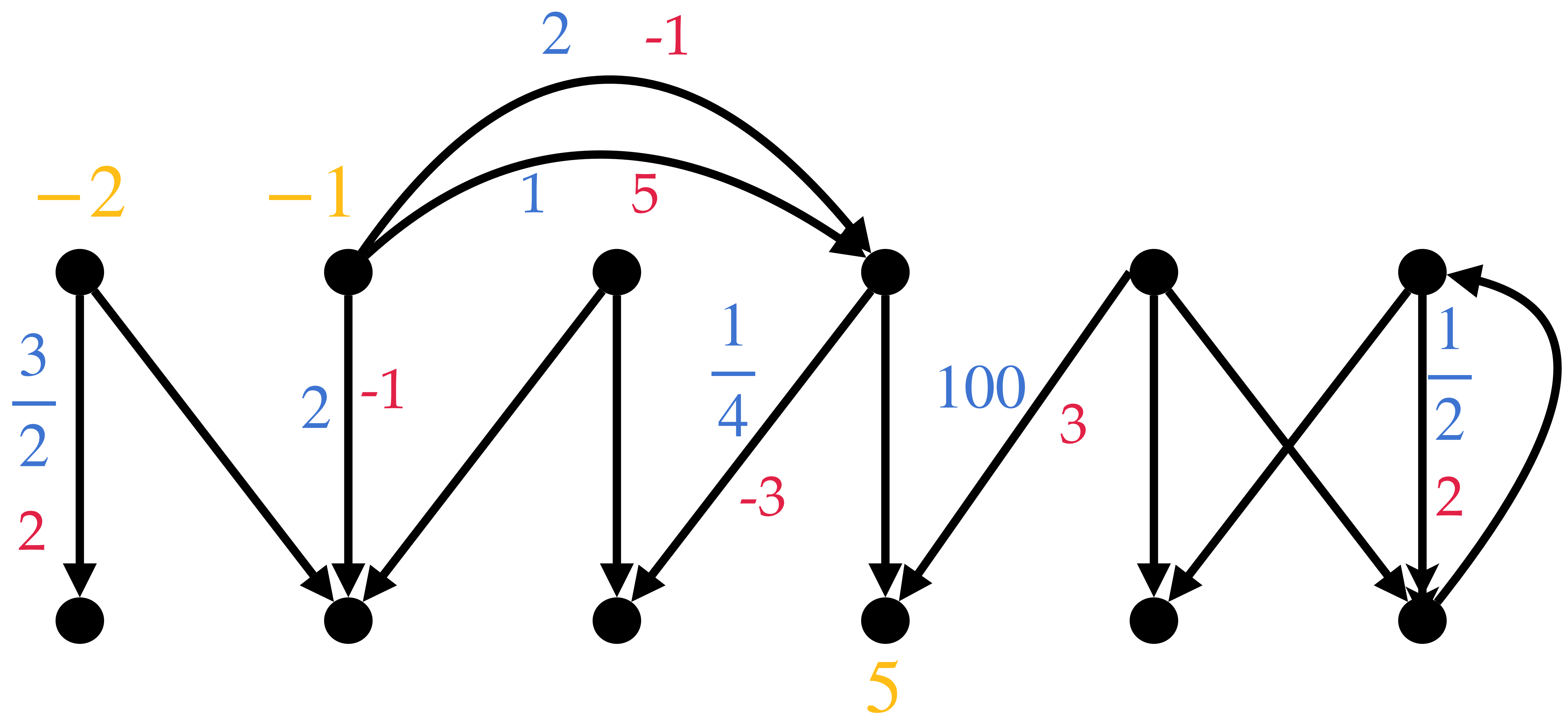
Dadush, Koh, N., Olver, Végh '24:
There exists a strongly polynomial time algorithm for LP with **at most two nonzero** entries per column.

The minimum-cost generalized flow problem

Primal: $\min\langle c, x\rangle : Ax = b, x \geq 0$ ***Dual***: $\max\langle y, b\rangle : A^\top y \leq c$

Primal: $$\min\langle c, x\rangle : \sum_{e\in\delta^-(i)} \gamma_e x_e - \sum_{e\in\delta^+(i)} x_e = b_i, \forall i \in [n], \quad x \geq 0$$

Dual: $$\max\langle y, b\rangle : \gamma_e y_j - y_i \leq c_e \quad \forall e = (i,j)$$



Hochbaum '04: LP with 2 variables per column can be reduced to minimum-cost generalized flow

Prior strongly polynomial special cases

$$\min\langle c, x\rangle : \sum_{e\in\delta^-(i)} \gamma_e x_e - \sum_{e\in\delta^+(i)} x_e = b_i, \forall i \in [n], \quad x \geq 0 \qquad \textbf{(MCGF)}$$

Special gains γ

- $\gamma \equiv 1 \quad \Rightarrow$ Minimum cost flow problem
  $\Rightarrow$ **Tardos '85**
- $\gamma \in \mathbb{Z}^m$ and $\log(\|\gamma\|_\infty) = O(\text{poly}(m))$
  $\Rightarrow$ **Tardos '86**

Primal feasibility

- Végh '13: $\tilde{O}(m^2n^3)$ answering longstanding open question
- Olver, Végh '20: "Simpler and faster" now in $\tilde{O}(m^2n)$

Dual feasibility

- First *strongly* polynomial algorithm: Seminal work by Megiddo '83 introducing *parametric search technique* (Meta algorithm, binary search on steroids)
- Hochbaum-Naor '94: $\tilde{O}(mn^2)$ fastest deterministic
- First algorithm <u>not</u> relying on parametric search: Dadush, Koh, N. and Végh '21: usage of Discrete Newton method

Our Road to solve the MCGF problem

"A Simpler and Faster Strongly Polynomial Algorithm for Generalized Flow Maximization" - Olver, Végh STOC '17, JACM '20. Fastest/Cleanest (combinatorial) **primal feasibility** algorithm

↓

Question: Find a more combinatorial/structured algorithm that solves 2VPI? (Somewhen in 2019)

↓

Discrete Newton Method (DN) is strongly polynomial for dual feasibility- Dadush, Koh, N., Végh '20
First combinatorial **dual feasibility** algorithm.

Question: Combine combinatorial **primal feasibility** and **dual feasibility** algorithms to tackle **optimization MCGF** problem?

↓

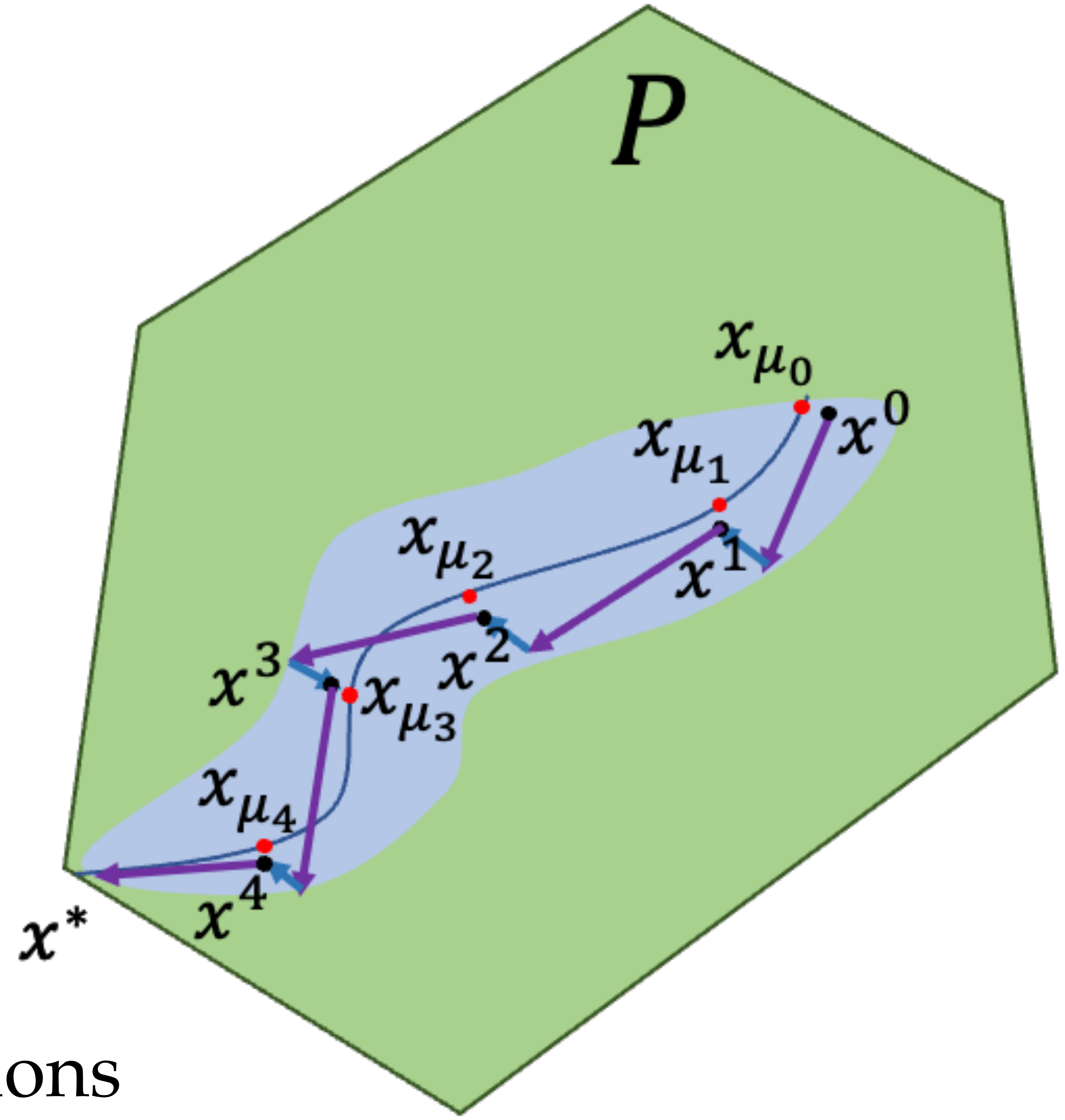
No progress :(

Question: IPM are usually most efficient methods for LP. Is there an IPM with running time $f(m, n)$?

