ORF522 – Linear and Nonlinear Optimization

10. Interior-point methods for linear optimization

Bartolomeo Stellato — Fall 2021

Ed Forum

- Is it true that theonline algorithm of recomputing the optimal solution when we found out about a new constraint is the main use case for the dual simplex method?
- How do you determine the magnitude of u (or maybe the range of u's) to prevent it from changing the optimal basis but also not so small that the analyses does not provide meaningless information?

Section 5.1 LO

Recap

Adding new variables

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

Solution $x^\star, y^\star$

Adding new variables

minimize	$c^T x$	$\longrightarrow$	minimize	$c^T x + c_{n+1} x_{n+1}$
subject to	$Ax = b$		subject to	$Ax + A_{n+1} x_{n+1} = b$
	$x \geq 0$			$x, x_{n+1} \geq 0$

Solution $x^\star, y^\star$

Adding new variables

minimize	$c^T x$	$\longrightarrow$	minimize	$c^T x + c_{n+1} x_{n+1}$
subject to	$Ax = b$		subject to	$Ax + A_{n+1} x_{n+1} = b$
	$x \geq 0$			$x, x_{n+1} \geq 0$

Solution $x^\star, y^\star$

Solution $(x^\star, 0), y^\star$ **optimal** for the new problem?

Adding new variables

Optimality conditions

minimize	$c^T x + c_{n+1} x_{n+1}$
subject to	$Ax + A_{n+1} x_{n+1} = b$
	$x, x_{n+1} \geq 0$

$\longrightarrow$ Solution $(x^\star, 0)$ is still **primal feasible**

Adding new variables

Optimality conditions

minimize $\quad c^T x + c_{n+1} x_{n+1}$

subject to $\quad Ax + A_{n+1} x_{n+1} = b \longrightarrow$ Solution $(x^\star, 0)$ is still **primal feasible**

$\quad x, x_{n+1} \geq 0$

Is $y^\star$ still **dual feasible**?

$$A_{n+1}^T y^\star + c_{n+1} \geq 0$$

$$\bar{c} = \begin{bmatrix} A^T y + c \\ A_{n+1}^T y + c_{n+1} \end{bmatrix} \geq 0 \quad ?$$

Adding new variables

Optimality conditions

$$
\begin{array}{ll}
\text{minimize} & c^T x + c_{n+1} x_{n+1} \\
\text{subject to} & Ax + A_{n+1} x_{n+1} = b \\
& x, x_{n+1} \geq 0
\end{array}
$$

⟶ Solution $(x^\star, 0)$ is still **primal feasible**

Is $y^\star$ still **dual feasible**?

$$A_{n+1}^T y^\star + c_{n+1} \geq 0$$

Yes	**Otherwise**
$(x^\star, 0)$ still **optimal** for new problem	Primal simplex

Adding new constraints

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

Solution $x^\star, y^\star$

Adding new constraints

minimize	$c^T x$		minimize	$c^T x$
subject to	$Ax = b$	$\longrightarrow$	subject to	$Ax = b$
	$x \geq 0$			$a_{m+1}^T x = b_{m+1}$
				$x \geq 0$

Solution $x^\star, y^\star$

Adding new constraints

minimize	$c^T x$		minimize	$c^T x$
subject to	$Ax = b$	$\longrightarrow$	subject to	$Ax = b$
	$x \geq 0$			$a_{m+1}^T x = b_{m+1}$
				$x \geq 0$

Solution $x^\star, y^\star$

Dual

$$\begin{array}{ll} \text{maximize} & -b^T y \\ \text{subject to} & A^T y + a_{m+1} y_{m+1} + c \geq 0 \end{array}$$

Adding new constraints

minimize $c^T x$
subject to $Ax = b$
$x \geq 0$

Solution $x^\star, y^\star$

$\longrightarrow$

minimize $c^T x$
subject to $Ax = b$
$a_{m+1}^T x = b_{m+1}$
$x \geq 0$

Dual

maximize $-b^T y$
subject to $A^T y + a_{m+1} y_{m+1} + c \geq 0$

Solution $x^\star, (y^\star, 0)$ **optimal** for the new problem?

Adding new constraints
Optimality conditions

maximize $\quad -b^T y$

subject to $\quad A^T y + a_{m+1} y_{m+1} + c \geq 0$ ⟶ Solution $(y^\star, 0)$ is still **dual feasible**

Adding new constraints

Optimality conditions

maximize $-b^T y$

subject to $A^T y + a_{m+1} y_{m+1} + c \geq 0$ ⟶ Solution $(y^\star, 0)$ is still **dual feasible**

Is $x^\star$ still **primal feasible**?

$$Ax = b$$
$$a_{m+1}^T x = b_{m+1}$$
$$x \geq 0$$

Adding new constraints

Optimality conditions

maximize $-b^T y$

subject to $A^T y + a_{m+1} y_{m+1} + c \geq 0$ ⟶ Solution $(y^\star, 0)$ is still **dual feasible**

Is $x^\star$ still **primal feasible**?

$$
\begin{aligned}
&Ax = b\\
&a_{m+1}^T x = b_{m+1}\\
&x \geq 0
\end{aligned}
$$

Yes

$x^\star$ still **optimal** for new problem

Otherwise

Dual simplex

Today's lecture

[Chapter 14, NO][Chapters 17/18, LP]

- History
- Newton's method
- Central path
- Primal-dual path-following algorithm

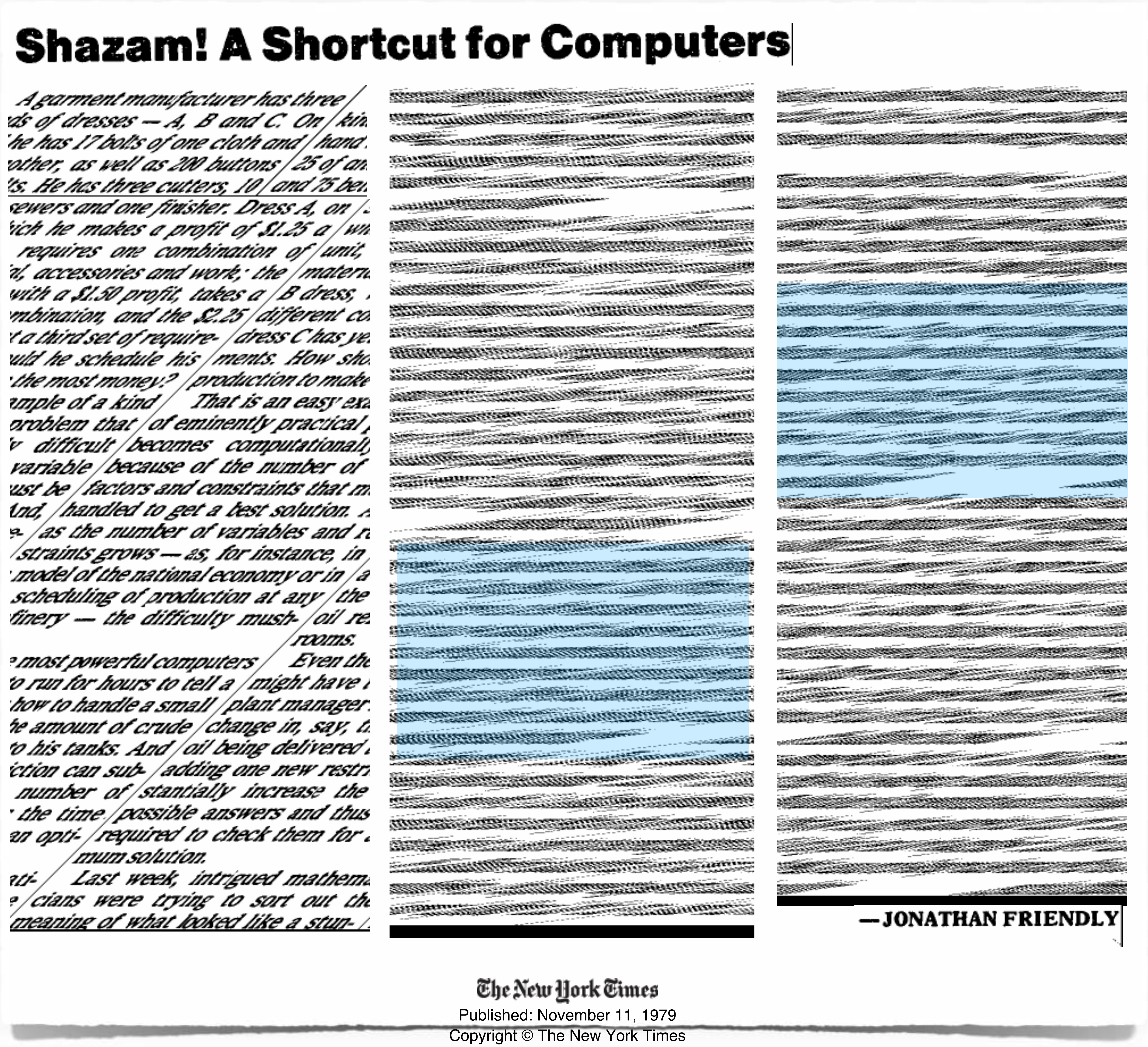
History

Ellipsoid method

Khachian (1979)

Answer to major question

Is worst-case LP complexity polynomial? Yes!



