Layered-Least-Squares Interior Point Methods

a combinatorial optimization perspective

László Végh
London School of Economics

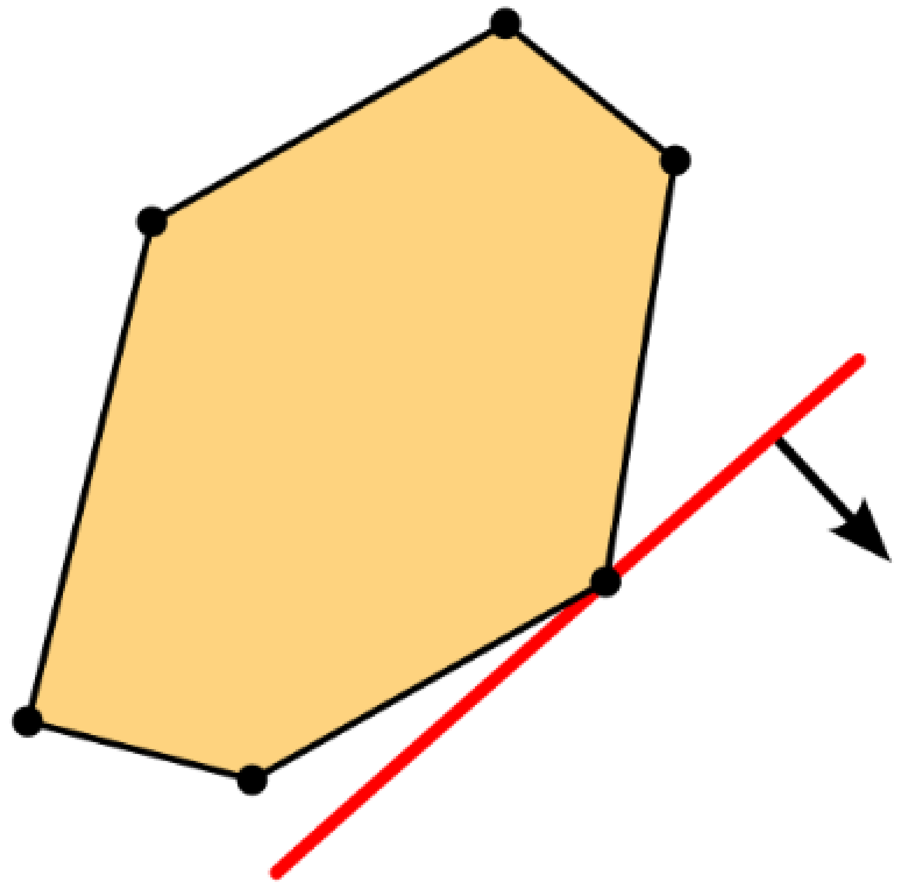
joint work with
Daniel Dadush, Sophie Huiberts (CWI),
and
Bento Natura (LSE)

Duke Algorithms Seminar, 6 Nov 2020

Linear Programming

In standard form for $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n$,

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



Timeline

?

Strongly polynomial algorithm for LP

Smale's 9th question

1980s

Interior point methods

Karmarkar

1970s

Ellipsoid Method

Khachiyan

1940s

Simplex Method

Dantzig

1820s

Origins

Fourier

Weakly vs Strongly Polynomial Algorithms for LP

LP with n variables, m constraints

L: encoding length of the input.

weakly polynomial

- $\text{poly}(m, n, L)$ basic arithmetic operations.
- Standard variants of Ellipsoid and interior point methods: running time bound heavily relies on L.

strongly polynomial

- $\text{poly}(m, n)$ basic arithmetic operations.
- PSPACE: all numbers occurring in the algorithm must remain polynomially bounded in input size.

Strongly Polynomial Algorithms for LP

Network flow problems

- Maximum flow: Edmonds–Karp–Dinitz '70-72
- Min-cost flow: Tardos '85

Special classes of LP

- Feasibility of 2-variable-per-inequality systems: Megiddo '83
- Discounted Markov Decision Processes: Ye '05, Ye '11
- Maximum generalized flow problem: V. '17, Olver–V. '20
- ...

Dependence on the constraint matrix only

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

Running time dependent only on constraint matrix A, **but not on** b and c.

General LP

- 'Combinatorial LPs'
  If A integral and $|\det(B)| \leq \Delta$ for all square submatrices of A, then LP solvable in $\mathrm{poly}(m, n, \log \Delta)$ arithmetic operations: Tardos '86
- 'Layered-least-squares (LLS) Interior Point Method'
  LP solvable in $O(n^{3.5} \log \bar{\chi}_A)$ linear system solves: Vavasis–Ye '96

Dependence on the constraint matrix only

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

Running time dependent only on constraint matrix A, **but not on** b and c.

'Layered-least-squares (LLS) Interior Point Method'
LP solvable in $O(n^{3.5} \log \bar{\chi}_A)$ linear system solves: Vavasis–Ye '96

Condition number $\bar{\chi}_A$

- $\bar{\chi}_A = O(2^{L_A})$.
- Governs the stability of layered-least-squares solutions.
- Depends only on the subspace $\ker(A)$.
- NP-hard to approximate within a factor $2^{\mathrm{poly}(\mathrm{rank}(A))}$: Tunçel '99

Scale Invariance

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

$A \in \mathbb{R}^{m \times n}$. Let $\mathbf{D}$ denote the set of $n \times n$ positive diagonal matrices.

- **Diagonal rescaling (LP') of (LP):** Replace $A' = AD, c' = Dc, b' = b$ for some $D \in \mathbf{D}$.
- **Central path is invariant under rescaling**: central path of (LP) mapped to that of (LP') by $(x(\mu), s(\mu), y(\mu)) \to (D^{-1}x(\mu), Ds(\mu), y(\mu))$.
- Standard interior point methods are **invariant** under rescaling.
- The Vavasis–Ye (VY) algorithm is **not scaling invariant**:
  LLS steps solve linear system indexed by the current variable ordering $x_{\pi[1]} > \cdots > x_{\pi[n]}$ and the ratios $x_{\pi[i]}/x_{\pi[i+1]}, i \in [n-1]$.
- VY Algorithm uses $O(n^{3.5} \log \bar{\chi}_A)$ LLS steps [VY96].

Is there a scaling invariant LLS interior point method?

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

- A scaling invariant version of LLS would automatically give $O(n^{3.5} \log \bar{\chi}_A^*)$ iterations instead of $O(n^{3.5} \log \bar{\chi}_A)$ iterations.
- $\bar{\chi}_A^*$ can be arbitrarily smaller than $\bar{\chi}_A$.

Prior work:

- The (scaling invariant) Mizuno–Todd–Ye Predictor-Corrector algorithm finds an ε-approximate solution in time $O(n^{3.5} \log \bar{\chi}_A^* + n^2 \log\log(1/\varepsilon))$: Monteiro–Tsuchiya '05.
- $O(n^{3.5} \log \bar{\chi}_A^*)$ iterations scaling invariant algorithm, but each iteration depends on the bit-complexity b and c: Lan–Monteiro–Tsuchiya '09.
- Results on the central path curvature from information geometry also suggest that $\bar{\chi}_A^*$ should be the `right' dependence.

Our contributions: Dadush–Huiberts–Natura–V. '20

A scaling invariant algorithm

We give a scaling invariant LLS interior point method, settling the open question posed by Monteiro and Tsuchiya in '03.

Improved Runtime

We show a running time bound of $O(n^{2.5}\log(\bar{\chi}^*_A + n)\log n)$ linear system solves. This is achieved via an improved analysis that also applies to the original VY algorithm.