Predictor - Corrector Path Following

Mizuno-Todd-Ye '93

- Given x^0 in 'neighborhood' around x_{μ_0} for some $\mu_0 > 0$
- Compute iterates $x^1, \ldots, x^t$ by alternating between
  - Predictor steps: decrease μ by moving 'down' the central path
  - Corrector steps: move back 'closer' to the central path for the same μ (Newton step).



Each iteration takes $O(1)$ linear system solves

Standard analysis: Decrease μ by a factor of 2 in $O(\sqrt{m})$ iterations

Prior Exact Interior Point Methods

$\min\langle c, x\rangle : Ax = b, x \geq 0$, m variables, n equalities

Layered-least-squares (LLS)

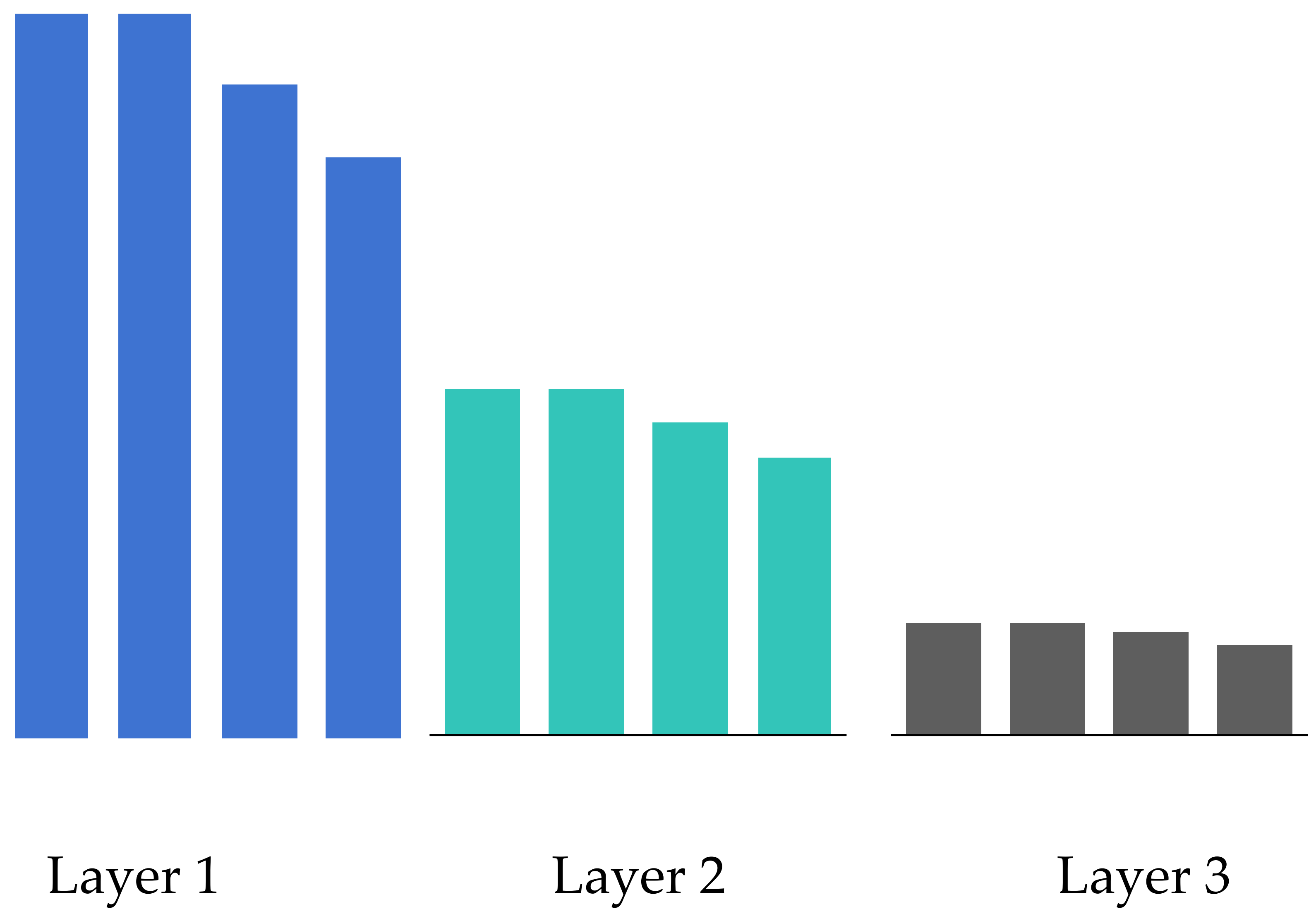
Vavasis-Ye '96, Monteiro-Tsuchiya '03 - '05,

Trust-region based IPM

Lan-Monteiro-Tsuchiya '09

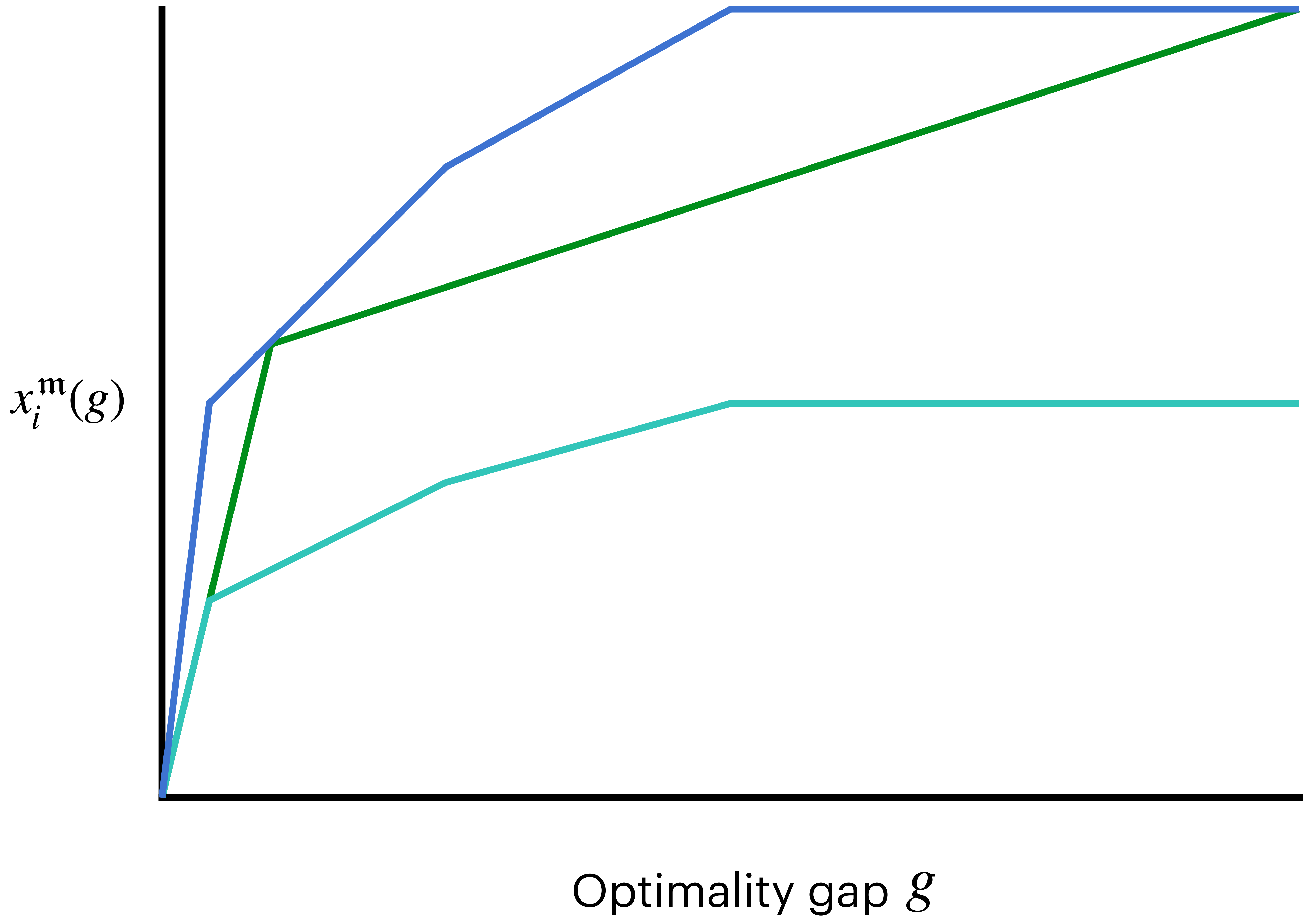
Scaling-invariant LLS

Dadush-Huiberts-N.-Végh '20



Number of iterations to solve LP *exactly* depends on condition number of matrix A