Shazam! A Shortcut for Computers

A garment manufacturer has three
-ds of dresses — A, B and C. On
-he has 17 bolts of one cloth and
-other, as well as 200 buttons
-ts. He has three cutters, 10
sewers and one finisher. Dress A, on
-ich he makes a profit of $1.25 a
requires one combination of
-d, accessories and work; the
with a $1.50 profit, takes a
-mbination, and the $2.25
-t a third set of require-
-uld he schedule his
the most money?
-ample of a kind
problem that
-y difficult
variable
-ust be
And,
straints grows — as, for instance, in
model of the national economy or in
scheduling of production at any
-finery — the difficulty mush-
most powerful computers
to run for hours to tell a
how to handle a small
the amount of crude
to his tanks. And
-ction can sub-
number of
the time
-an opti-
-ati-
meaning of what looked like a stun-

kin- / hand / 25 of an / and 75 be- / wh- / unit, / materi- / B dress, / different co- / dress C has ye- / ments. How sho- / production to make / That is an easy exa- / of eminently practical / becomes computationally / because of the number of / factors and constraints that m- / handled to get a best solution. A- / as the number of variables and r- / a / the / oil re- / rooms. / Even the / might have / plant manager / change in, say, t- / oil being delivered / adding one new restr- / stantially increase the / possible answers and thus / required to check them for a / mum solution. / Last week, intrigued mathema- / cians were trying to sort out the

—JONATHAN FRIENDLY

The New York Times

Published: November 11, 1979

Ellipsoid method

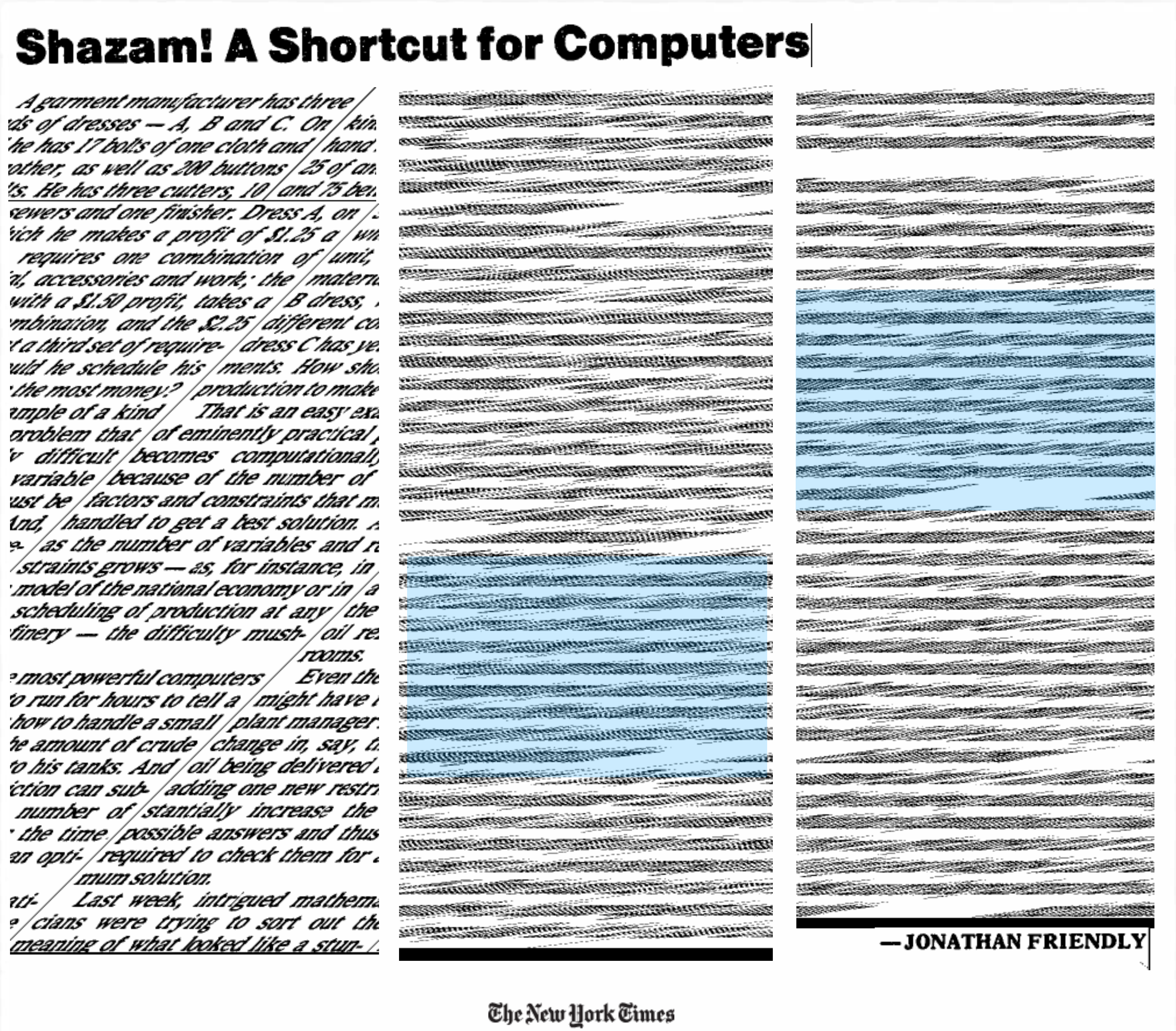
Khachian (1979)

Answer to major question

Is worst-case LP complexity polynomial? Yes!

Drawbacks

Very inefficient. Much slower than simplex!



Published: November 11, 1979

Ellipsoid method

Khachian (1979)

Answer to major question

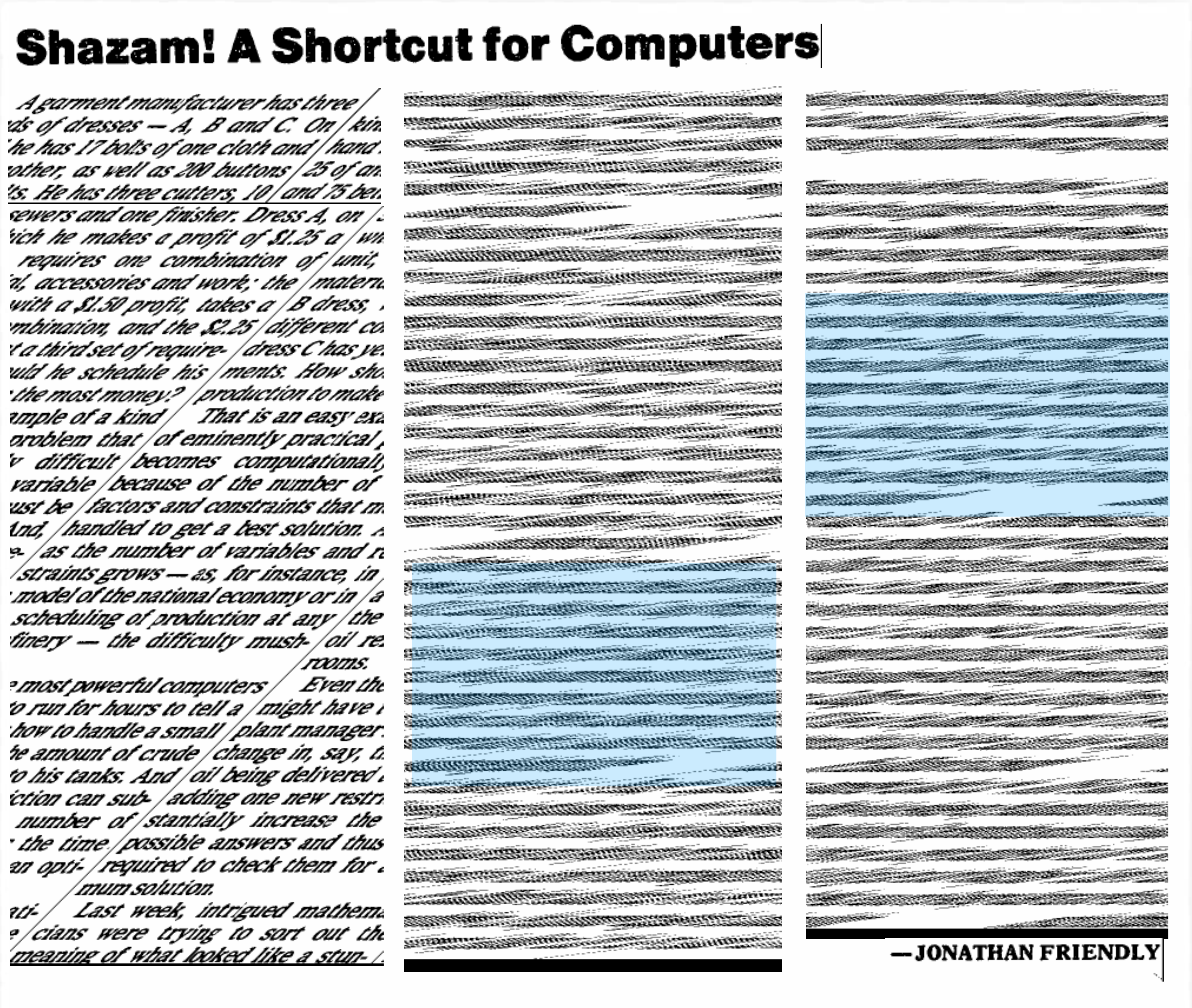
Is worst-case LP complexity polynomial? Yes!