Finding a nearly-optimal rescaling of A

Given $A \in \mathbb{R}^{m\times n}$, in $O(n^2m^2 + n^3)$ time, we can compute

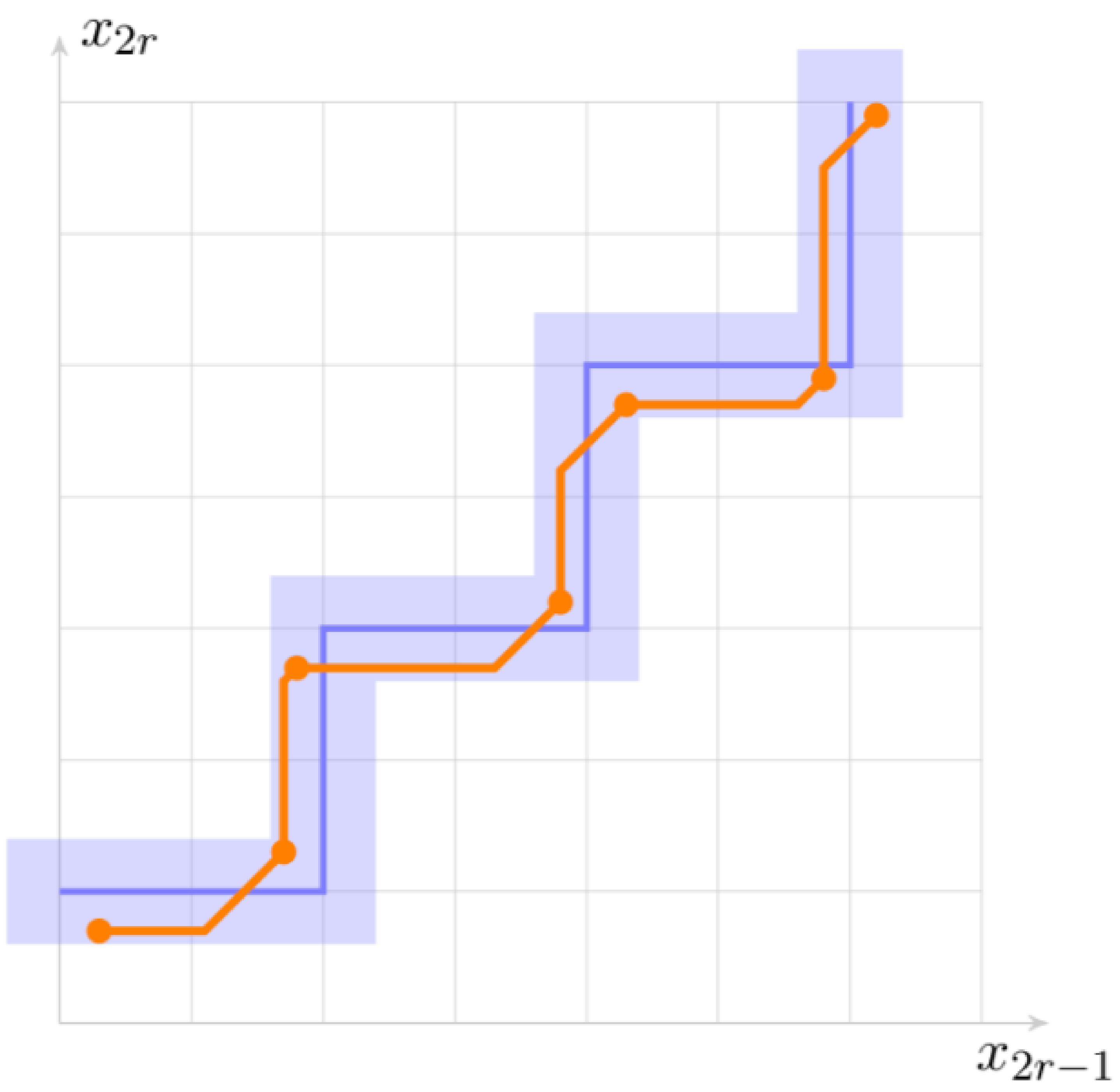
(i) rescaling $D \in \mathbf{D}$ satisfying $\bar{\chi}^*_A \le \bar{\chi}_{AD} \le n(\bar{\chi}^*_A)^3$.

(ii) $t \ge 1$ satisfying $t \le \chi_A \le n(\bar{\chi}^*_A)^2 t$.

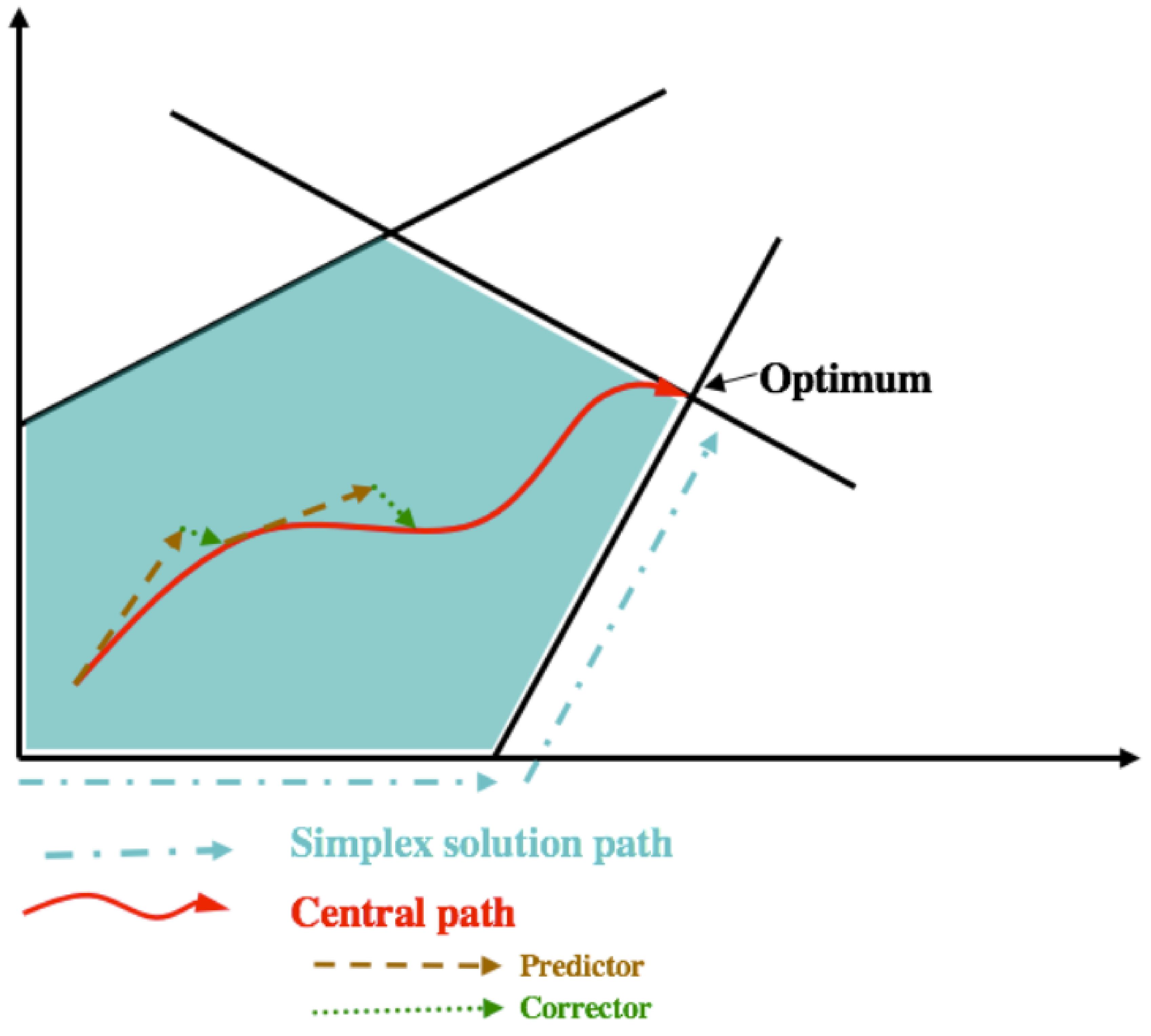
Limitation

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.