Straight Line Complexity

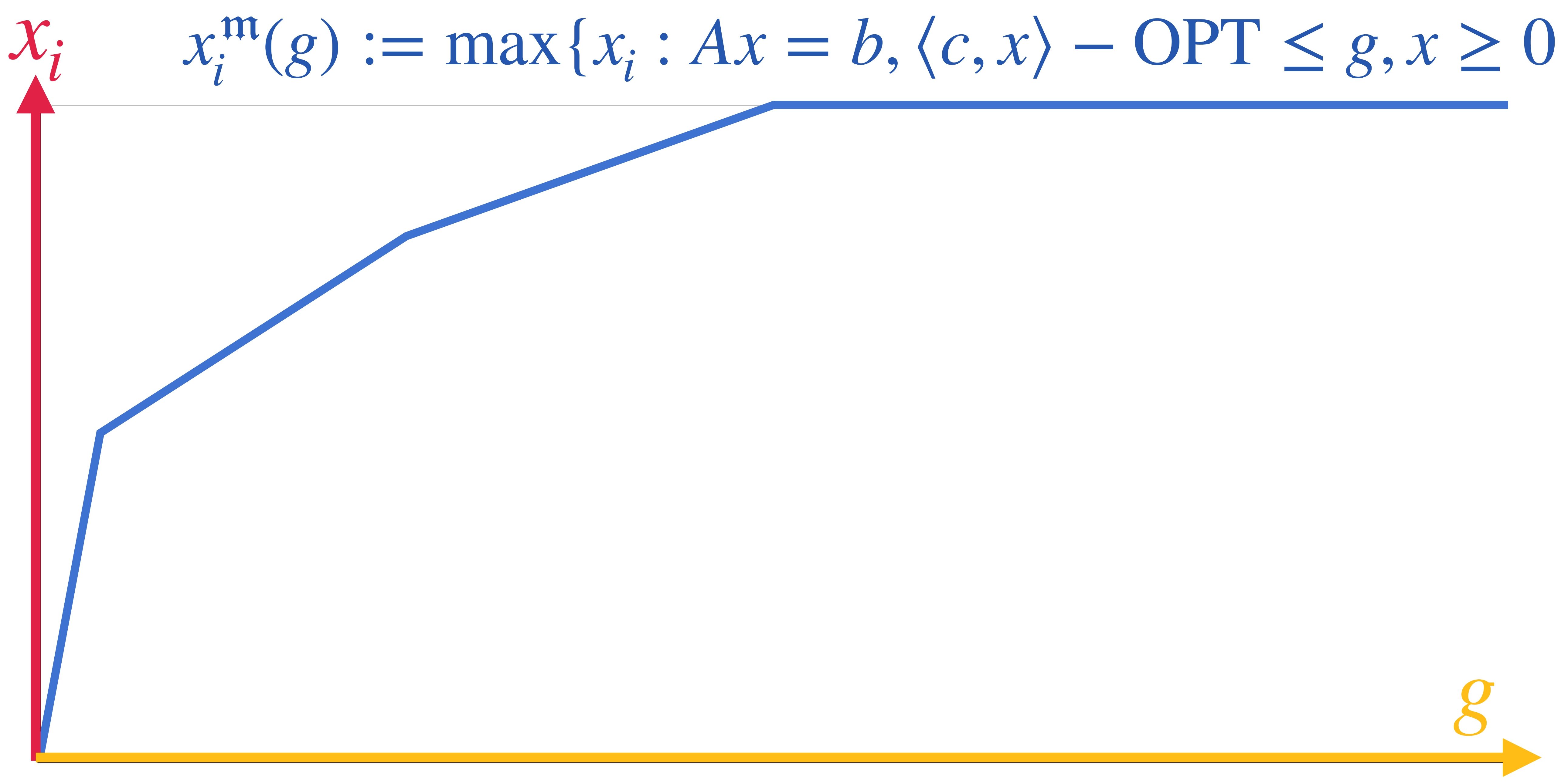


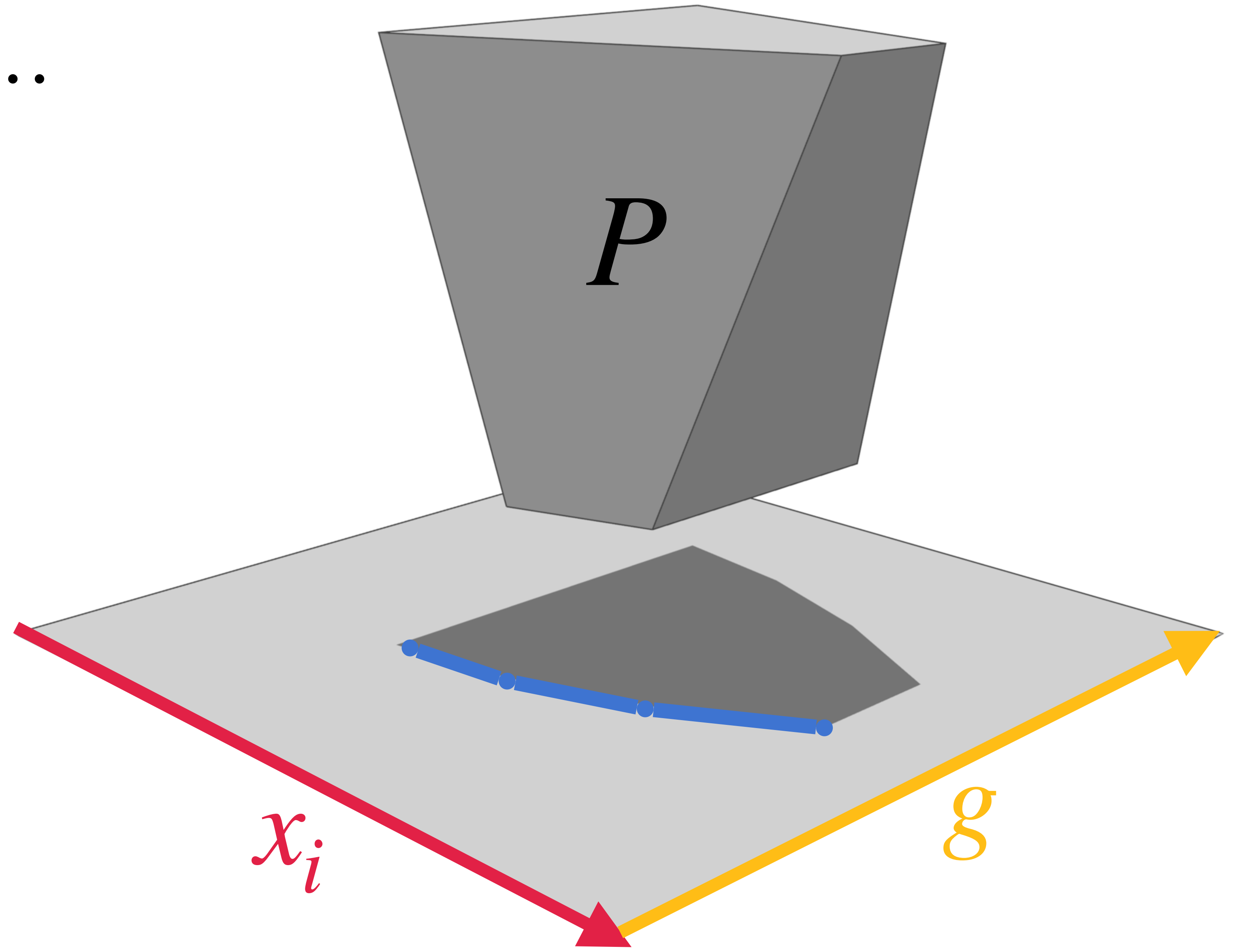
The max central path

$$\min\langle c, x\rangle : Ax = b, x \geq 0, \quad A \in \mathbb{R}^{n\times m}, P := \{x : Ax = b, x \geq 0\}$$

For any variable $i \in [m]$...

$x_i^{\mathrm{m}}(g) := \max\{x_i : Ax = b, \langle c, x\rangle - \mathrm{OPT} \leq g, x \geq 0\}$





$x_i^{\mathrm{m}}(g)$ is:

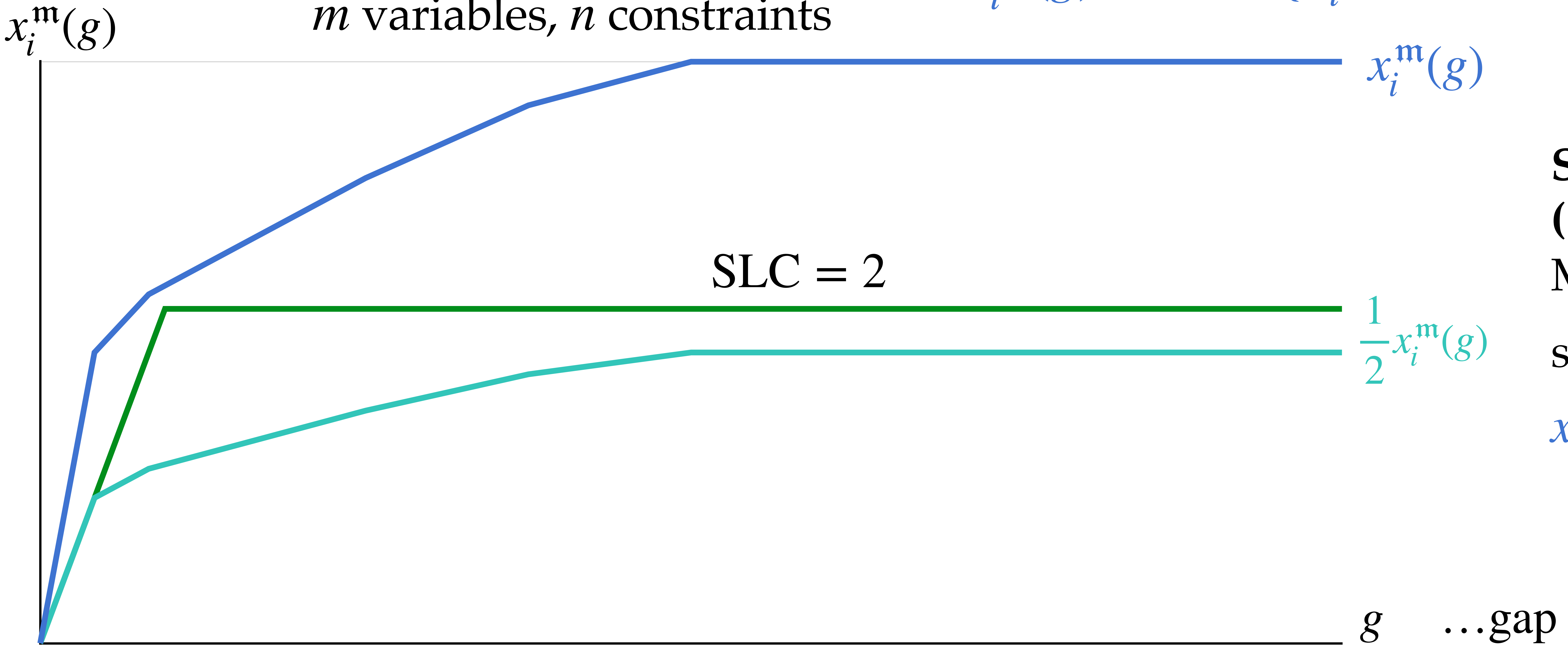
- Concave
- Monotone increasing
- Piecewise linear
- #pieces $\leq$ min(#edges of P, #vertices of P)