Drawbacks

Very inefficient. Much slower than simplex!

Benefits

Motivated new research directions



The New York Times
Published: November 11, 1979

Interior-point methods

1950s-1960s: nonlinear convex optimization

- Sequential unconstrained optimization (Fiacco & McCormick), Logarithmic barrier method (Frish), affine scaling method (Dikin), etc.
- No worst-case complexity theory but often good practical performance

Interior-point methods

1950s-1960s: nonlinear convex optimization

- Sequential unconstrained optimization (Fiacco & McCormick), Logarithmic barrier method (Frish), affine scaling method (Dikin), etc.
- No worst-case complexity theory but often good practical performance

1980s-1990s: interior point methods

- Karmarkar's algorithm (1984)
- Competitive with simplex, often faster for larger problems

The New York Times

NEW YORK, MONDAY, NOVEMBER 19, 1984

AMERICANS IN POLL VIEW GOVERNMENT MORE CONFIDENTLY

TREASURY MAY ASK INCREASES IN TAXES OF SOME CONCERNS

Albany Leaders Predicting a Cut In Income Taxes

Cocaine Traffickers Kill 17 in Peru Raid On Antidrug Team

Vote Comes to a 'Homeland,' But African Problems Linger

Homeless Spend Nights in City Welfare Office

Breakthrough in Problem Solving

By JAMES GLEICK

A 28-year-old mathematician at A.T.&T. Bell Laboratories has made a startling theoretical breakthrough in the solving of systems of equations that often grow too vast and complex for the most powerful computers.

The discovery, which is to be formally published next month, is already circulating rapidly through the mathematical world. It has also set off a deluge of inquiries from brokerage houses, oil companies and airlines, industries with millions of dollars at stake in problems known as linear programming.

These problems are fiendishly complicated systems, often with thousands of variables. They arise in a variety of commercial and government applications, ranging from allocating time on a communications satellite to routing millions of telephone calls over long distances. Linear programming is particularly useful whenever a limited, expensive resource must be spread most efficiently among competing users. And investment companies use the approach in creating portfolios with the best mix of stocks and bonds.

Faster Solutions Seen

The Bell Labs mathematician, Dr. Narendra Karmarkar, has devised a radically new procedure that may speed the routine handling of such problems by businesses and Government agencies and also make it possible to tackle problems that are now far out of reach.

"This is a path-breaking result," said Dr. Ronald L. Graham, director of mathematical sciences for Bell Labs in Murray Hill, N.J. "Science has its moments of great progress, and this may well be one of them."

Because problems in linear programming can have billions or more possible answers, even high-speed computers cannot check every one. So computers must use a special procedure, an algorithm, to examine as few answers as possible before finding the best one — typically the one that minimizes cost or maximizes efficiency.

A procedure devised in 1947, the simplex method, is now used for such prob-

Continued on Page A19, Column 1

Newton’s method

Newton's method for nonlinear equations

$h: \mathbb{R}^n \to \mathbb{R}^m$

Goal: solve

$$h(x) = 0$$

Derivative

$$Dh = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_n} \end{bmatrix}$$

Newton's method for nonlinear equations

Derivative

$$Dh = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_n} \end{bmatrix}$$

Goal: solve

$$h(x) = 0$$

First-order approximation

$$h(x) \approx h(\bar{x}) + Dh(\bar{x})(x - \bar{x})$$

→

Iteratively set to zero

$$h(x^k) + Dh(x^k)(x^{k+1} - x^k) = 0$$

Newton's method for nonlinear equations

Derivative

$$Dh = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \dots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \dots & \frac{\partial h_m}{\partial x_n} \end{bmatrix}$$

Goal: solve

$$h(x) = 0$$

First-order approximation

$$h(x) \approx h(\bar{x}) + Dh(\bar{x})(x - \bar{x})$$

→

Iteratively set to zero

$$h(x^k) + Dh(x^k)(x^{k+1} - x^k) = 0$$

Iterations

- Solve $Dh(x^k)\Delta x = -h(x^k)$
- $x^{k+1} \leftarrow x^k + \Delta x$

Newton method

Convergence

x st. h(x) = 0

Iterations

- Solve $Dh(x^k)\Delta x = -h(x^k)$
- $x^{k+1} \leftarrow x^k + \Delta x$

Remarks

- Iterations can be **expensive** (linear system solution)
- **Fast (quadratic) convergence** close to the solution $x^\star$

Optimality conditions

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \end{array}$$

Optimality conditions

				Primal		**Dual**
minimize	$c^T x$	$\longrightarrow$	minimize	$c^T x$	maximize	$-b^T y$
subject to	$Ax \leq b$		subject to	$Ax + s = b$	subject to	$A^T y + c = 0$
				$s \geq 0$		$y \geq 0$

Optimality conditions

		Primal	**Dual**
minimize $c^T x$	$\longrightarrow$	minimize $c^T x$	maximize $-b^T y$
subject to $Ax \le b$		subject to $Ax + s = b$	subject to $A^T y + c = 0$
		$s \ge 0$	$y \ge 0$

Optimality conditions

$$Ax + s - b = 0$$

$$A^T y + c = 0$$

$$s_i y_i = 0$$

$$s, y \ge 0$$

$s_i = (b - Ax)_i$

Main idea

$s_i y_i = 0 \ \forall i \iff \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_m \end{bmatrix} \begin{bmatrix} y_1 & & \\ & \ddots & \\ & & y_m \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = 0$

Optimality conditions

$$h(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY\mathbf{1} \end{bmatrix} = 0$$

$$s, y \geq 0$$

$$S = \mathbf{diag}(s)$$
$$Y = \mathbf{diag}(y)$$

- Apply variants of Newton's method to solve $h(x, s, y) = 0$
- Enforce $s, y > 0$ (strictly) at every iteration
- **Motivation** avoid getting stuck in "corners"

Newton's method for optimality conditions

Root-finding equation

$$h(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY\mathbf{1} \end{bmatrix} = 0$$

Linear system

$$\underbrace{\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix}}_{Dh} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \underbrace{\begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} \end{bmatrix}}_{-h}$$

Residuals

$$r_p = Ax + s - b$$

$$r_d = A^T y + c$$

Newton's method for optimality conditions

Root-finding equation

$$h(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY\mathbf{1} \end{bmatrix} = 0$$

Linear system

$$\overbrace{\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix}}^{Dh} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \overbrace{\begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} \end{bmatrix}}^{-h}$$

Residuals

$$r_p = Ax + s - b$$

$$r_d = A^T y + c$$

Line search to enforce $x, s > 0$

$$(x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)$$

Newton’s method for optimality conditions

Root-finding equation

$$h(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY\mathbf{1} \end{bmatrix} = 0$$

Linear system

$$\overbrace{\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix}}^{Dh} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \overbrace{\begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} \end{bmatrix}}^{-h}$$