A crash course on Interior Point Methods



Complementary Slackness

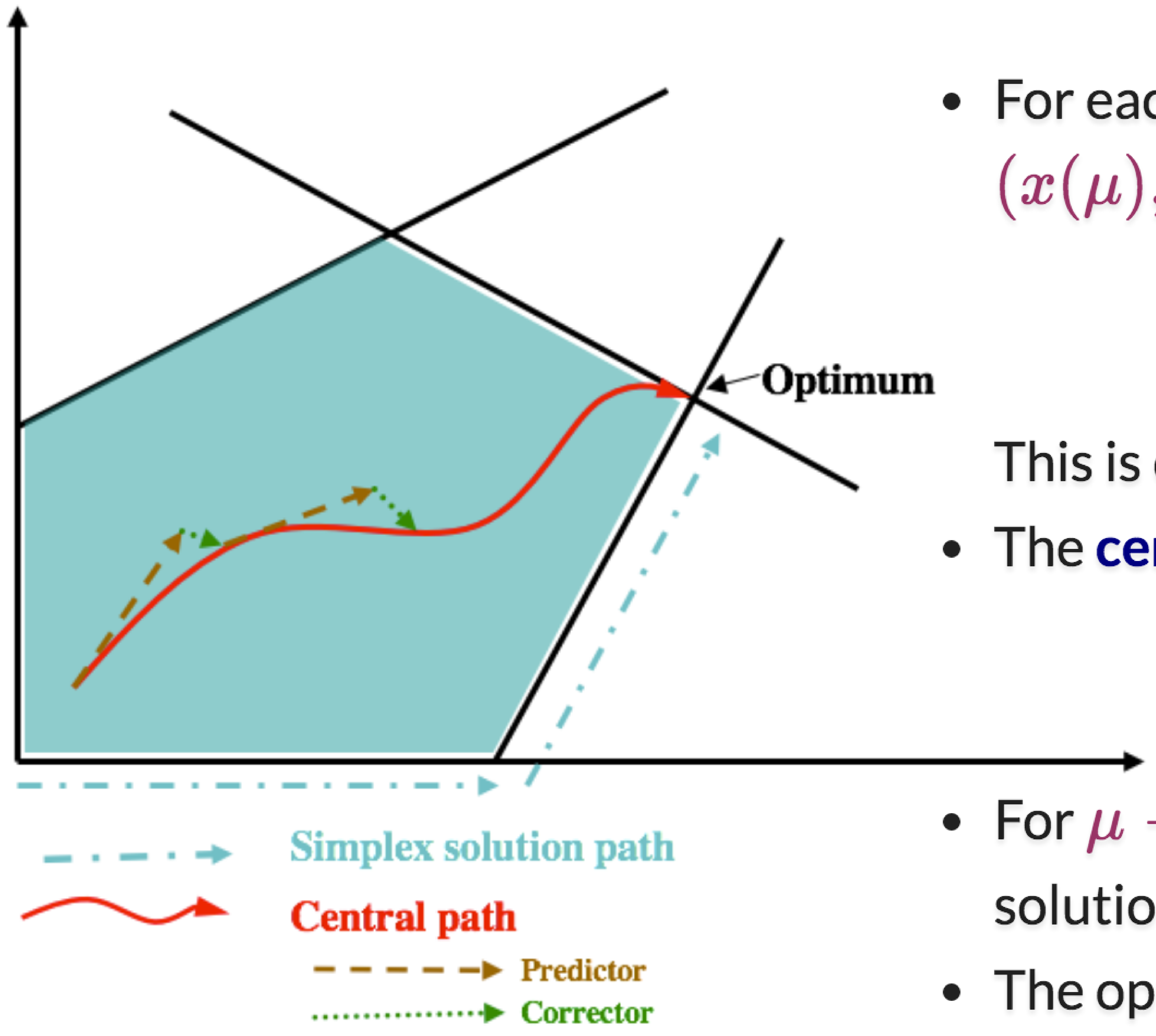
$$\begin{aligned} \min\ c^\top x \\ Ax = b \\ x \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \max\ y^\top b \\ A^\top y + s = c \\ s \geq 0 \end{aligned}$$

A feasible pair $(x, s) \geq 0$ is optimal if and only if $x^\top s = 0 \Leftrightarrow x_i = 0$ or $s_i = 0, \forall i \in [n]$.

Weak duality

$$c^\top x = (y^\top A + s^\top)x = y^\top b + s^\top x \geq y^\top b.$$

The central path



- For each $\mu > 0$, there exists a unique solution $w(\mu) = (x(\mu), y(\mu), s(\mu))$ such that

$$x(\mu)_i s(\mu)_i = \mu \quad \forall i \in [n]$$

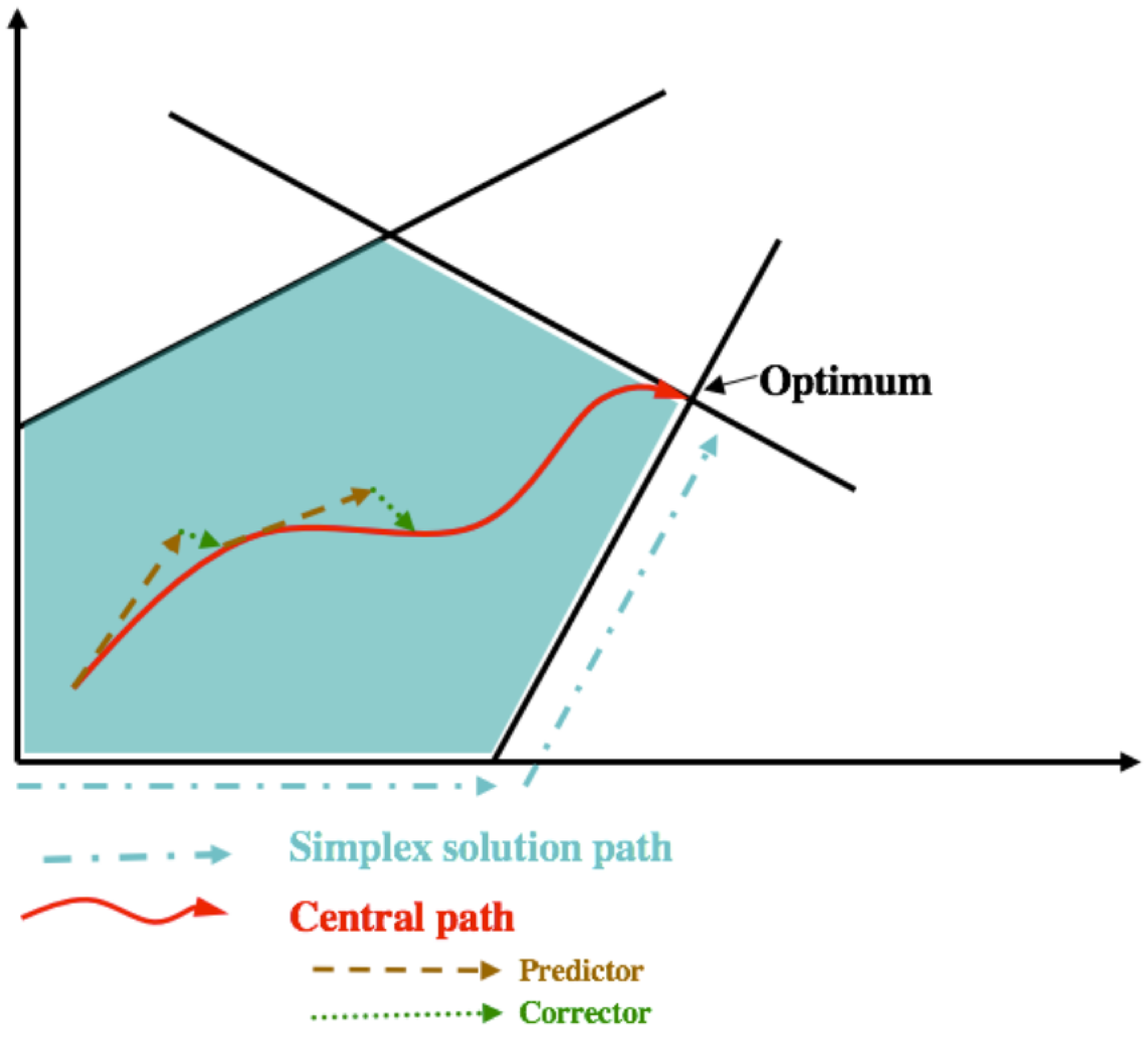
  This is called the **central path element** for μ.
- The **central path** is the algebraic curve formed by