- Breakpoints of x_i^{m} correspond to vertices of P
- Line segments of x_i^{m} correspond to edges of P

Straight line complexity

LP: $\min\langle c, x\rangle : Ax = b, x \geq 0,$
m variables, n constraints

$x_i^{\mathfrak{m}}(g) := \max\{x_i : Ax = b, \langle c, x\rangle - \mathrm{OPT} \leq g, x \geq 0\}$



Straight Line Complexity (SLC^i):
Minimum number of linear segments between $\frac{1}{2}x_i^{\mathfrak{m}}$ and $x_i^{\mathfrak{m}}$ on $[0,\infty]$.

Theorem (Allamigeon, Dadush, Loho, N., Végh '22):
Given a suitable initial point, there exists an IPM that solves an LP in strongly polynomial many iterations if for all variables $i \in [m]$ we have that $\mathrm{SLC}(x_i^{\mathfrak{m}}) = O(\mathrm{poly}(m, n))$.

SLC for maximum flow

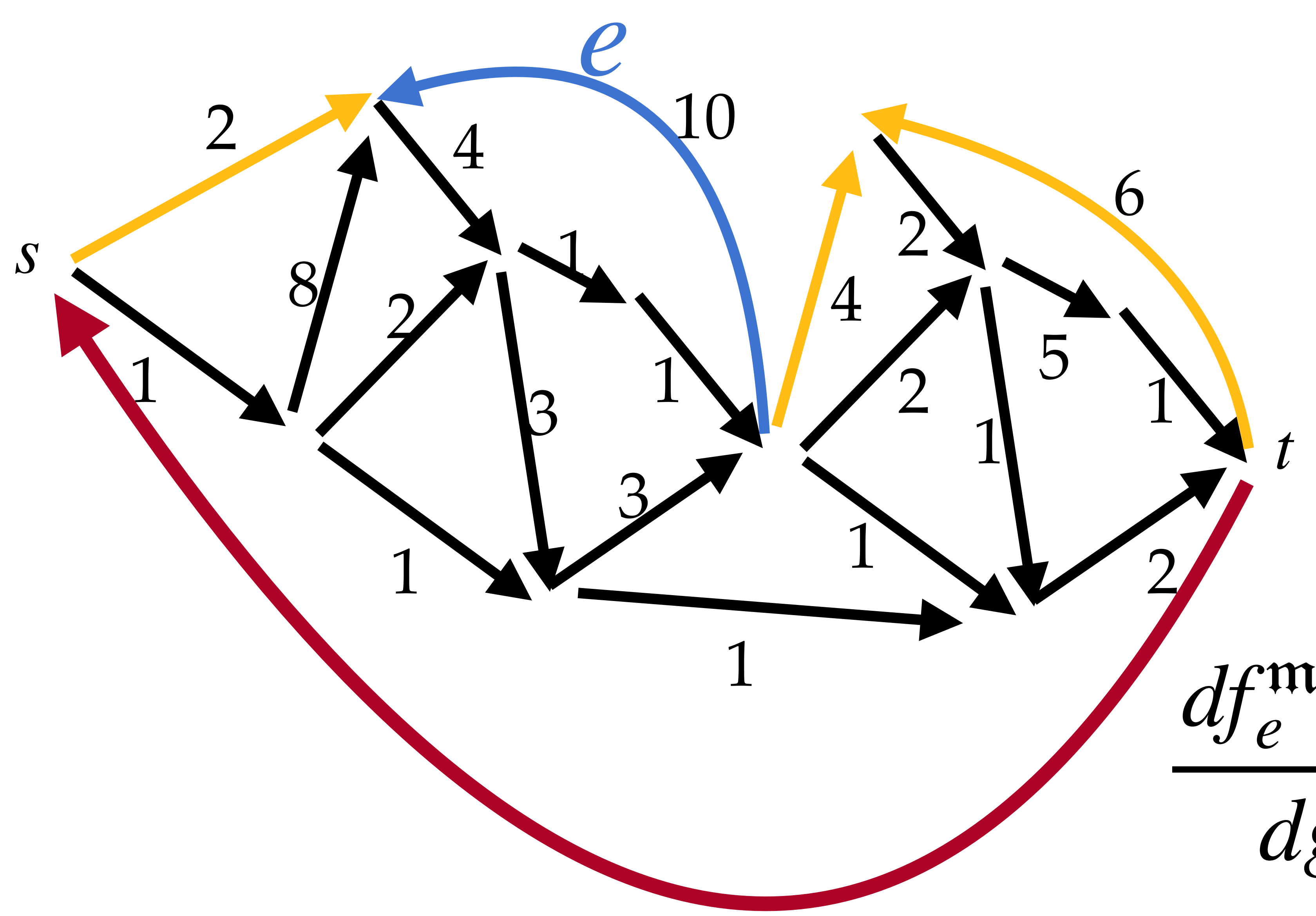
Instance: directed graph $G = (V, E)$, capacities $u : E \to \mathbb{R}_{\geq 0}$, special arc ts

Goal: $\max f_{ts} : \sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0 \forall v \in V(G), \mathbf{0} \leq f \leq u$

Todo: Analyze the SLC of $f_e^{\mathfrak{m}}$ for some edge e. Recall: the segments of $f_e^{\mathfrak{m}}$ correspond to edges of the flow polytope. Edges of the flow polytope correspond to cycles in the graph.

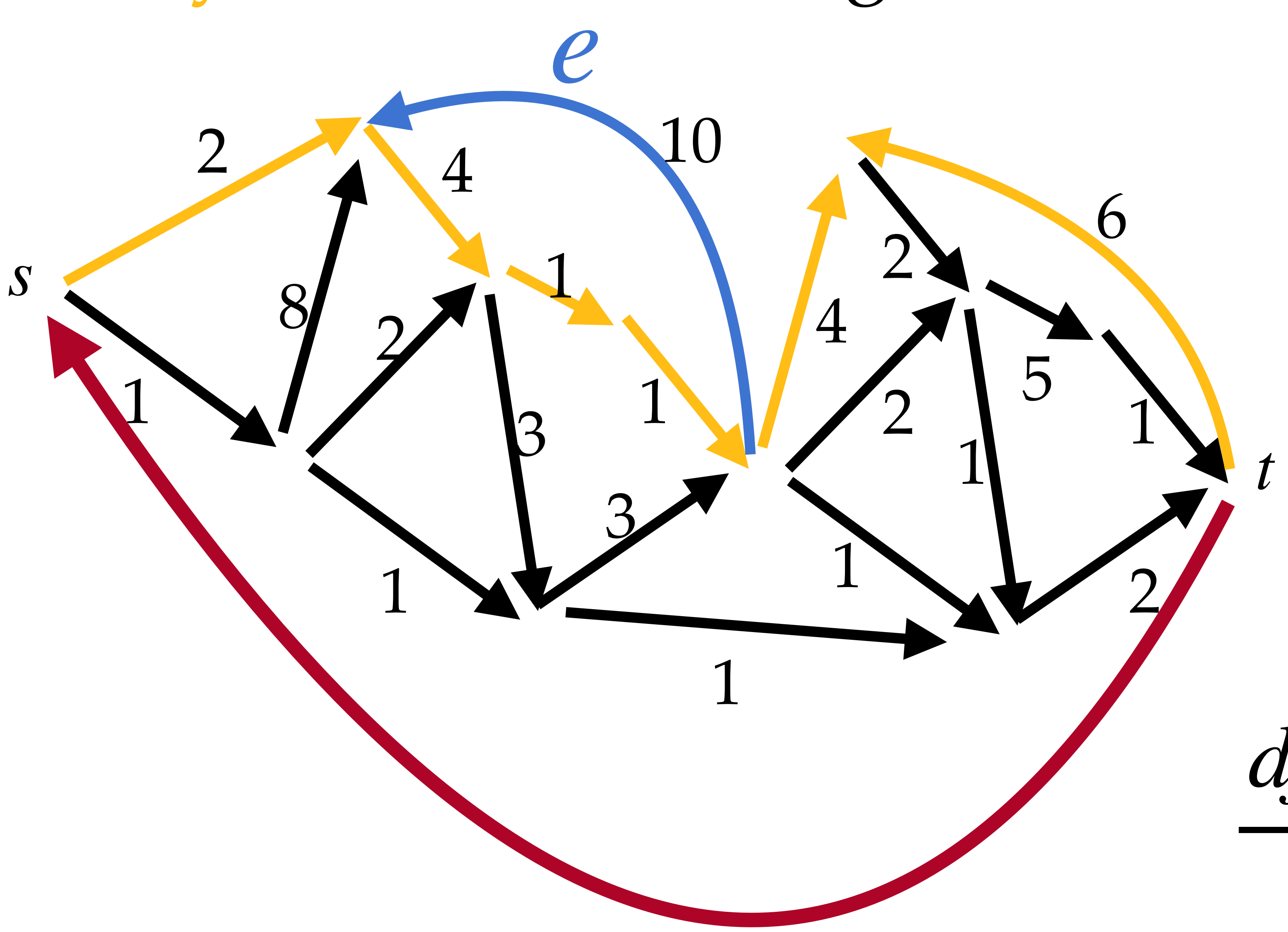
There are only two types of circuits involving the edge e:

Cycles involving the arc e



$$\frac{df_e^{\mathfrak{m}}(g)}{dg} = 1$$

Cycles ***not*** involving the arc e



$$\frac{df_e^{\mathfrak{m}}(g)}{dg} = 0$$

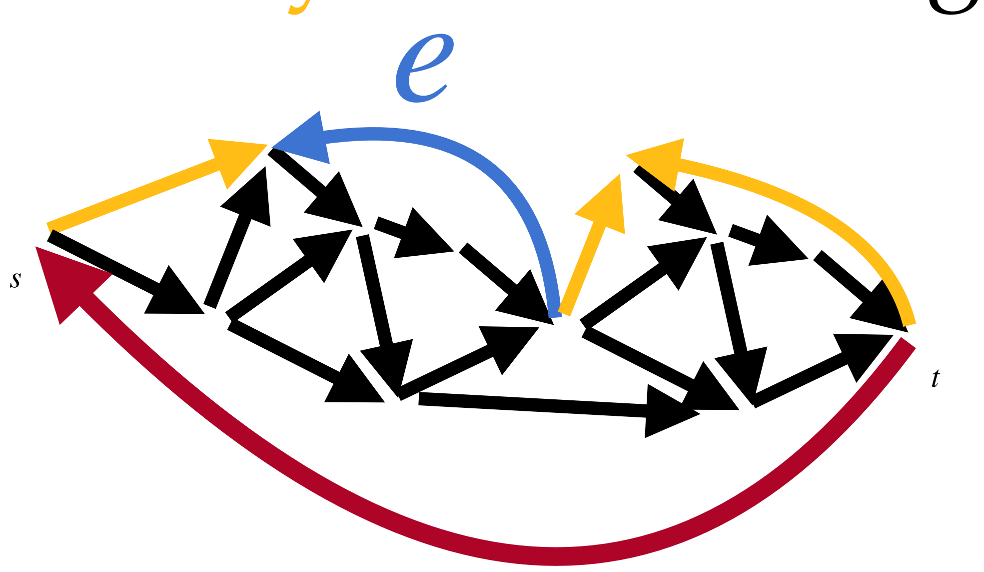
SLC for maximum flow

Instance: directed graph $G = (V, E)$, capacities $u : E \to \mathbb{R}_{\geq 0}$, special arc ts

Goal: $\max f_{ts} : \sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0 \, \forall v \in V(G), \mathbf{0} \leq f \leq u$

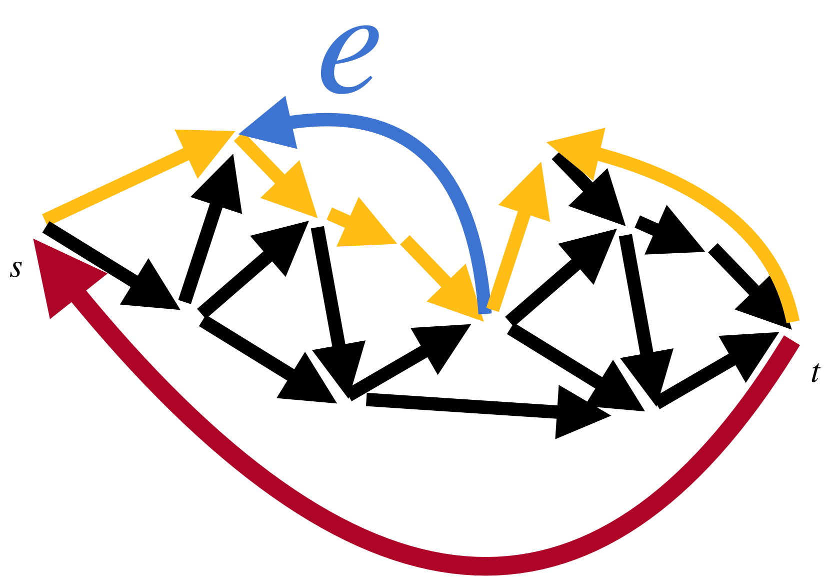
Todo: Analyze the SLC of f_e^{m} for some edge e. Recall: the segments of f_e^{m} correspond to edges of the flow polytope. Edges of the flow polytope correspond to cycles in the graph.

Cycles involving the arc e



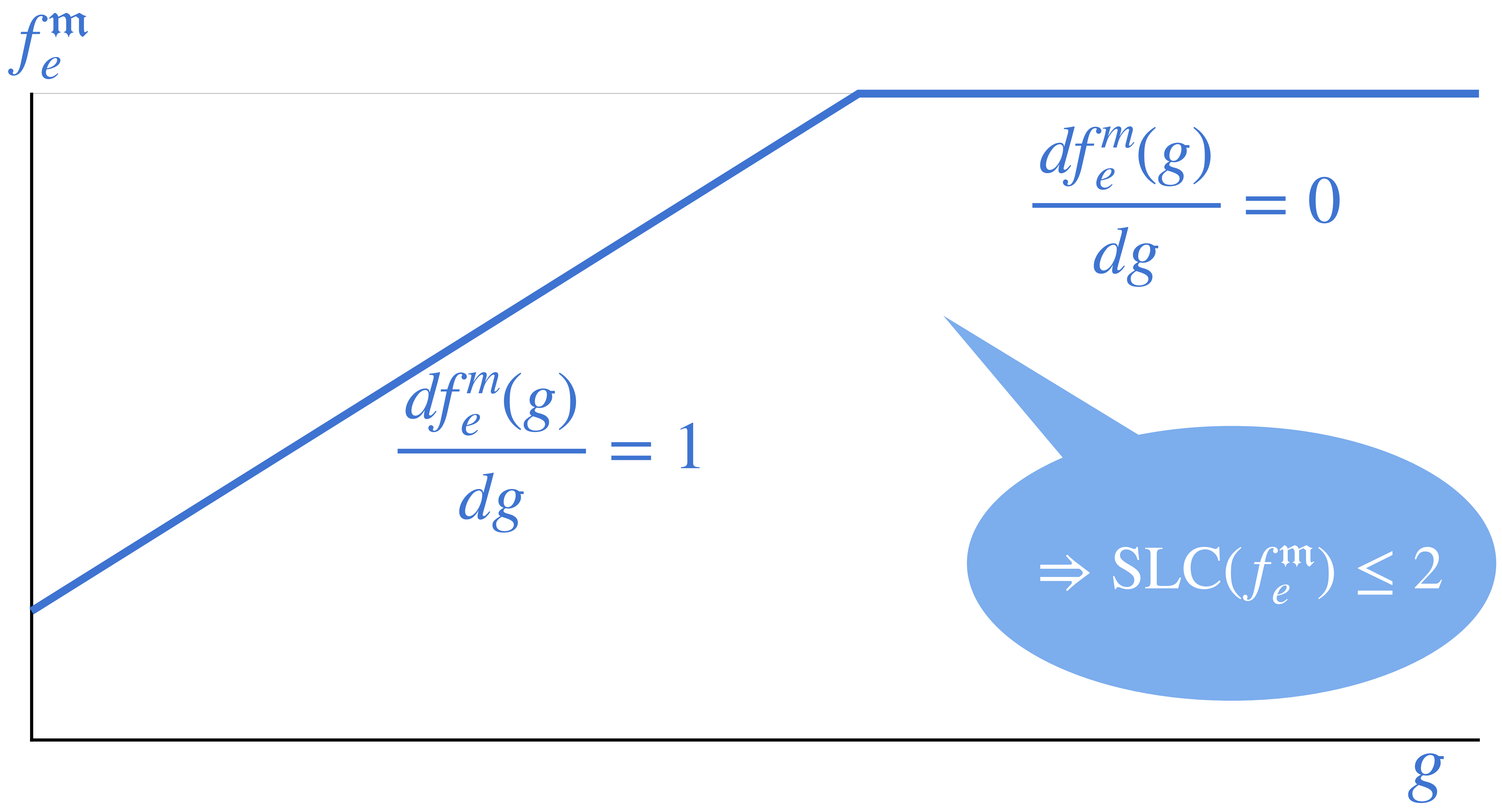
$$\frac{df_e^m(g)}{dg} = 1$$

Cycles ***not*** involving the arc e



$$\frac{df_e^m(g)}{dg} = 0$$

$\Rightarrow$



The Zoo of LP subclasses

Strongly polynomial (known before 2022)

LP in small dimension $n = O(\log^2(m)/\log\log m)$

Specialized Interior Point Methods are strongly polynomial

Combinatorial LP: A integral, $\|A\|_\infty = 2^{O(\text{poly}(n))}$

- Shortest Path
- Bipartite Matching
- Maximum flow
- Minimum-cost flow
- Multi-commodity flow

- Primal Feasibility of MCGF
- Dual Feasibility of MCGF
- Discounted Markov Decision Processes (MDP)

Klee-Minty cubes

Markov Decision Processes?

MCGF

LP

Minimum cost generalized flow

$$\min\langle c, x\rangle : \sum_{e\in\delta^-(i)} \gamma_e x_e - \sum_{e\in\delta^+(i)} x_e = b_i, \forall i \in [n], \quad x \geq 0 \quad \text{(MCGF)}$$

Our Road to solve the MCGF problem

"A Simpler and Faster Strongly Polynomial Algorithm for Generalized Flow Maximization" - Olver, Végh STOC '17, JACM '20. Fastest/Cleanest (combinatorial) **primal feasibility** algorithm

↓

Question: Find a more combinatorial/structured algorithm that solves 2VPI? (Somewhen in 2019)

↓

Discrete Newton Method (DN) is strongly polynomial for dual feasibility- Dadush, Koh, N., Végh '20
First combinatorial **dual feasibility** algorithm.