Residuals

$$r_p = Ax + s - b$$

$$r_d = A^T y + c$$

Line search to enforce $x, s > 0$

$$(x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)$$

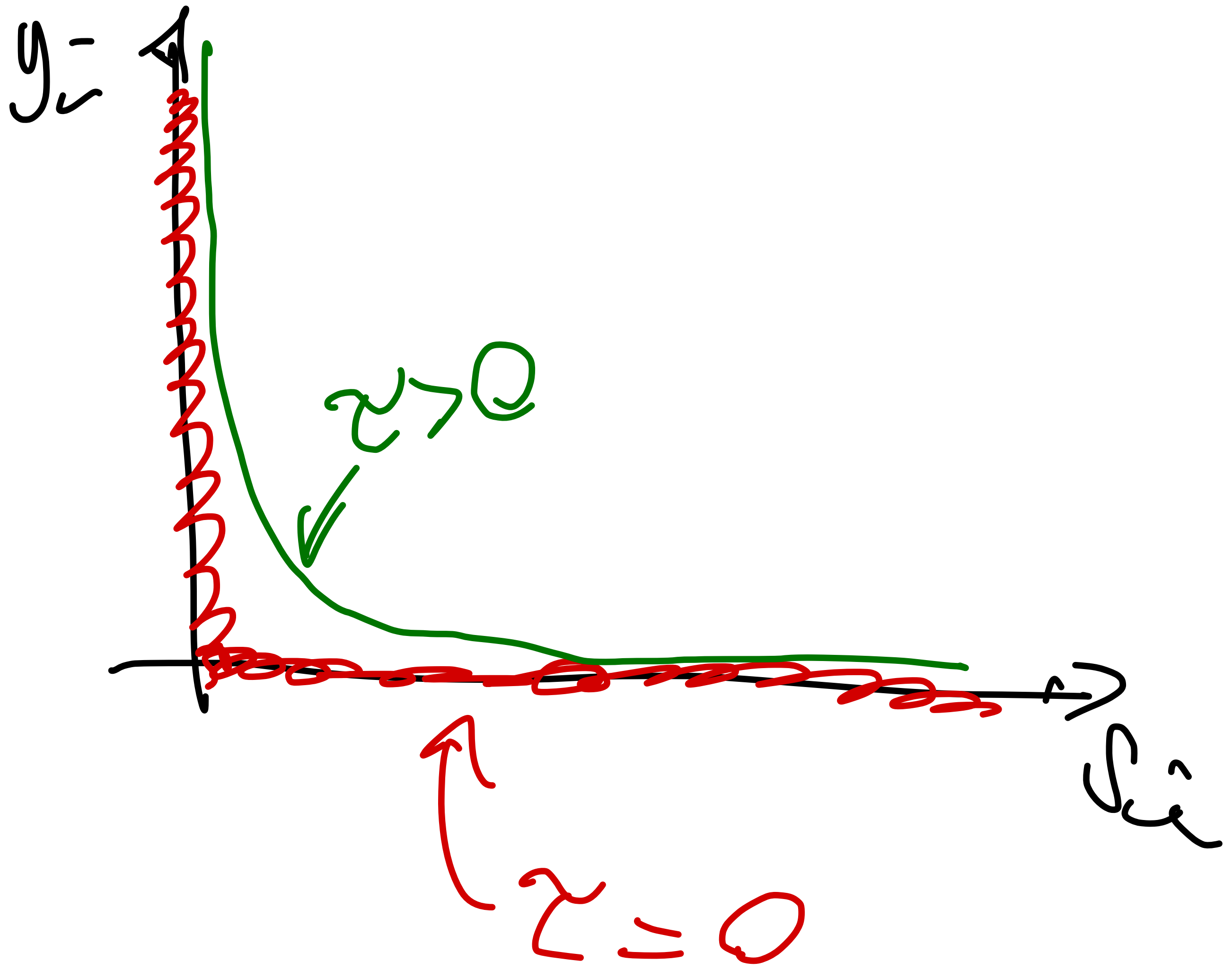
Issue

Pure **Newton’s step** does not allow significant progress towards

$h(x, s, y) = 0$ **and** $x, y \geq 0$.

Central path

Smoothed optimality conditions



Optimality conditions

$$Ax + s - b = 0$$

$$A^T y + c = 0$$

$$s_i y_i = \tau$$

$$s, y \geq 0$$

← Same τ for every pair

Same optimality conditions for a "smoothed" version of our problem

Newton’s method for smoothed optimality conditions

Smoothed optimality conditions

$$h_\tau(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY\mathbf{1} - \tau\mathbf{1} \end{bmatrix} = 0$$

$$s, y \geq 0$$

Newton's method for smoothed optimality conditions

Smoothed optimality conditions

$$h_\tau(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY\mathbf{1} - \tau\mathbf{1} \end{bmatrix} = 0$$

$$s, y \geq 0$$

Linear system

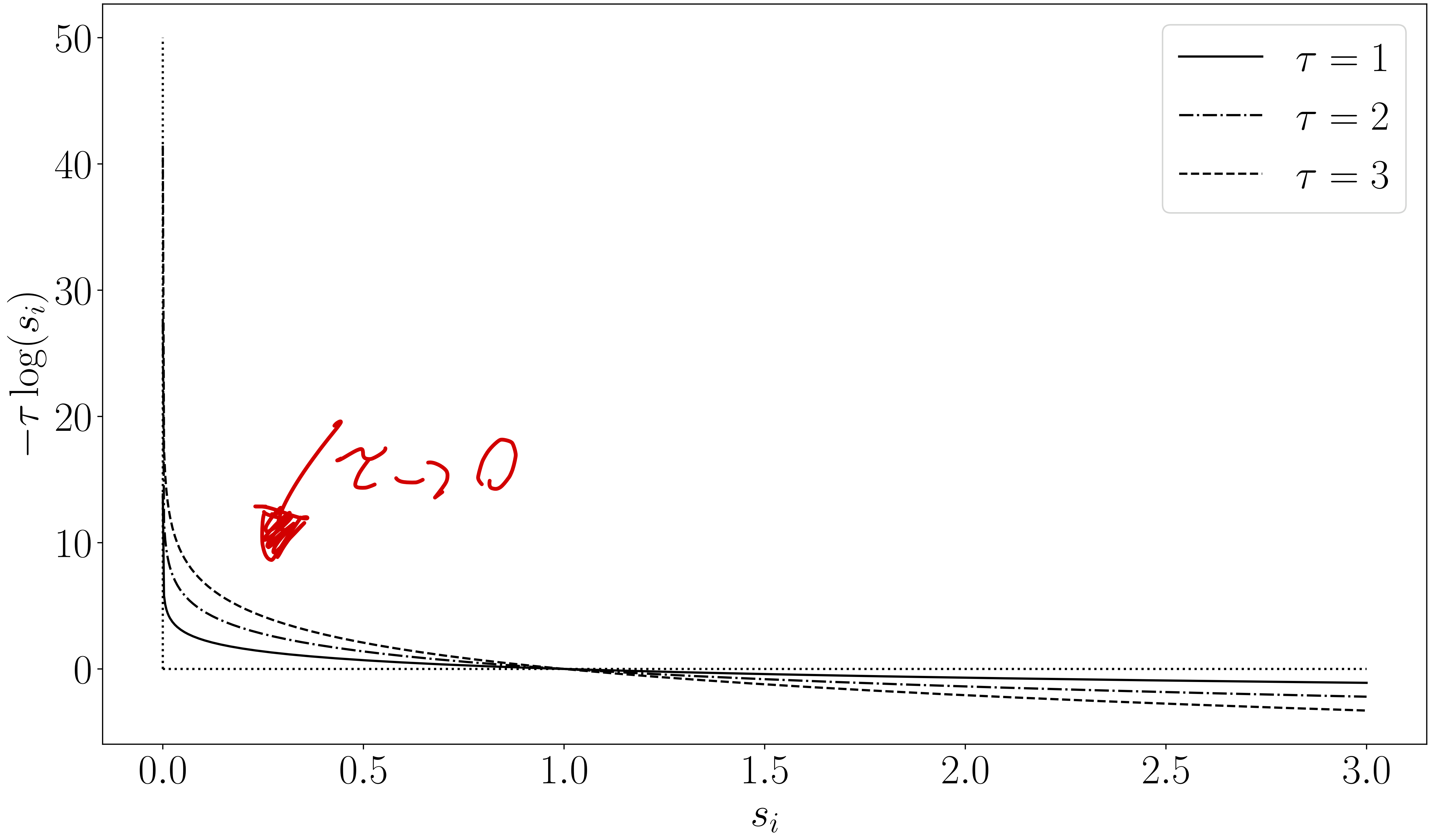
$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY + \tau\mathbf{1} \end{bmatrix}$$

Line search to enforce $x, s > 0$

$$(x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)$$

Logarithmic barrier

$$\phi(s) = -\tau \sum_{i=1}^{m} \log(s_i) \quad \text{on domain} \quad s_i > 0$$



As $\tau \to 0$ it approximates

$$\mathcal{I}_{s_i \geq 0} = \begin{cases} 0 & \text{if } s_i \geq 0 \\ \infty & \text{otherwise} \end{cases}$$

Smoothed problem

$$
\begin{array}{ll}
\text{minimize} & c^T x \\
\text{subject to} & Ax + s = b \\
& s \geq 0
\end{array}
$$

Smoothed problem

minimize $\quad c^T x$

subject to $\quad Ax + s = b$

$\quad s \geq 0$

$\longrightarrow$

minimize $\quad c^T x + \phi(x) = c^T x - \tau \sum_{i=1}^{m} \log(s_i)$

subject to $\quad Ax + s = b$

Smoothed problem

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax + s = b \\ & s \geq 0 \end{aligned} \longrightarrow \begin{aligned} \text{minimize} \quad & c^T x + \phi(x) = c^T x - \tau \textstyle\sum_{i=1}^m \log(s_i) \\ \text{subject to} \quad & Ax + s = b \end{aligned}$$

Dual cost

$$g(y) = \underset{x,s}{\text{minimize}}\ \mathcal{L}(x, s, y) = c^T x + \phi(s) + y^T (Ax + s - b)$$

Smoothed problem

minimize $\quad c^T x$

subject to $\quad Ax + s = b$

$\qquad\qquad s \geq 0$

$\longrightarrow$

minimize $\quad c^T x + \phi(x) = c^T x - \tau \sum_{i=1}^m \log(s_i)$

subject to $\quad Ax + s = b$

Dual cost

$$g(y) = \underset{x,s}{\text{minimize}}\ \mathcal{L}(x, s, y) = c^T x + \phi(s) + y^T (Ax + s - b)$$

$$\frac{\partial \mathcal{L}}{\partial x} = A^T y + c = 0$$

$$\frac{\partial \mathcal{L}}{\partial s_i} = -\tau \frac{1}{s_i} + y_i = 0 \qquad \Longrightarrow s_i y_i = \tau$$

Central path

minimize $\quad c^T x - \tau \sum_{i=1}^m \log(s_i)$
subject to $\quad Ax + s = b$