$$\{w(\mu) : \mu > 0\}$$

- For $\mu \to 0$, the central path converges to an optimal solution $w^* = (x^*, y^*, s^*)$.
- The optimality gap is $s(\mu)^\top x(\mu) = n\mu$.

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 two types of steps:
- **Predictor steps:** 'shoot down' the central path, decreasing μ by a factor at least $1 - \beta/\sqrt{n}$. May move slightly 'farther' from the central path.
- **Corrector steps:** do not change parameter μ, but move back 'closer' to the central path.
- Within $O(\sqrt{n})$ iteration, μ decreases by a factor 2.



Predictor Newton step

We obtain the step direction $\Delta w = (\Delta x, \Delta y, \Delta s)$ as solutions to

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

and

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

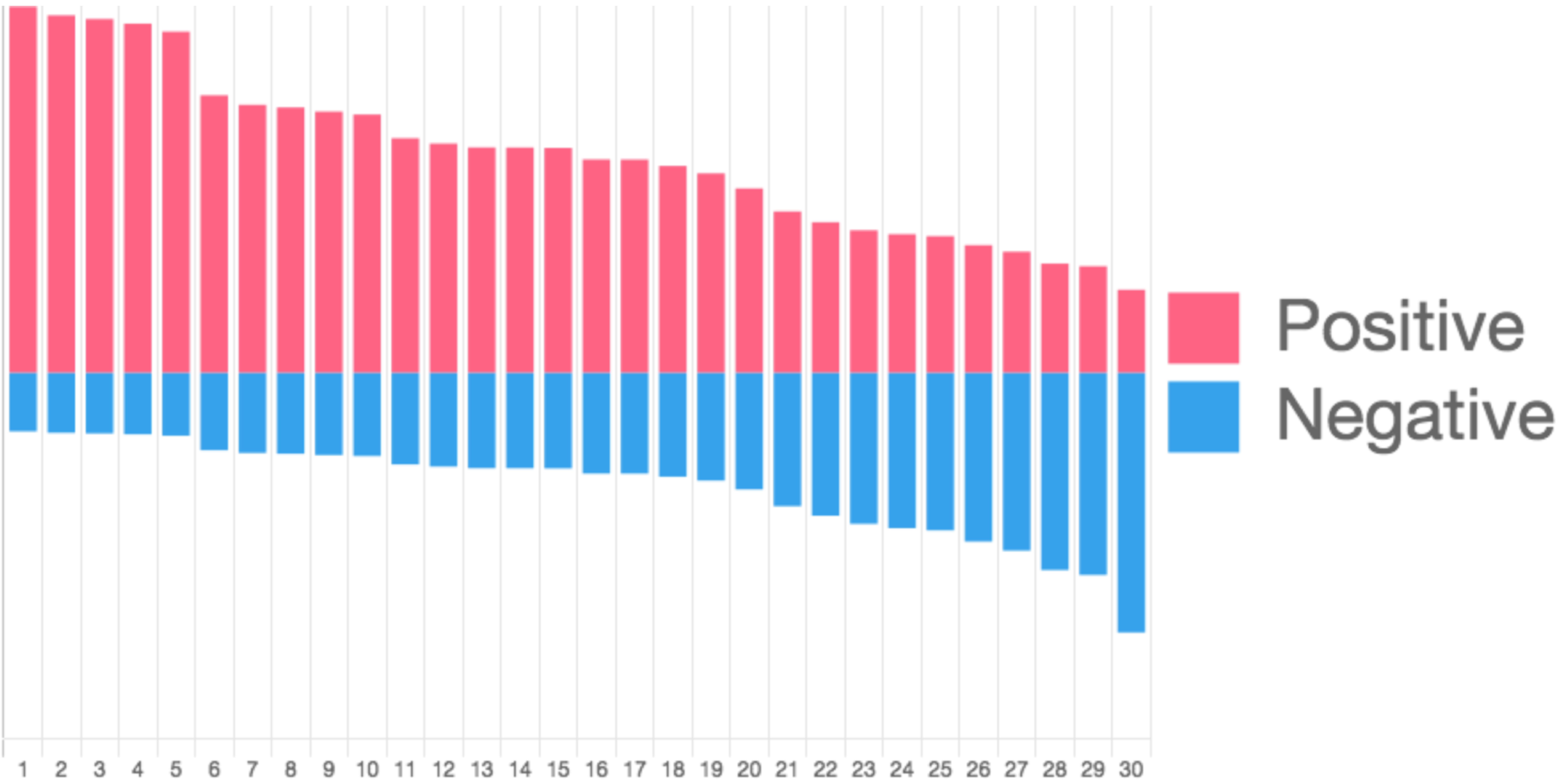
Next iterate is obtained as

$$w' = w + \alpha \Delta w = (x + \alpha \Delta x, y + \alpha \Delta y, s + \alpha \Delta s)$$

for some $\alpha \in [0, 1]$. Set α large as possible subject to not moving too far away from the path.

Changes of x_i and s_i variables in the MTY Predictor-Corrector algorithm

Press key "i" to run iteration



Recent weakly polynomial IPM successes

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

- Randomized $\tilde{O}(\sqrt{m}(\mathrm{nnz}(A) + m^2)L)$ algorithm, where nnz is the number of non-zeros: Lee–Sidford '14-15
- Randomized algorithm in 'current' matrix multiplication time $\tilde{O}(n^\omega L)$, $\omega \approx 2.37$: Cohen–Lee–Song '18
- Deterministic algorithm with same runtime: van den Brand '20
- $\tilde{O}((mn + m^3)L)$: vdBLSS '20

For special problems:

- $\tilde{O}(m^{3/2}\log^2(U/\varepsilon))$ algorithm for an additive ε approximation for lossy generalized flow problems: Daitch–Spielman '08
- $\mathcal{O}(m^{10/7})$ algorithm for max s-t flow and min s-t cut problems in directed graphs with unit capacities: Mądry '13

The mysterious $\bar{\chi}_A$

through a matroidal lens

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

Definition.

$$\bar{\chi}_A := \sup\left\{\|A^\top \left(ADA^\top\right)^{-1} AD\| \ : D \in \mathbf{D}\right\}$$

Introduced by Dikin '67, Stewart '89, Todd '90, ...