Question: Combine combinatorial **primal feasibility** and **dual feasibility** algorithms to tackle **optimization MCGF** problem?

↓

No progress :(

Question: IPM are usually most efficient methods for LP. Is there an IPM with running time $f(m, n)$ ...even Simplex does it.

↓

Allamigeon, Dadush, Loho, N., Végh '22 : Yes, $f(m, n) = 2^{O(m)}$.
"IPM are not worse than Simplex"

↓

Question: Are IPM strongly polynomial for MCGF?

→ Yes :)

Circuits

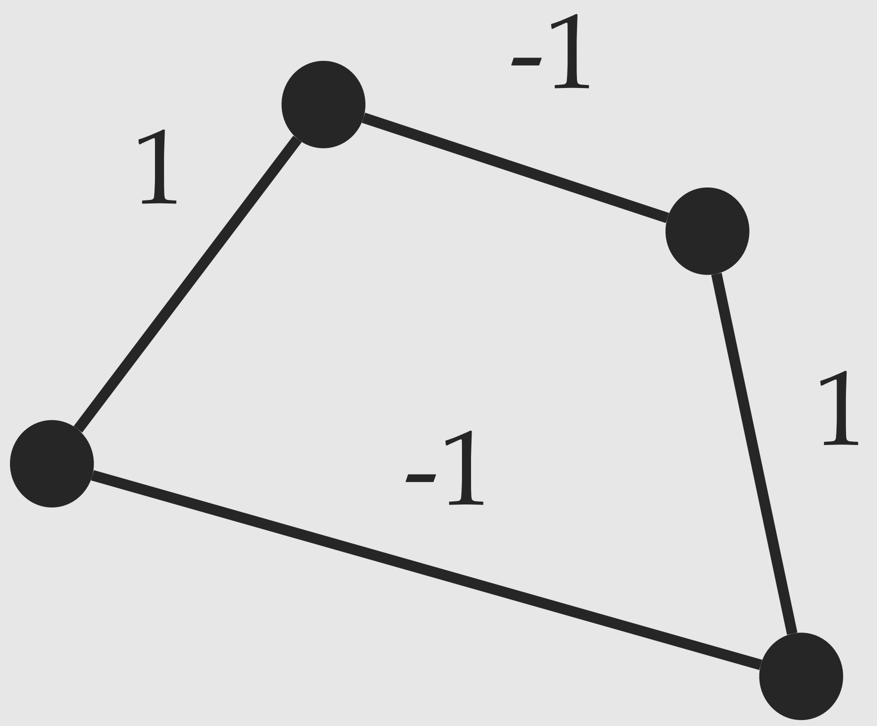
… of linear subspaces…

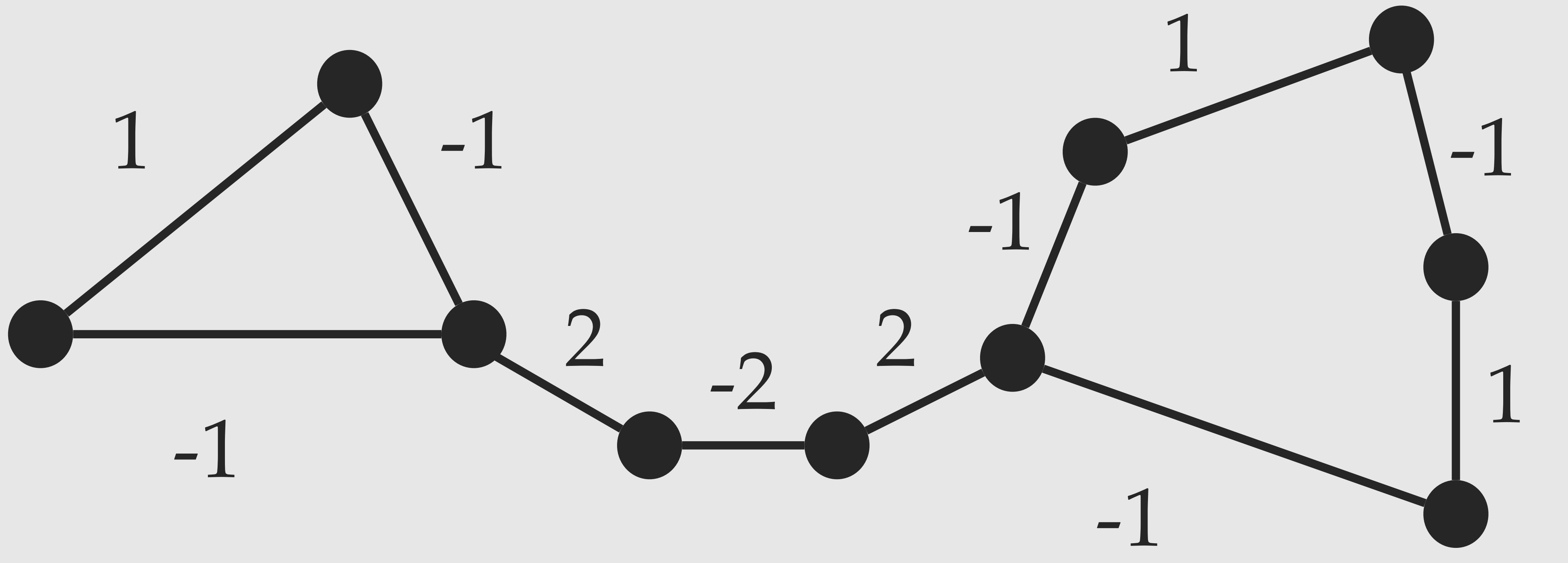
Circuits in simple graphs

Circuits in general are vectors x s.t. $Ax = 0$ and $\nexists y \neq \mathbf{0} : Ay = 0,\ \mathrm{supp}(y) \subsetneq \mathrm{supp}(x)$

Circuits in undirected graphs

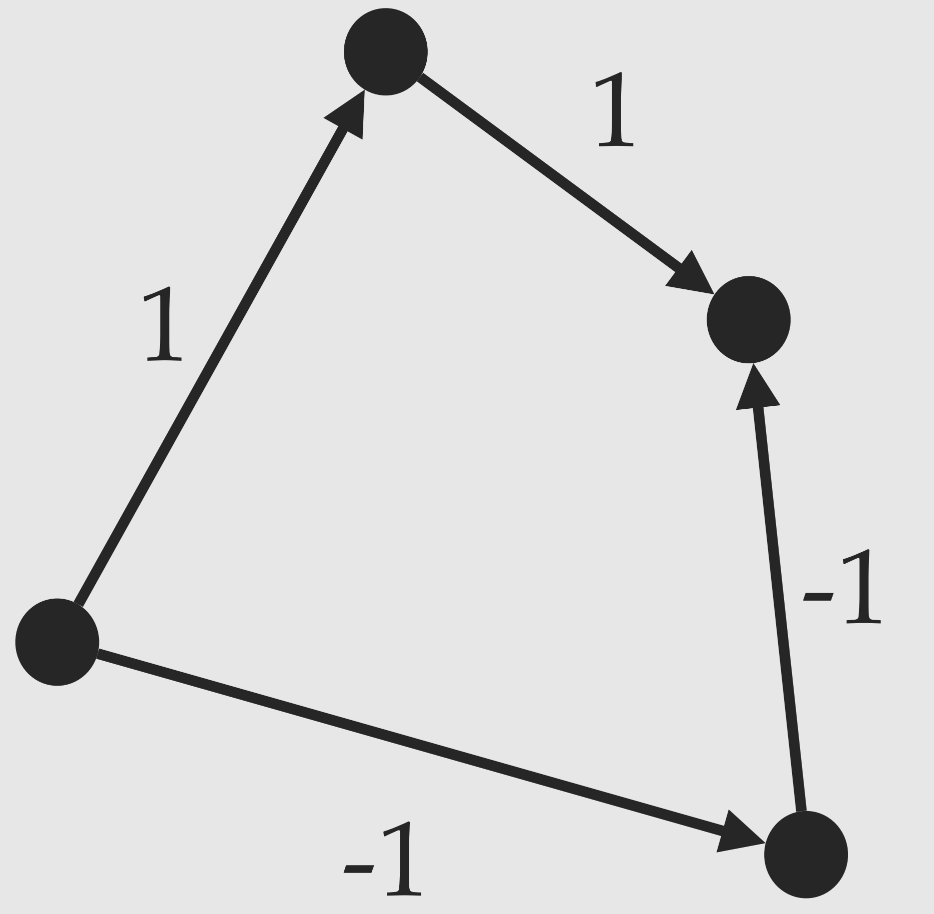
$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$