Set of points $(x^\star(\tau), s^\star(\tau), y^\star(\tau))$
with $\tau > 0$ such that

$$\begin{aligned} Ax + s - b &= 0 \\ A^T y + c &= 0 \\ s_i y_i &= \tau \\ s, y &\geq 0 \end{aligned}$$

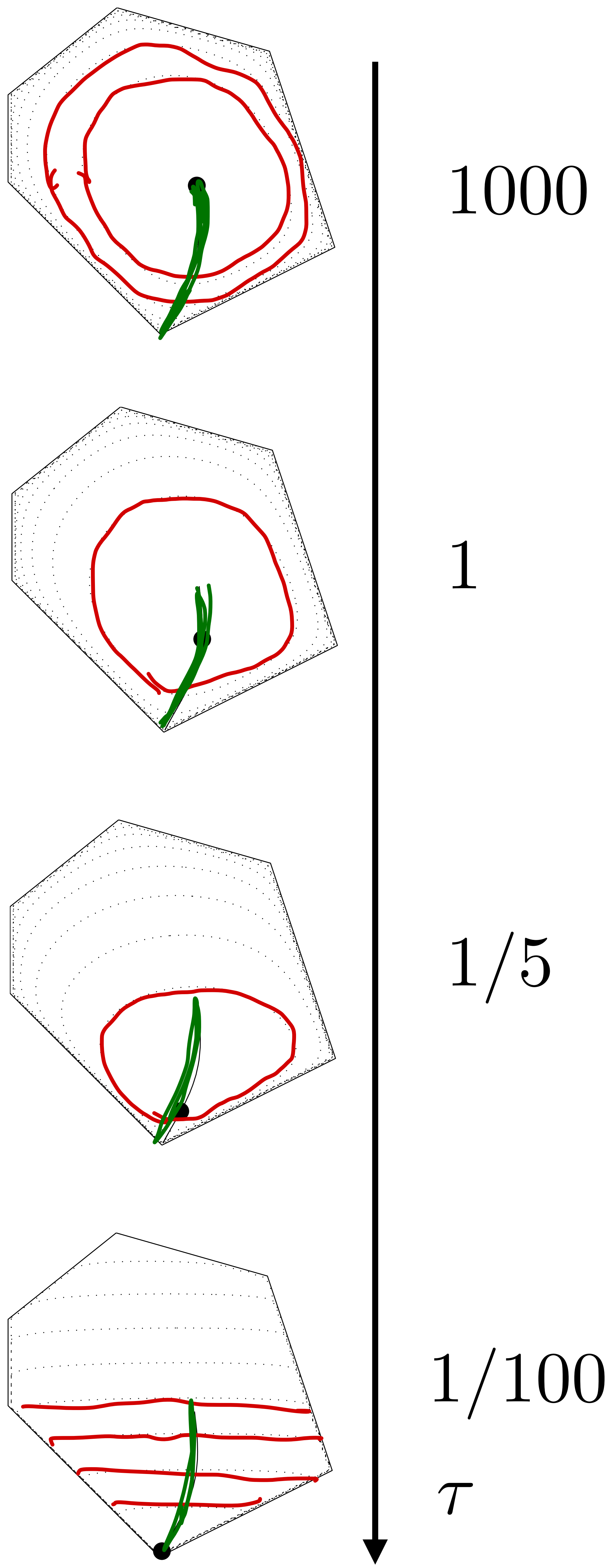
Central path

minimize $\quad c^T x - \tau \sum_{i=1}^m \log(s_i)$
subject to $\quad Ax + s = b$

Set of points $(x^\star(\tau), s^\star(\tau), y^\star(\tau))$ with $\tau > 0$ such that

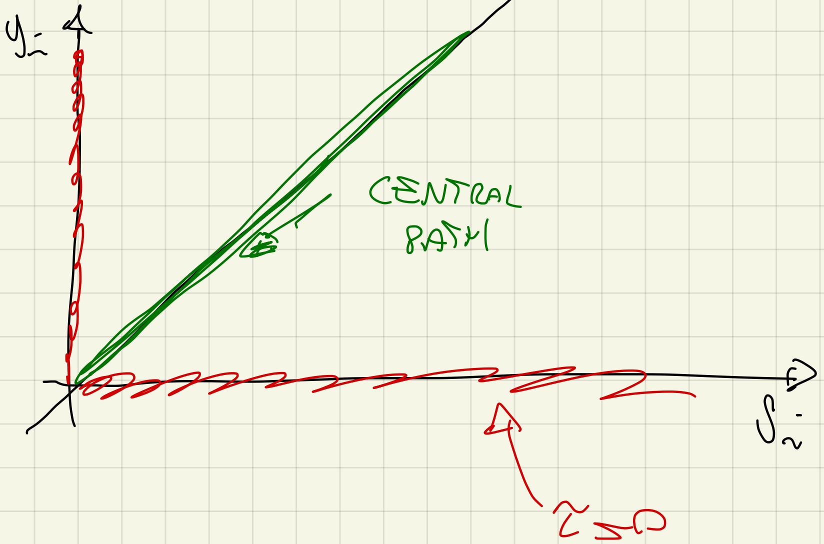
$$Ax + s - b = 0$$
$$A^T y + c = 0$$
$$s_i y_i = \tau$$
$$s, y \geq 0$$

Analytic Center
$\tau \to \infty$



Main idea

Follow central path as $\tau \to 0$



y_i
s_i
CENTRAL
PATH
$\tilde{z} = 0$

Primal-dual path-following method

Duality measure

Definition

$$\mu = \frac{s^T y}{m}$$

Average value of the pairs $s_i y_i$

It describes the "desirability" of each point in the search space

Algorithm step

Linear system

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} + \sigma\mu\mathbf{1} \end{bmatrix}$$

Duality measure

$$\mu = \frac{s^T y}{m}$$

Centering parameter

$$\sigma \in [0, 1]$$

Algorithm step

Linear system

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} + \sigma\mu\mathbf{1} \end{bmatrix}$$

Duality measure

$$\mu = \frac{s^T y}{m}$$

Centering parameter

$\sigma \in [0, 1]$

$\sigma = 0 \quad \Rightarrow$ Newton step

$\sigma = 1 \quad \Rightarrow$ Centering step towards $(x^\star(\mu), s^\star(\mu), y^\star(\mu))$

Algorithm step

Linear system

$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} + \sigma\mu\mathbf{1} \end{bmatrix}$$

Duality measure

$$\mu = \frac{s^T y}{m}$$

Centering parameter

$\sigma \in [0, 1]$

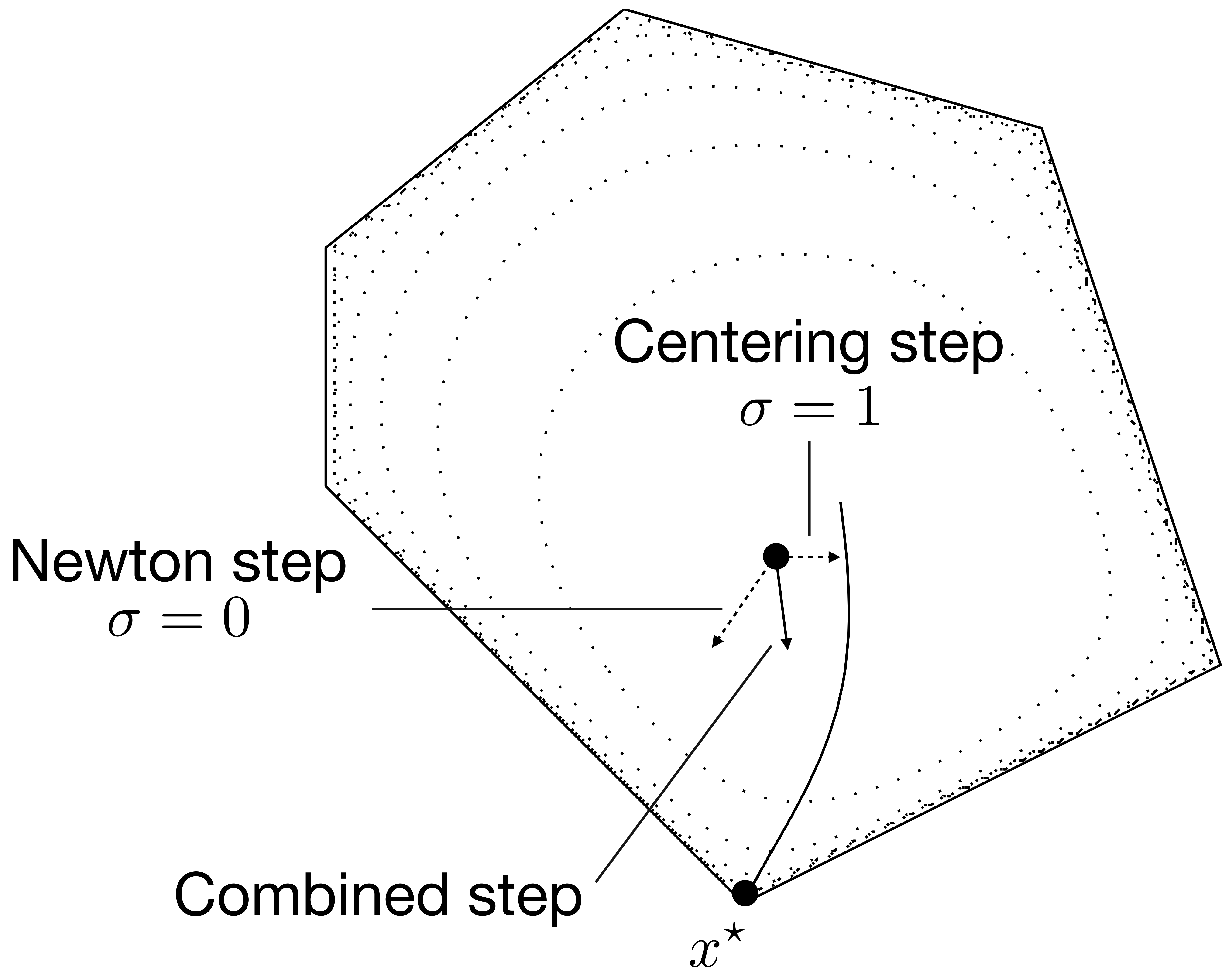
$\sigma = 0 \quad \Rightarrow \quad$ Newton step