A convenient characterization: lifting cost

$\bar{\chi}_A$ is the minimum number $M \geq 1$ such that for all $I \subseteq [n]$ and $x \in \pi_I(\ker(A))$, there exists $\hat{x} \in \ker(A)$ satisfying $\hat{x}_I = x$ and $\|\hat{x}\| \leq M\|x\|$.

Ex: If $(1, 2, *, *) \in \ker(A)$, $\exists (1, 2, x, y) \in \ker(A)$ with $\|(1, 2, x, y)\| \leq M\|(1, 2)\|$.

Properties of $\bar{\chi}_A$

$$\bar{\chi}_A = \sup\left\{\left\|A^\top\left(ADA^\top\right)^{-1}AD\right\| : D \in \mathbf{D}\right\}$$

We also use $\bar{\chi}_W = \bar{\chi}_A$ for the subspace $W = \ker(A)$.

Lemma. *The following hold:*

(i) *If $A \in \mathbb{Z}^{n\times m}$ then $\bar{\chi}_A$ is bounded by $2^{O(L_A)}$, where L_A is the input bit length of A.*

(ii) $\bar{\chi}_A = \max\left\{\|B^{-1}A\| : B \text{ non-singular } m \times m \text{ submatrix of } A\right\}.$

(iii) $\bar{\chi}_W = \bar{\chi}_{W^\perp}.$

Application: Final rounding step in standard IPMs

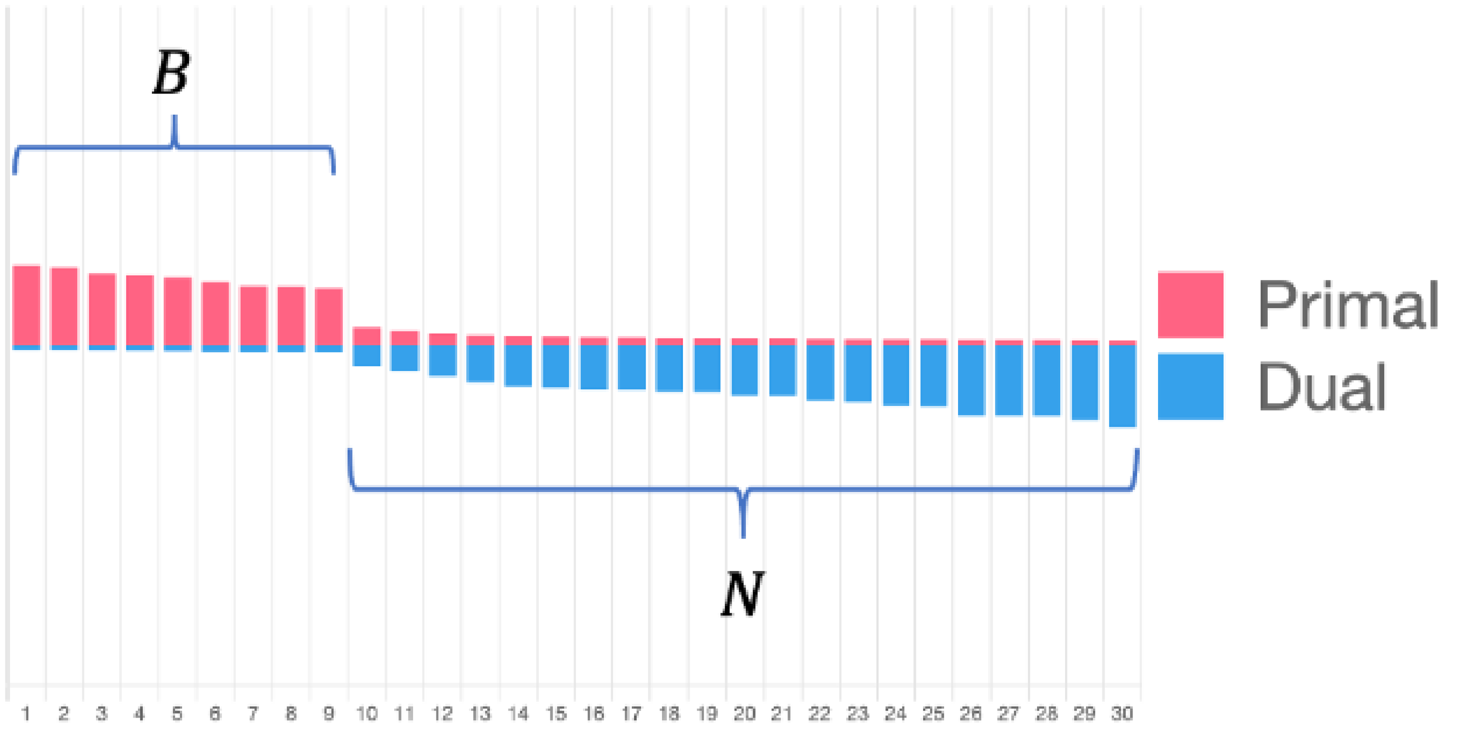
$$\begin{aligned} \min\ c^\top x \qquad\qquad & \max\ y^\top b \\ Ax = b \qquad\qquad & A^\top y + s = c \\ x \geq 0 \qquad\qquad & s \geq 0 \end{aligned}$$

- There exists a partition $[n] = B^* \cup N^*$ and optimal (x^*, y^*, s^*) such that $B^* = \mathrm{supp}(x^*),\ N^* = \mathrm{supp}(s^*)$.
- Given $w = (x, y, s)$ close to the central path with 'small enough' $\mu = x^\top s/n$, such that for each $i \in [n]$, either $x_i \geq M s_i$ or $x_i \leq s_i/M$ for M "very large".

- Assume we correctly 'guess' the optimal partition B^*, N^* via

$$B := \{i : x_i > s_i\}, \quad N := \{i : x_i \leq s_i\}.$$

- We now have $x_i \geq M x_j$ and $M s_i \leq s_j$ for $i \in B$ and $j \in N$.



Application: Final rounding step in standard IPMs

We will move to $\bar{w} = w + \Delta w := (\bar{x}, \bar{s}, \bar{y})$ such that $\mathrm{supp}(\bar{x}) \subseteq B, \mathrm{supp}(\bar{s}) \subseteq N, \bar{x}, \bar{s} \geq 0$.
If we succeed, $\bar{x}$, $\bar{s}$ are optimal since $\bar{x}^\top \bar{s} = 0$.

For this purpose, we will compute $(\Delta x, \Delta s, \Delta y)$ satisfying

$$A\Delta x = 0,\ \Delta x_N = -x_N,\ A^\top \Delta y + \Delta s = 0, s_B = -s_B,$$

together with $x + \Delta x \geq 0,\ s + \Delta s \geq 0$. For this last inequality, it suffices to achieve

$$\|\Delta x_B\| \leq \min_{i \in B} x_i,\ \|\Delta s_N\| \leq \min_{j \in N} s_j.$$

Natural choices are

$$\Delta x := \arg\min \left\{\|u\| : u_N = -x_N, Au = 0\right\},$$
$$\Delta s := \arg\min \left\{\|v\| : v_B = -s_B, A^\top z + v = 0\right\}.$$

For the above choice, we have

$$\|\Delta_B x\| \leq \bar{\chi}_A \|x_N\| \leq \frac{n\bar{\chi}_A}{M} \min_{i \in B} x_B,$$
$$\|\Delta_N s\| \leq \bar{\chi}_A \|s_B\| \leq \frac{n\bar{\chi}_A}{M} \min_{j \in N} s_j.$$

The circuit imbalance measure

...the "combinatorial" sister of $\bar{\chi}_A$

Definition. A **circuit** of A is a minimal linearly dependent subset of columns $C \subseteq [n]$. Let $\mathcal{C}$ denote the set of all circuits.