Circuits in directed graphs

$$A = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & -1 \end{bmatrix}$$



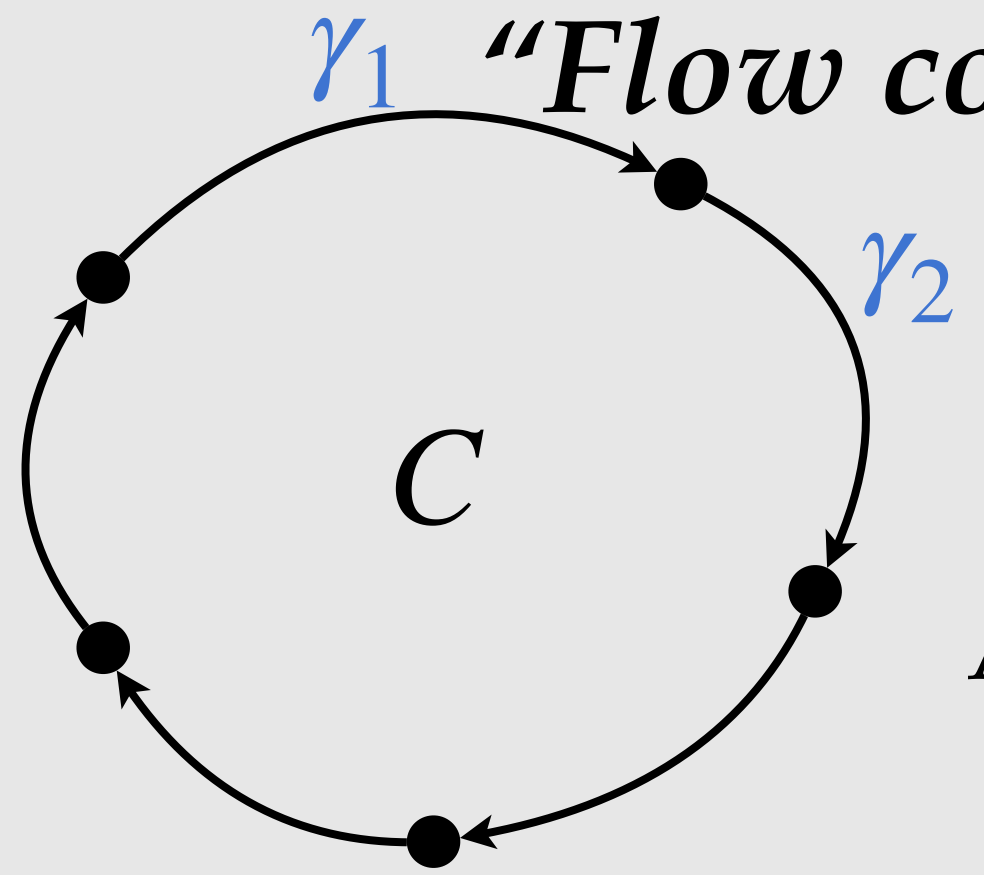
Circuits in generalized flows

LP: $\min\langle c, x\rangle : Ax = b, x \geq 0,\ m$ variables, constraints

Circuits in general are vectors x s.t. $Ax = 0$ and $\nexists y \neq \mathbf{0} : Ay = 0, \text{supp}(y) \subsetneq \text{supp}(x)$

For generalized flow: support-minimal vectors x such that $\sum_{e\in\delta^-(i)} \gamma_i x_e - \sum_{e\in\delta^+(i)} x_e = 0, \forall i \in [n]$

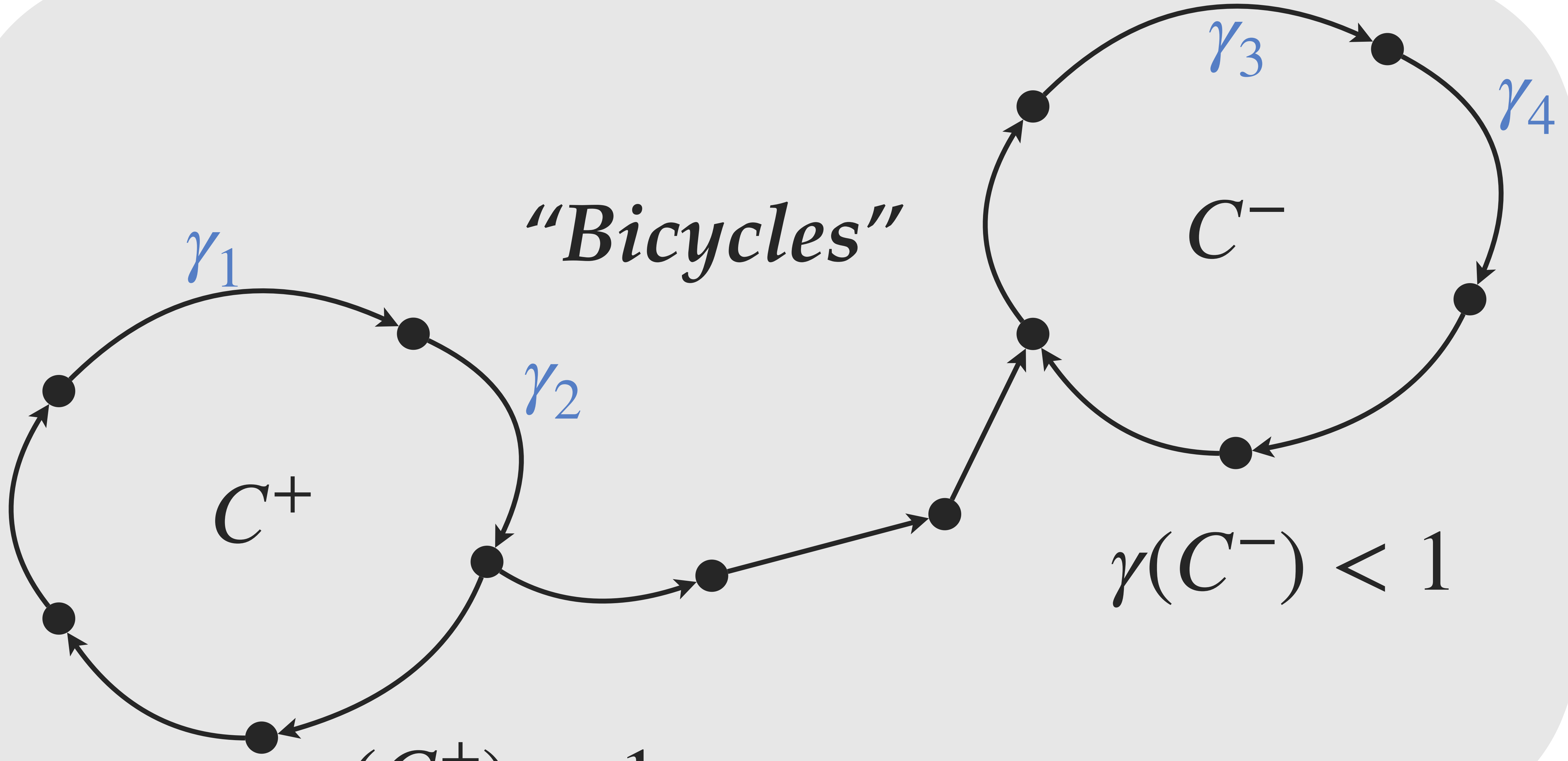
"Flow conserving cycle"



A cycle is a circuit **iff** $\gamma(C) = 1$

$\gamma(C) := \prod_{e\in E(C)} \gamma_e$

"Bicycles"