$\sigma = 1 \quad \Rightarrow \quad$ Centering step towards $(x^\star(\mu), s^\star(\mu), y^\star(\mu))$

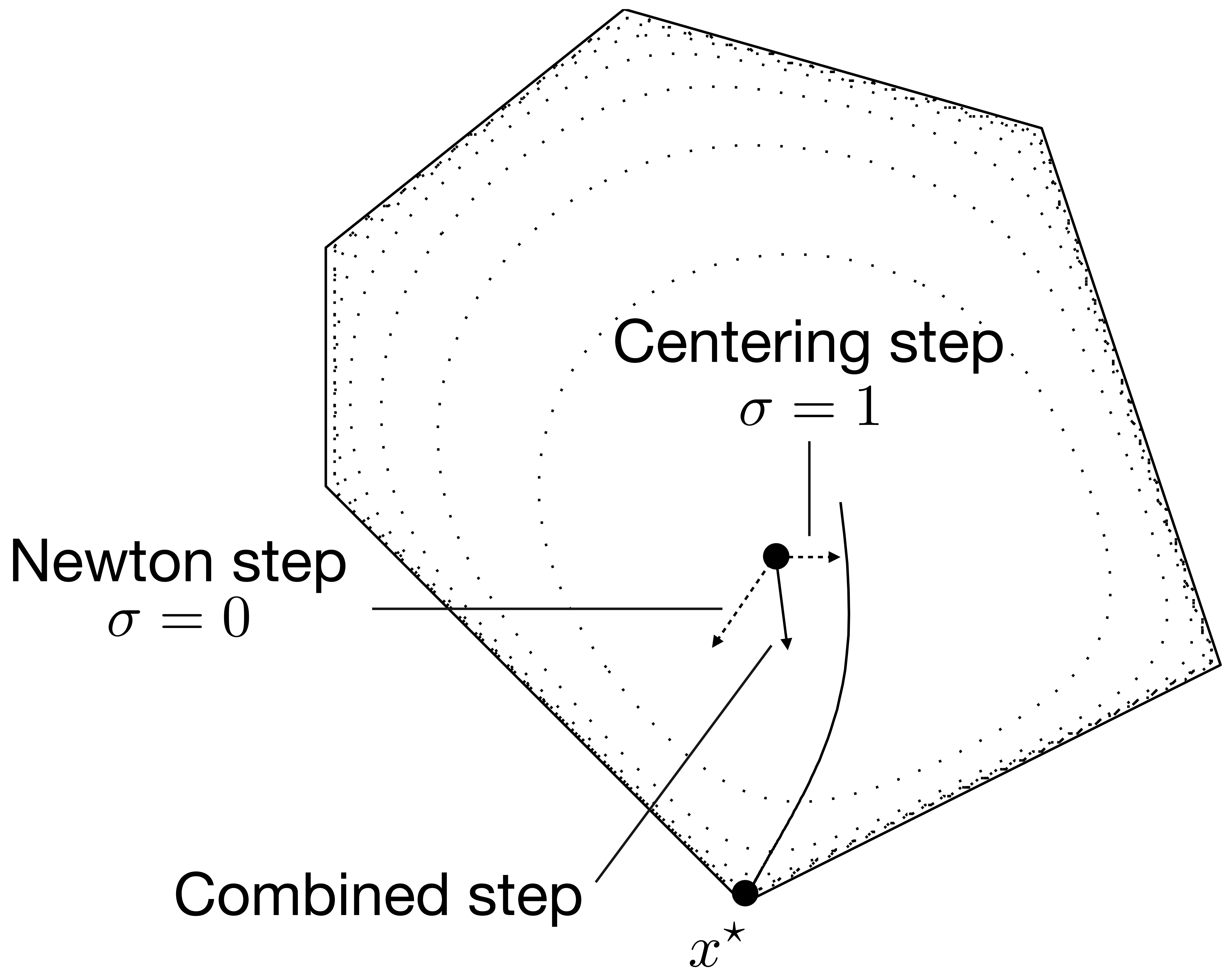
Line search to enforce $x, s > 0$

$$(x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)$$

Path-following algorithm idea

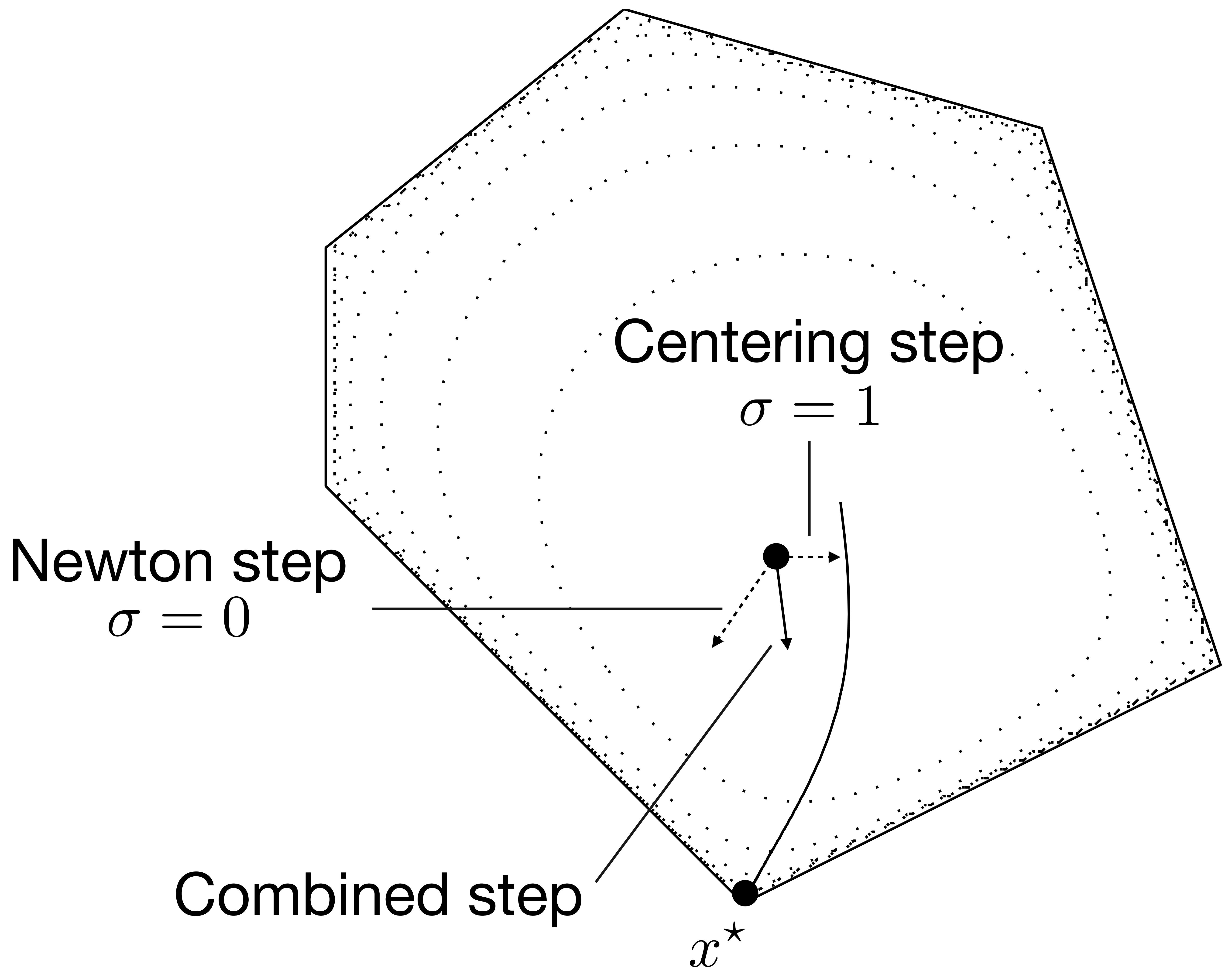


Path-following algorithm idea



Centering step
It brings towards the **central path** and is usually biased towards $s, y > 0$.
No progress on duality measure μ