Definition. The circuit imbalance between $i, j \in [n]$ is

$$\kappa_{ij} := \max \left\{ \left| \frac{g_j}{g_i} \right| : \{i, j\} \subset \operatorname{supp}(g),\ Ag = 0,\ \operatorname{supp}(g) \in \mathcal{C} \right\}$$

The **circuit imbalance measure** of A is $\kappa_A := \max_{i,j\in[n]} \kappa_{ij}$

Lemma. *If A is a TU-matrix, then $\kappa_A = 1$. More generally, if A is integer, then $\kappa_A \leq \Delta$, where Δ is the max. subdeterminant.*

Theorem. *[DHNV20]* $\sqrt{1+\kappa_A^2} \leq \bar{\chi}_A \leq n\kappa_A.$

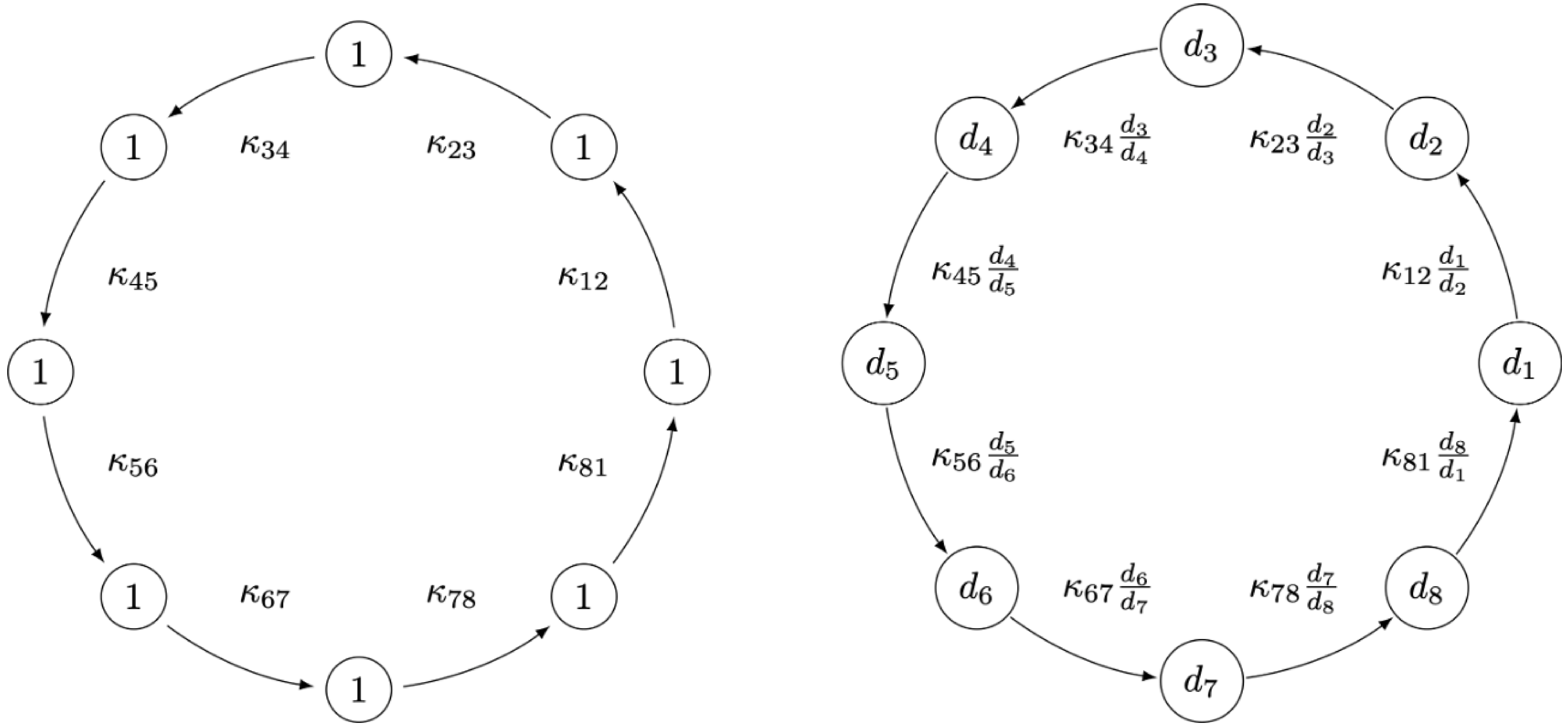
The optimal rescaling κ^*_A

$$\kappa^*_A := \inf\{\kappa_{AD} : D \in \mathbf{D}\}.$$

A near-optimal rescaling D for κ_{AD} will be near-optimal for $\bar{\chi}_{AD}$

Example: Let $\ker(A) = \mathrm{span}((0,1,1,M),(1,0,M,1))$.
Thus, $\{2,3,4\}$ and $\{1,3,4\}$ are circuits, and $\kappa_A \geq M$.
Can we find a D such that κ_{AD} is small?

Cycles are invariant under rescalings



Theorem. *[DHNV20]*

$$\kappa_A^* = \max\left\{ \prod_{(i,j)\in H} \kappa_{ij}^{1/|H|} \,:\, H \text{ is a cycle in the complete graph on } [n] \right\}$$