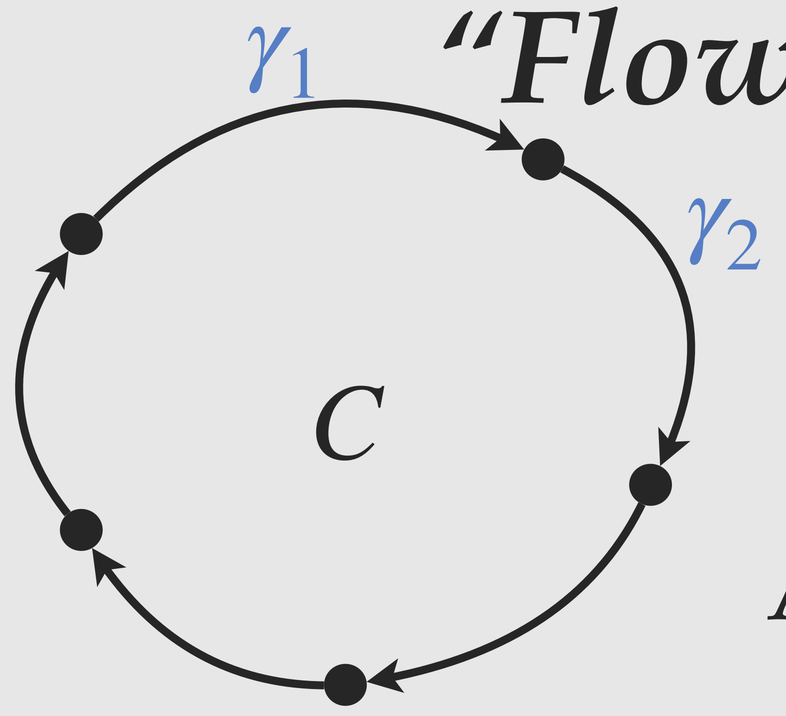


$\gamma(C^-) < 1$

$\gamma(C^+) > 1$

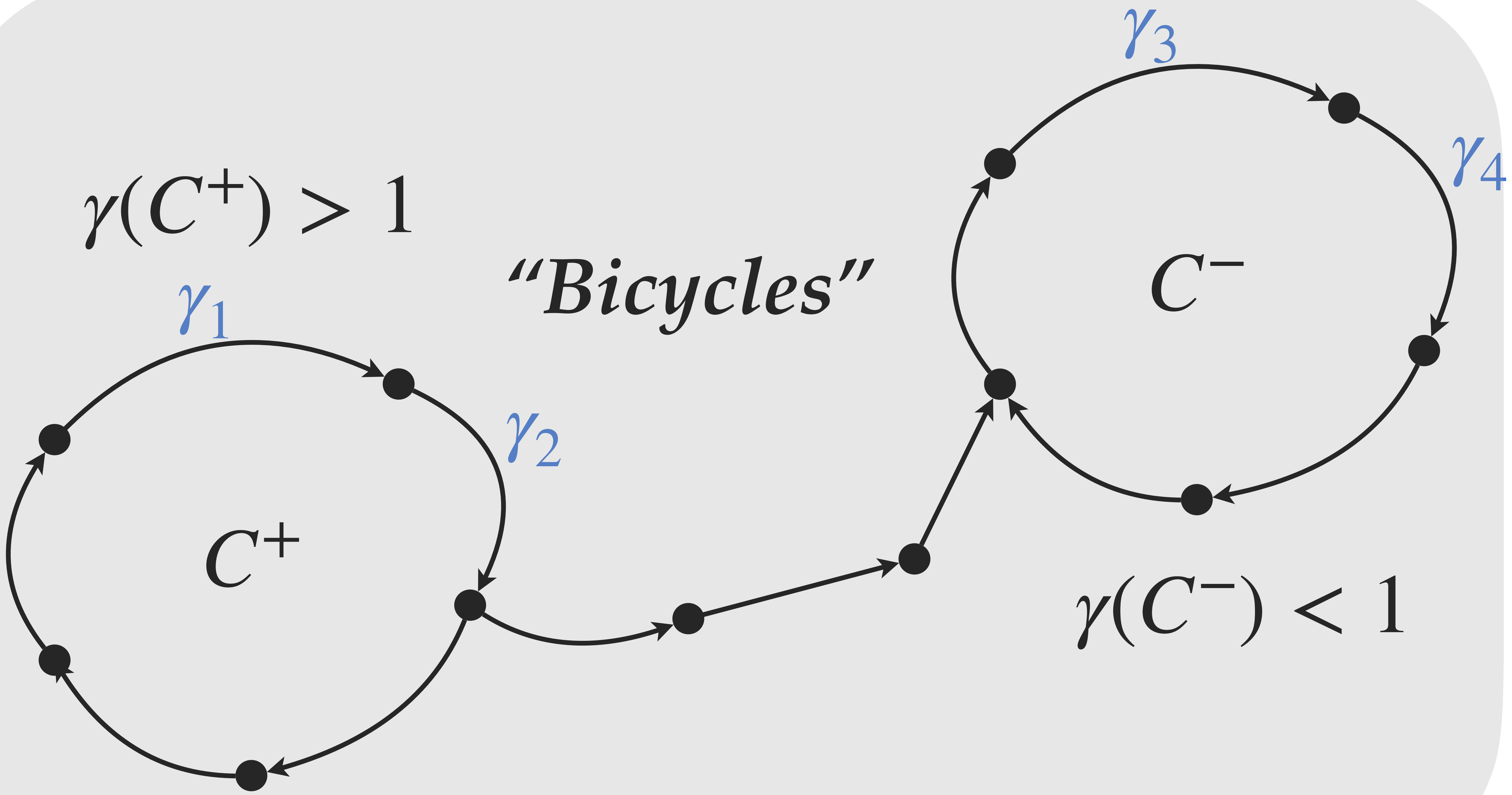
Small circuit cover for MCGF

"Flow conserving cycle"



A cycle is a circuit **iff** $\gamma(C) = 1$

$$\gamma(C) := \prod_{e \in E(C)} \gamma_e$$

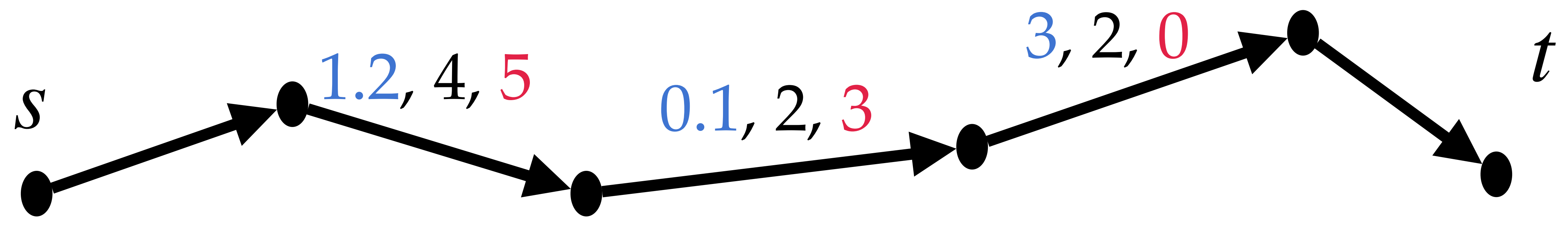


Theorem (Dadush, Koh, N., Olver, Végh '24+):
In the extended residual graph induced by the optimal solution x^*, there exists a collection of $O(mn)$ *bicycles* and *flow conserving cycles* that *dominates* all other bicycles and flow conserving cycles.

Path covers

Combinatorial problem: Given a directed graph $G = (V, E)$ where edges have

gains capacities cost



Question 1: Is there an s-t walk W of length $\leq n$ such that

- gain(W) := $\prod_{e \in W} \gamma(e)$ is **maximum**
- capacity(W) := flow sent to t without violating capacities is **maximum**
- cost(W) := cost per unit of flow sent to t is **minimum ?**

No!

Question 2: Is there a collection $\mathscr{W}, |\mathscr{W}| = \text{poly}(m)$ of s-t walks W of length $\leq n$ such that for any s-t walk W of length $\leq n$ there exists $W^* \in \mathscr{W}$ s.t. $(\text{gain}(W), \text{capacity}(W), 1/\text{cost}(W)) \leq (\text{gain}(W^*), \text{capacity}(W^*), 1/\text{cost}(W^*))$?

No!

Question 3: Is there a collection $\mathscr{W}, |\mathscr{W}| = \text{poly}(m)$ of s-t walks W of length $\leq \text{poly}(m)$ such that for any s-t walk W of length $\leq n$ there exists $W^* \in \mathscr{W}$ s.t. $(\text{gain}(W), \text{capacity}(W), 1/\text{cost}(W)) \leq \text{poly(m)}\ (\text{gain}(W^*), \text{capacity}(W^*), 1/\text{cost}(W^*))$?

Yes!

Our result

Question 3: Is there a collection $\mathscr{W}, |\mathscr{W}| = \text{poly}(m)$ of s-t walks W of length $\leq \text{poly}(m)$ such that for any s-t walk W of length $\leq n$ there exists $W^* \in \mathscr{W}$ s.t. $(\text{gain}(W), \text{capacity}(W), 1/\text{cost}(W)) \leq \text{poly(m)}\ (\text{gain}(W^*), \text{capacity}(W^*), 1/\text{cost}(W^*))$?

Yes!

$\Downarrow$ …a lot of extra effort…

Theorem (Dadush, Koh, N., Olver, Végh '24+):
For every edge $e \in E(G)$ we have that , $\text{SLC}(x_e^{\text{m}}) = O(mn \log(mn))$.

$+$

Theorem (Allamigeon, Dadush, Loho, N., Végh '22):
Given a suitable initial point, there exists an IPM that solves an LP in strongly polynomial many iterations if for all variables $i \in [m]$ we have that $\text{SLC}(x_i^{\text{m}}) = O(\text{poly}(m, n))$.

$=$

Initialized algorithm with strongly polynomially many iterations for minimum cost generalized flow

Initialization

...usually an afterthought...

Why standard initialization techniques have a hard time

Primal: $\min\langle c, x\rangle : Ax = b, x \geq 0, \quad A \in \mathbb{R}^{n\times m}$ Dual: $\max\langle y, b\rangle : A^\top y \leq c$

Approach 1: A large bounding box around the feasible region

Problem of Approaches 1: *How large has the box to be chosen? The computation model does not allow to access the bit complexity of the numbers in the input.*

Approach 2: Homogeneous self-dual initialization (Ye-Todd-Mizuno'94)

$$\begin{array}{llllll}
\min & & & & (n+1)\theta & \\
\text{s.t.} & & +Ax & -b\tau & +\bar{b}\theta & = \mathbf{0}, \\
& -A^\top y & & +c\tau & -\bar{c}\theta & \geq \mathbf{0}, \\
& b^\top y & -c^\top x & & +\bar{z}\theta & \geq \mathbf{0}, \\
& -\bar{b}^\top y & +\bar{c}^\top x & -\bar{z}\tau & & = -(n+1), \\
& y \text{ free}, & x \geq \mathbf{0} & \tau \geq 0, & \theta \text{ free}. &
\end{array}$$

Theorem (Ye-Todd-Mizuno '94):
The system on the left can be initialized on the central path and its optimal solution is exactly the optimal solution of the original system

Problem of Approaches 1 + 2: *The introduction of new constraints and variables modifies the matrix structure so that the systems does not have 2 nonzero entries per column anymore.*

Multistage initialization

Primal: $\min\langle c, x\rangle : Ax = b, x \geq \mathbf{0}, \quad A \in \mathbb{R}^{n\times m}$ Dual: $\max\langle y, b\rangle : A^\top y \leq c$

Stage 1: Conic feasibility

$\text{Solve} : \min\langle \mathbf{1}, \bar{x}\rangle : Ax - A\bar{x} = \mathbf{0}, \mathbf{0} \leq x \leq \mathbf{1}, \bar{x} \geq \mathbf{0}$

$\Rightarrow$ obtain x^* such that $x^* > \mathbf{0}$ and $Ax^* = \mathbf{0}$

Stage 2: Dual feasibility

$\text{Solve} : \min\langle c, x\rangle : Ax = \mathbf{0}, \mathbf{0} \leq x \leq \mathbf{1}$. Initialize with x^*

$\text{Dual} : \min\langle \mathbf{1}, z\rangle : A^\top y - z \leq c, z \geq \mathbf{0}$

$\Rightarrow$ the set of dual solutions with objective value 0 correponds to feasible solution

$\Rightarrow$ obtain y^* as solution near the analytic center of the original dual system.

Stage 3: Primal-dual optimization: Use y^* to initialize the original system.

Theorem: (Allamigeon, Dadush, Loho, N., Végh '22):
There exists an IPM that finds an optimal solution x^* to an LP in strongly polynomial time iff for all variables $i \in [m]$ we have that $\mathrm{SLC}(x_i^{\mathbf{m}}) = O(\mathrm{poly}(m, n))$.
Furthermore, x^* is near the analytic center of the optimal facet.

Note: In all stages the modification of the constraint matrix is "harmless".

Future theory directions

- Combinatorial strongly polynomial time algorithm for minimum-cost generalized flow? With improved running time?
- What is the true cost of making weakly polynomial algorithms strongly polynomial?
- How hard are Markov Decision Processes (MDP)?
- Why do IPM perform so well in practice?
- Universal exact methods for more general convex problems? Convex quadratic?

Strongly poly for general LP?

Thank you!