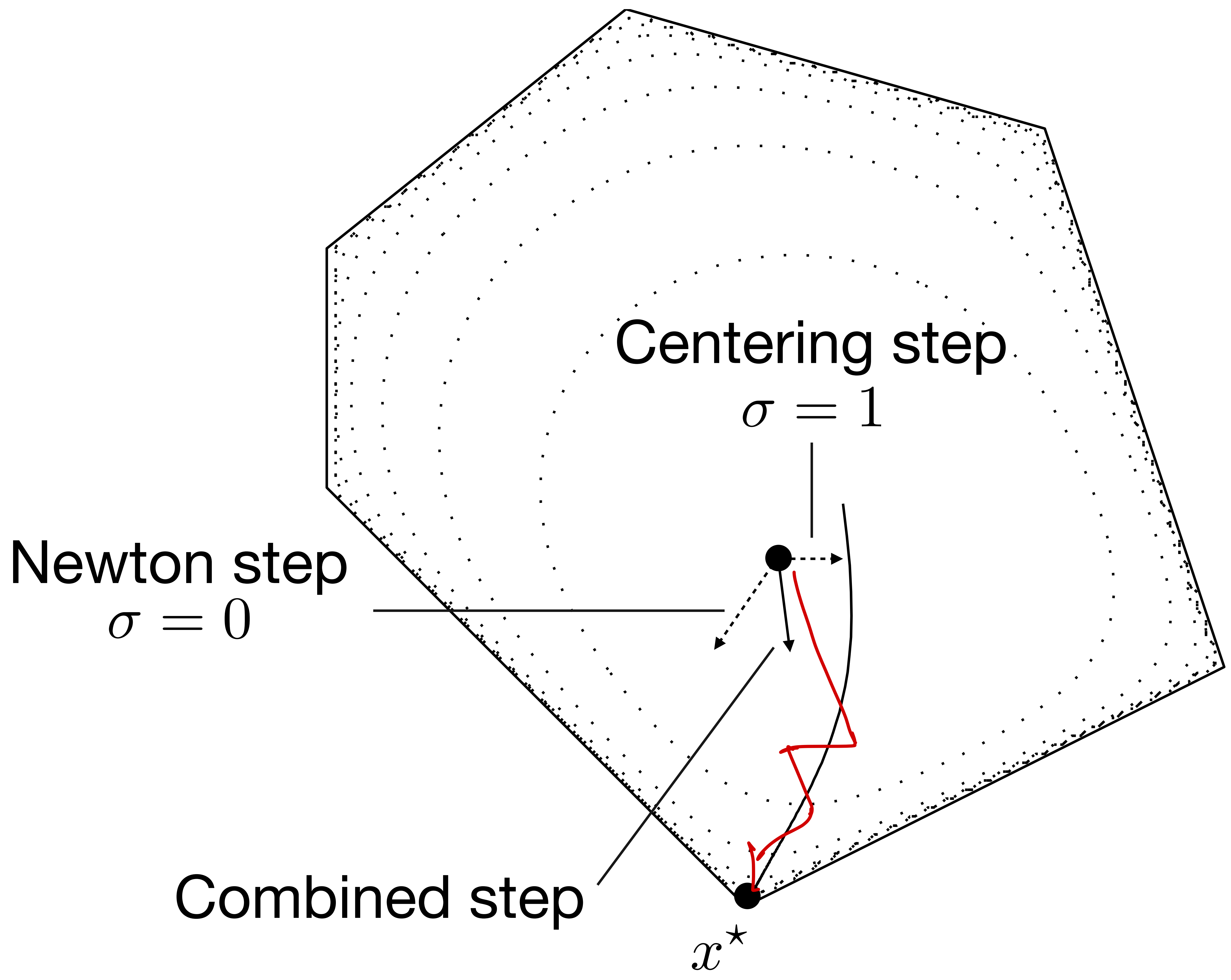
Path-following algorithm idea



Centering step
It brings towards the **central path** and is usually biased towards $s, y > 0$. **No progress** on duality measure μ

Newton step
It brings towards the **zero duality measure** μ. Quickly violates $s, y > 0$.

Path-following algorithm idea



Centering step
It brings towards the **central path** and is usually biased towards $s, y > 0$. **No progress** on duality measure μ

Newton step
It brings towards the **zero duality measure** μ. Quickly violates $s, y > 0$.

Combined step
Best of both worlds with longer steps

Primal-dual path-following algorithm

Initialization

1. Given (x_0, s_0, y_0) such that $s_0, y_0 > 0$

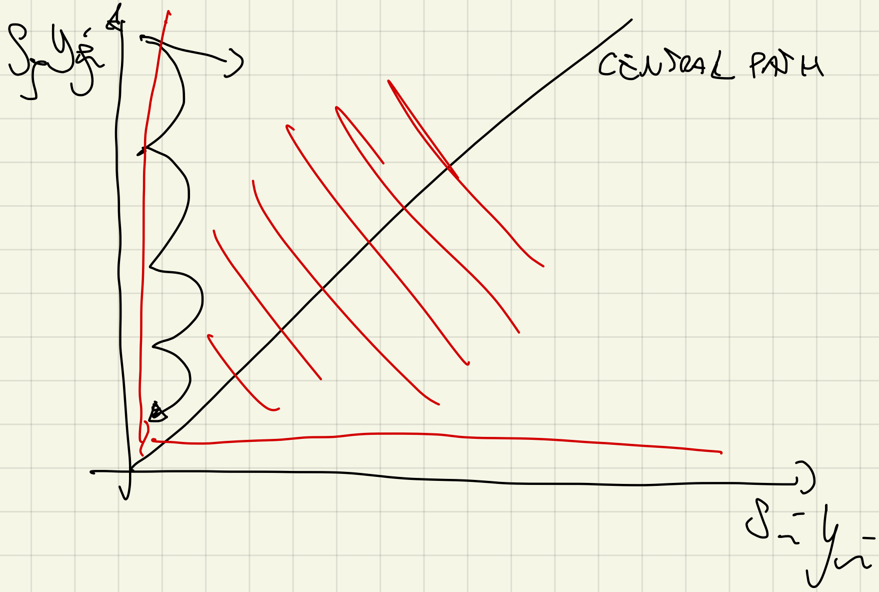
Iterations

1. Choose $\sigma \in [0, 1]$

2. Solve $\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY\mathbf{1} + \sigma\mu\mathbf{1} \end{bmatrix}$ where $\mu = s^T y / m$