Approximating κ^*_A

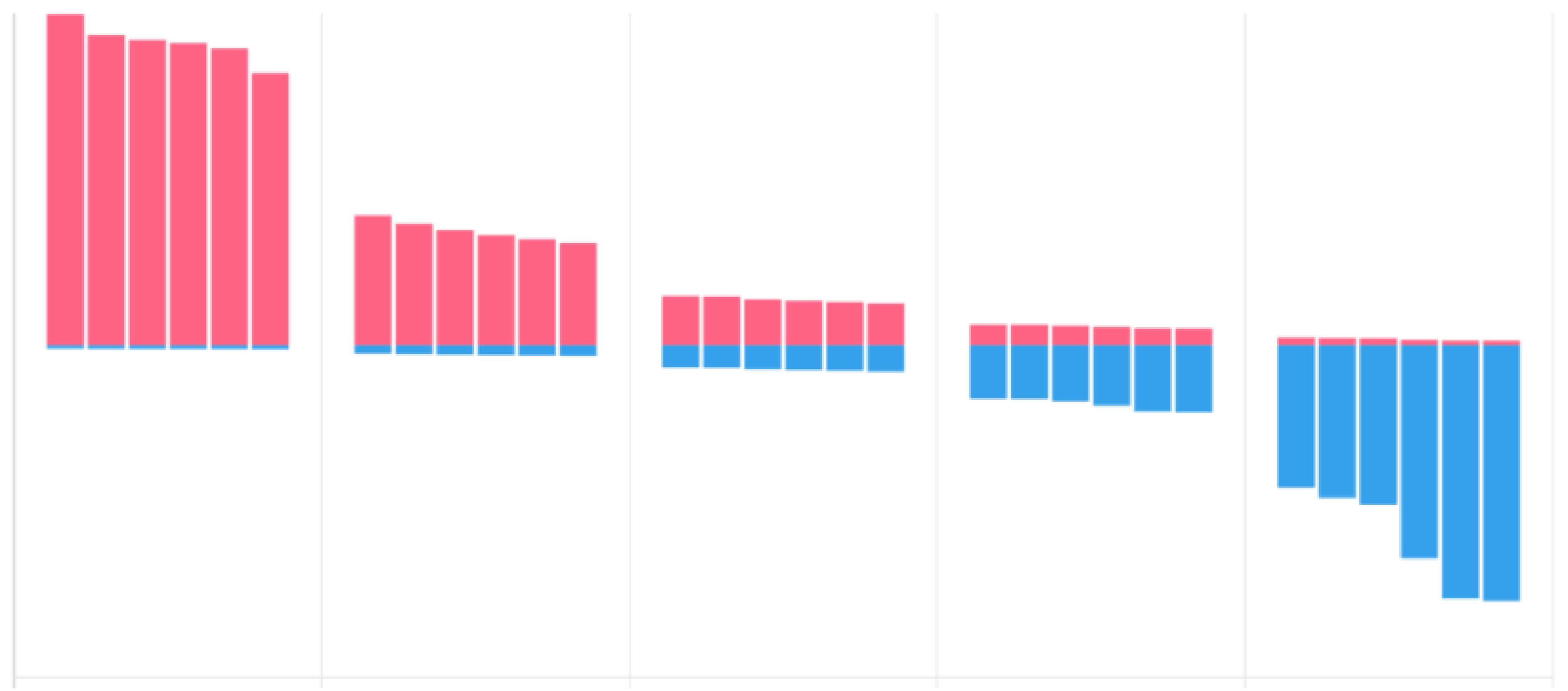
Theorem. *[DHNV20]*

$$\kappa^*_A = \max\left\{ \prod_{(i,j)\in H} \kappa_{ij}^{1/|H|} : H \text{ is a cycle in the complete graph on } [n] \right\}$$

- **Algorithmically,** the optimal rescaling can be obtained via a minimum-mean cycle computation for the costs $c_{ij} = -\log \kappa_{ij}$.
- **Minor caveat:** computing κ_{ij} values is NP-complete
- **Luckily,** for any $g \in \ker(A)$ with $i, j \in \operatorname{supp}(g) \in \mathcal{C}$,

$$|g_j/g_i| \geq \kappa_{ij}/(\kappa^*_A)^2.$$

Layered Least Squares Interior Point Methods

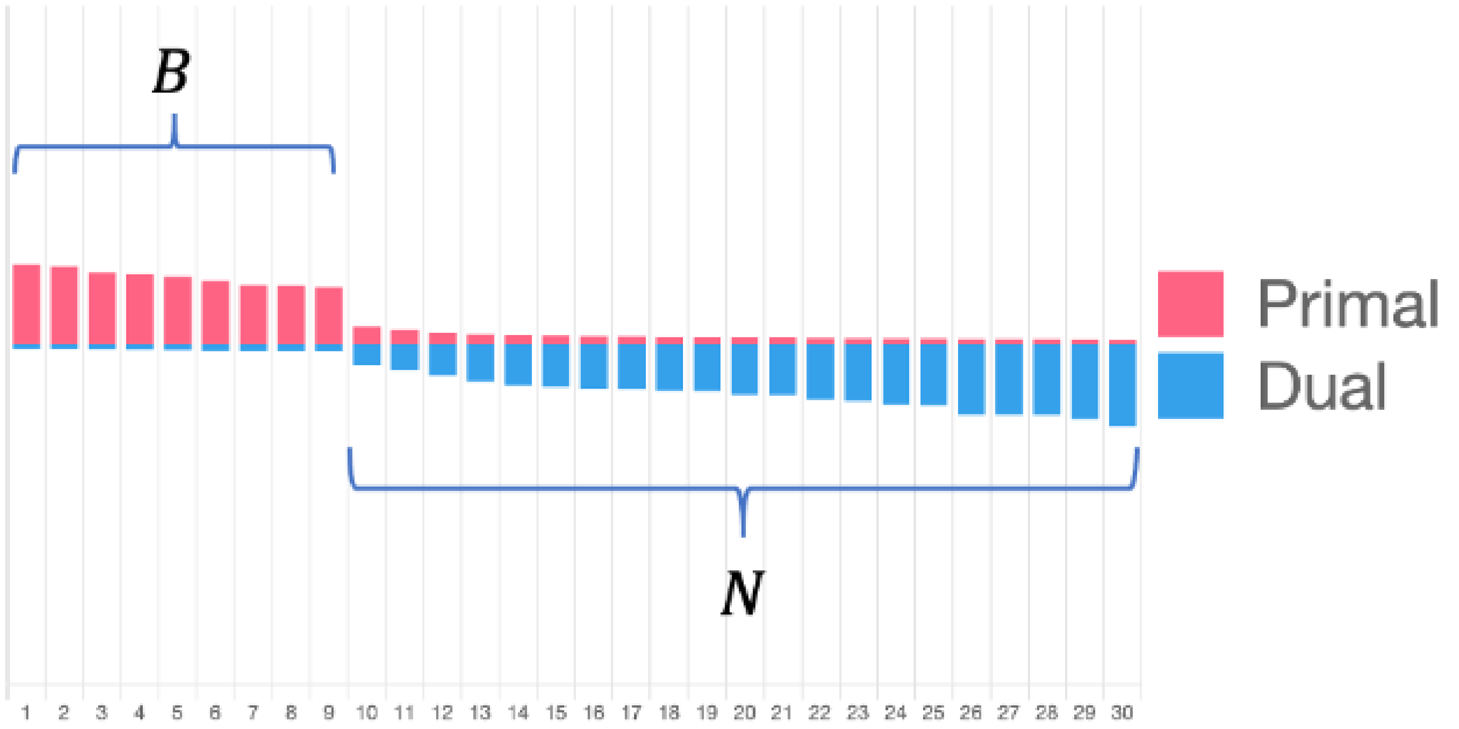


The Vavasis–Ye algorithm

Learning the optimal partition of variables

- Assume the Predictor-Corrector method has already 'found' the partition $[n] = B^* \cup N^*$:
  $x_i \gg x_j$ and $s_i \ll s_j$ if $i \in B^*, j \in N^*$.
- A simple projection would find the optimal solution, but the usual predictor step **does not**:

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



The Vavasis–Ye algorithm

Learning the optimal partition of variables

- **What does the Vavasis–Ye algorithm do here?**
- First, solve

$$\min \sum_{i \in N^*} \left(\frac{x_i + \Delta \bar{x}_i}{x_i} \right)^2 \quad \text{s.t. } A \Delta \bar{x} = 0$$

The N^* components of the optimal $\Delta \bar{x}$ give Δx_{N^*}; in this case, $\Delta x_{N^*} = 0$.

- Next, solve

$$\min \sum_{i \in B^*} \left(\frac{x_i + \Delta \bar{x}_i}{x_i} \right)^2 \quad \text{s.t. } A \Delta \bar{x} = 0,\ \Delta \bar{x}_{N^*} = \Delta x_{N^*}$$

The Vavasis–Ye algorithm

Layering and crossover events in the Vavasis–Ye algorithm

- Arrange the variables into layers $(J_1, J_2, \ldots, J_k)$ as follows:
- Order $x_1 \geq x_2 \geq \ldots \geq x_n$.
- Start a new layer after x_i whenever $x_i > \mathrm{poly}(n)\bar{\chi}_A x_{i+1}$.
- Variables on lower layers 'barely influence' those on higher layers.
- **Not scaling invariant!**

Layering and crossover events in the Vavasis–Ye algorithm

Progress measure

Definition. The variables x_i and x_j **cross over** between μ and μ' for $\mu > \mu' > 0$, if

(i) $\mathrm{poly}(n)(\bar{\chi}_A)^n x_j(\mu) \geq x_i(\mu)$.
(ii) $\mathrm{poly}(n)(\bar{\chi}_A)^n x_j(\mu'') < x_i(\mu'')$ on the central path for $0 < \mu'' < \mu'$.

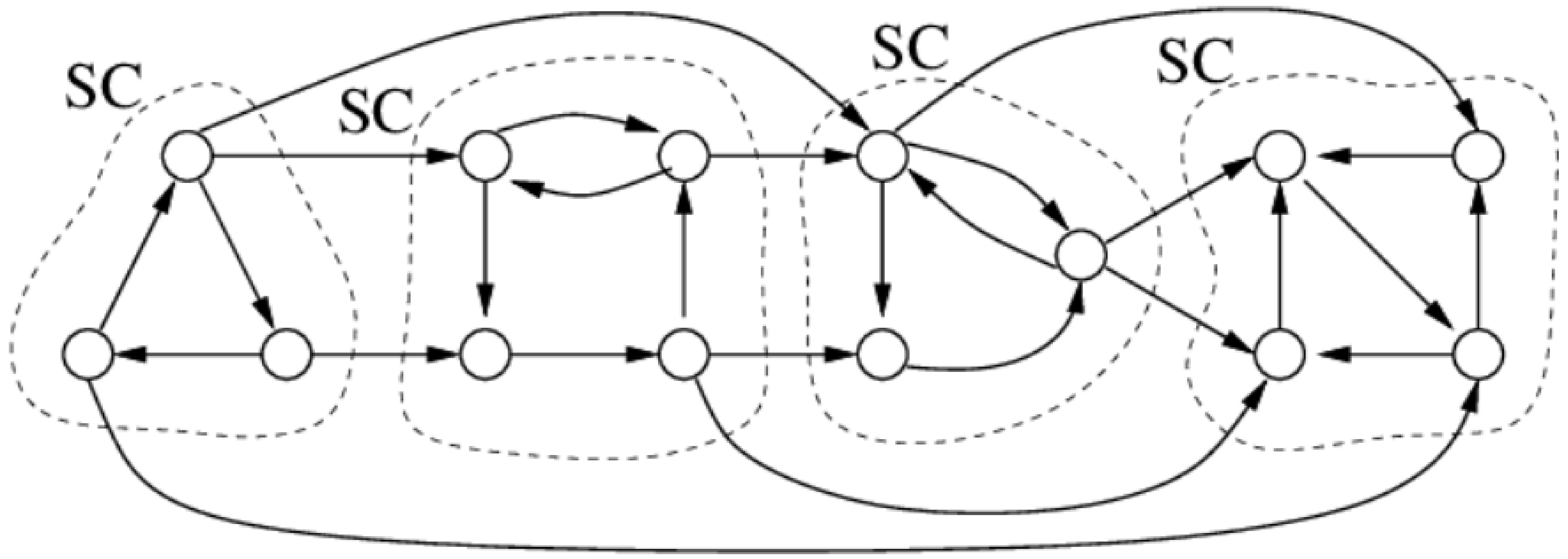
Lemma. *In the Vavasis–Ye algorithm, a crossover event happens within every* $O(n^{1.5} \log \bar{\chi}_A)$ *iterations.*

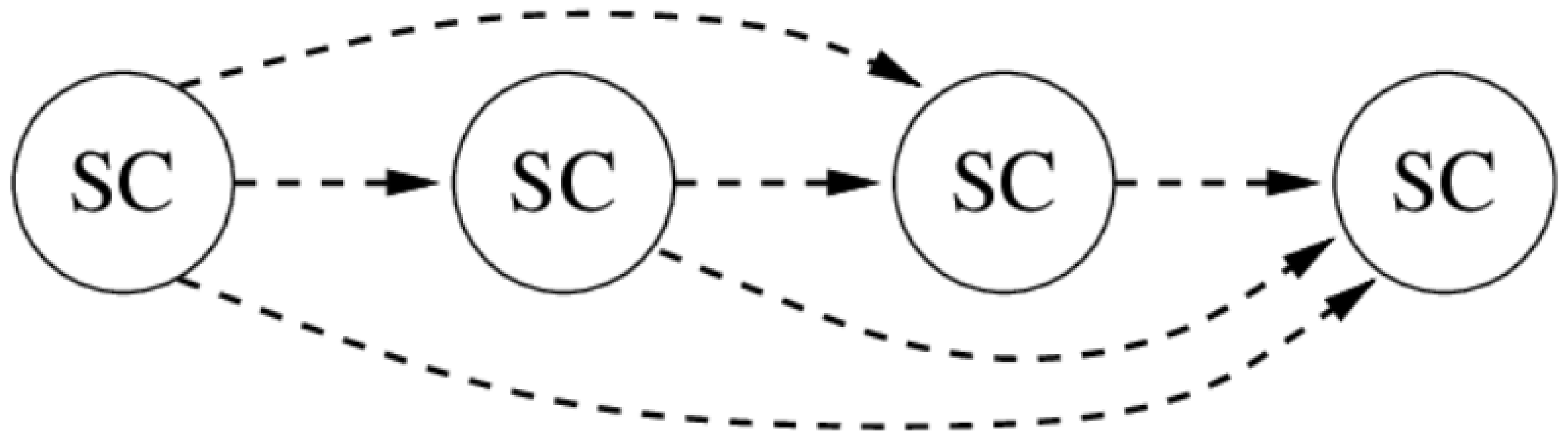
We do not know which two variables cross over!

Scale-invariant layering algorithm [DHNV20]

Instead of x_i/x_j, we look at the scale invariant $\kappa_{ij}x_i/x_j$.

- Layers are identified as the strongly connected components of the digraph formed by the edges (i, j) such that $\kappa_{ij}x_i/x_j \geq 1/\mathrm{poly}(n)$.
- We do not know the κ_{ij} values, but maintain gradually improving lower bounds on them.





Scale-invariant layering algorithm [DHNV20]

Improved convergence analysis

Definition. The variables x_i and x_j **cross over** between μ and μ' for $\mu > \mu' > 0$, if

(i) $\mathrm{poly}(n)(\bar{\chi}^*_A)^n x_j(\mu) \geq \kappa_{ij} x_i(\mu)$.
(ii) $\mathrm{poly}(n)(\bar{\chi}^*_A)^n x_j(\mu'') < \kappa_{ij} x_i(\mu'')$ on the central path for $0 < \mu'' < \mu'$.

- We do not use cross-over events directly, but a more fine-grained potential.
- This improves the overall number of iterations from $O(n^{3.5}\log(\bar{\chi}_A + n))$ to $O(n^{2.5}\log(\bar{\chi}^*_A + n)\log n)$
- Our improved analysis is also applicable to the original Vavasis–Ye algorithm.

Follow up work

- Black box LP solver achieving $\bar{\chi}_A$ dependence using approximate LP solvers: Dadush, Natura, V. '20. Using van den Brand '20: deterministic $O(mn^{\omega+1}\log(n)\log(\bar{\chi}_A+n))$ LP algorithm.

Open questions

- **Practical potential?** preprocessing LPs via column rescaling may be a good idea.
- Improve the running time, both the iteration count and by using faster linear algebra.
- Understand and further explore the combinatorics of $\bar{\chi}_A$ and κ_A.
- Strongly polynomial LP: need to 'leave' the central path or use path induced by different barrier.
- Can we unify strongly polynomial analyses for special LP classes using tools from LLS analysis?
- Can we get combinatorial bounds on the number of iterations of the LLS steps?
  Does 2^n iterations suffice to solve an n-variable LP?