3. Find maximum α such that $y + \alpha\Delta y > 0$ and $s + \alpha\Delta s > 0$

4. Update $(x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)$


CENTRAL PATH

Working towards optimality conditions

Optimality conditions satisfied **only at convergence**

Primal residual

$$r_p = Ax + s - b \to 0$$

Dual residual

$$r_d = A^T y + c \to 0$$

Complementary slackness

$$s^T y \to 0$$

Working towards optimality conditions

Optimality conditions satisfied **only at convergence**

Primal residual

$$r_p = Ax + s - b \to 0$$

Dual residual

$$r_d = A^T y + c \to 0$$

Complementary slackness

$$s^T y \to 0$$

Stopping criteria

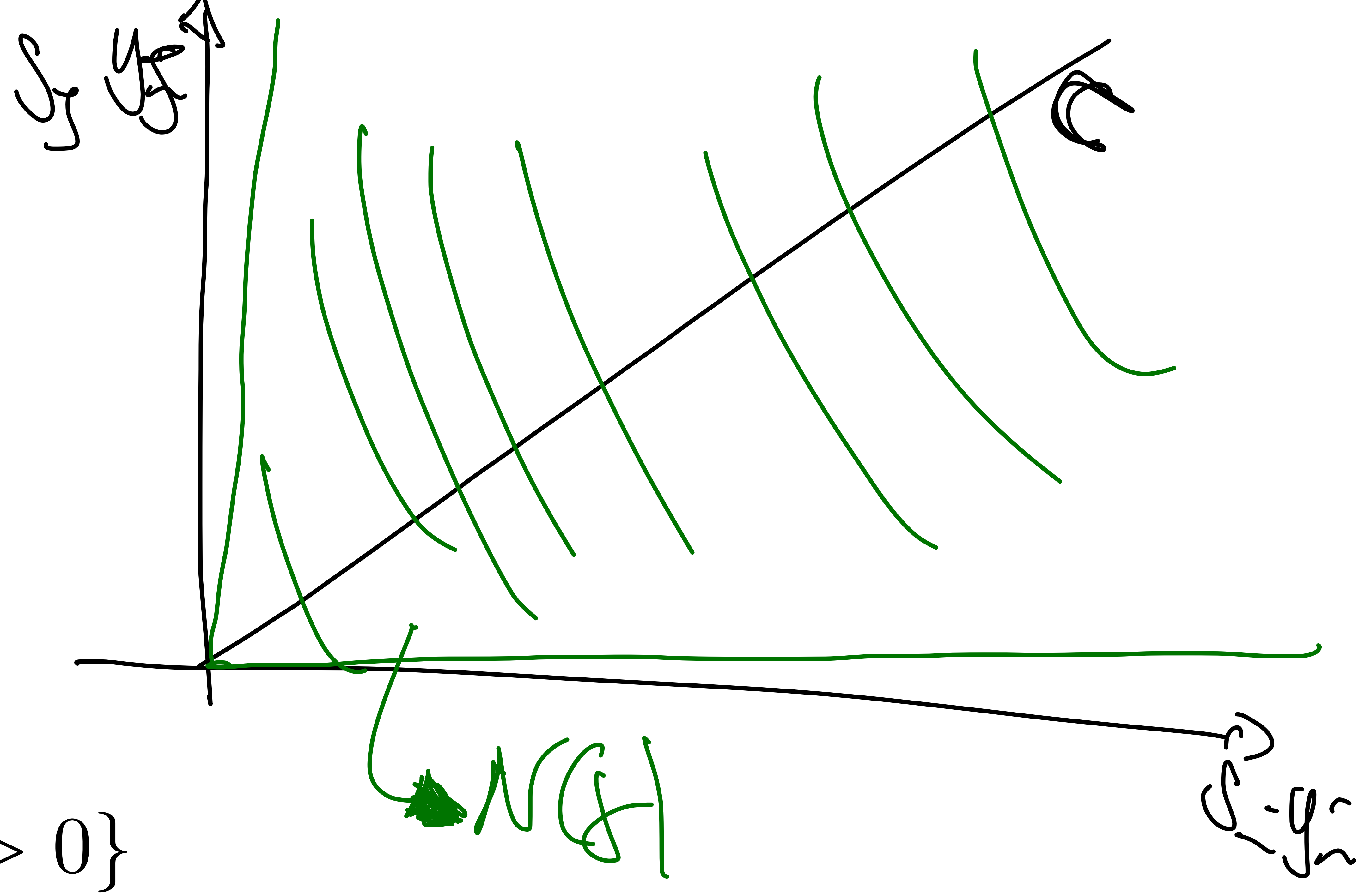
$$\|r_p\| \leq \epsilon_{\text{pri}}$$

$$\|r_d\| \leq \epsilon_{\text{dua}}$$

$$s^T y \leq \epsilon_{\text{gap}}$$

Convergence

Definitions



Primal-dual strictly feasible set

$$\mathcal{F}^\circ = \{(x, s, y) \mid Ax + s = b,\ A^T y + c = 0,\ s, y > 0\}$$

Central path neighborhood

$$\mathcal{N}(\gamma) = \{(x, s, y) \in \mathcal{F}^\circ \mid s_i y_i \geq \gamma\mu\}$$ with $\gamma \in (0, 1]$ (almost all the feasible region)

Theorem
[Page 402-406, NO]

Smallest decrement

$\mu_{k+1} \leq (1 - \delta/n)\, \mu_k$ with constant $\delta > 0$

Theorem
[Page 402-406, NO]

Smallest decrement

$\mu_{k+1} \leq (1 - \delta/n)\, \mu_k$ with constant $\delta > 0$

Iteration complexity

Given $(x_0, s_0, y_0) \in \mathcal{N}(\gamma)$, there exists $K = O(n \log(1/\epsilon))$ such that

$\mu_k \leq \epsilon \mu_0$ for all $k \geq K$

Theorem

[Page 402-406, NO]

Smallest decrement

$$\mu_{k+1} \leq (1 - \delta/n)\,\mu_k \qquad \text{with constant } \delta > 0$$

Iteration complexity

Given $(x_0, s_0, y_0) \in \mathcal{N}(\gamma)$, there exists $K = O(n \log(1/\epsilon))$ such that

$$\mu_k \leq \epsilon \mu_0 \qquad \text{for all } k \geq K$$

Remark Modified versions achieve $O(\sqrt{n} \log(1/\epsilon))$

Iteration complexity proof

[Page 402-406, NO]

$$\mu_{k+1} \leq (1 - \delta/n)\, \mu_k$$

Iteration complexity proof

[Page 402-406, NO]

$$\mu_{k+1} \leq (1-\delta/n)\,\mu_k$$

(take logarithm)

$$\log \mu_{k+1} \leq \log{(1-\delta/n)} + \log \mu_k$$

Iteration complexity proof

[Page 402-406, NO]

$\log\left(\frac{\mu_k}{\mu_0}\right) \leq k \log\left(1 - \delta/n\right)$

$$\mu_{k+1} \leq (1 - \delta/n)\, \mu_k$$

(take logarithm)

$$\log \mu_{k+1} \leq \log (1 - \delta/n) + \log \mu_k$$

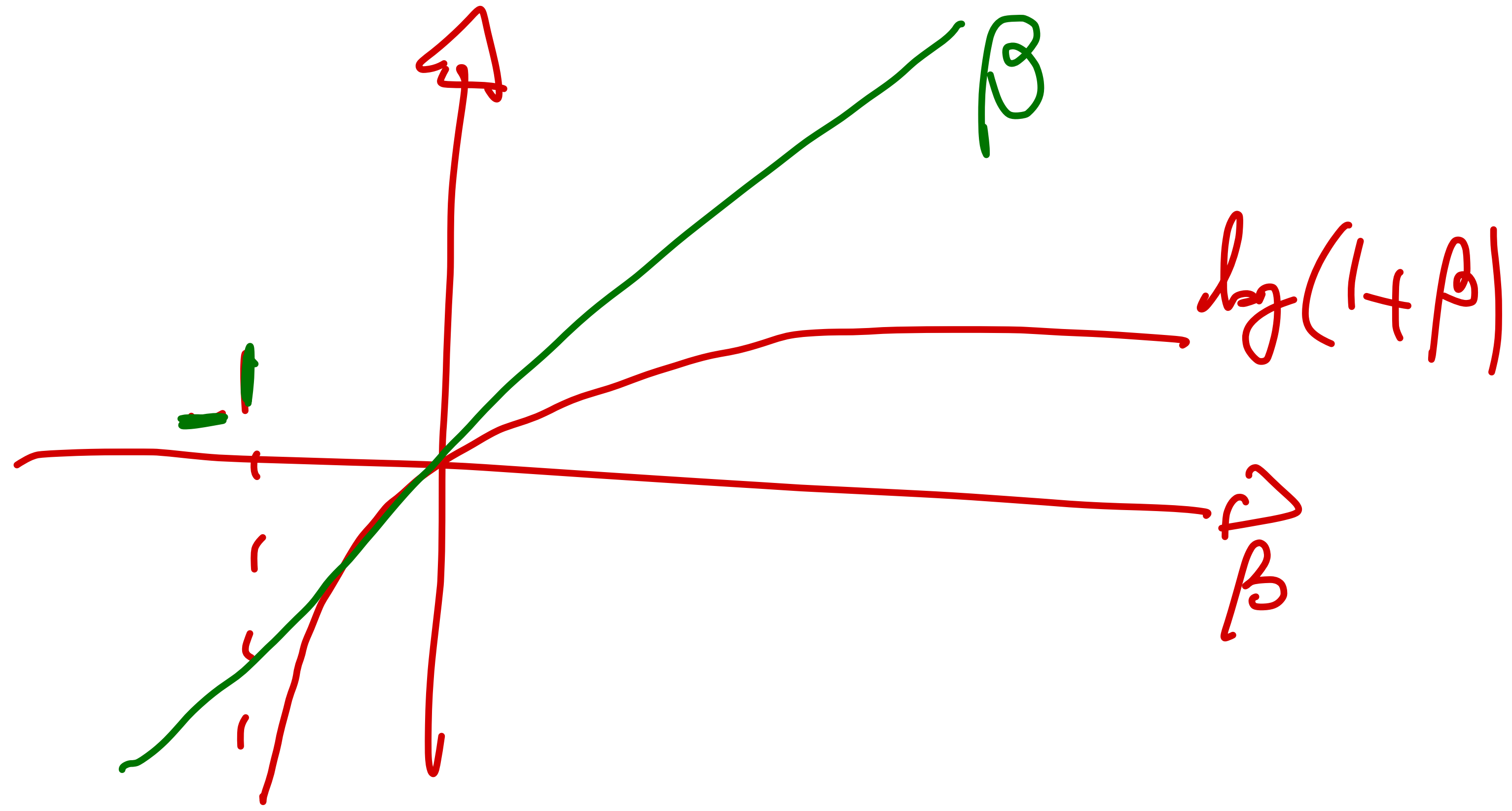
(apply iteratively)

$$\log \mu_k \leq k \log (1 - \delta/n) + \log \mu_0$$

Iteration complexity proof

[Page 402-406, NO]

$$\mu_{k+1} \leq (1 - \delta/n)\, \mu_k$$



(take logarithm)

$$\log \mu_{k+1} \leq \log(1 - \delta/n) + \log \mu_k$$

(apply iteratively)

$$\log \mu_k \leq k \log(1 - \delta/n) + \log \mu_0$$

Since $\log(1+\beta) \leq \beta, \quad \forall \beta > -1$ $\longrightarrow$ $\log(\mu_k/\mu_0) \leq k(-\delta/n)$

Iteration complexity proof

[Page 402-406, NO]

$$\mu_{k+1} \leq (1 - \delta/n)\, \mu_k$$

(take logarithm)

$$\log \mu_{k+1} \leq \log (1 - \delta/n) + \log \mu_k$$

$\longrightarrow$ (apply iteratively)

$$\log \mu_k \leq k \log (1 - \delta/n) + \log \mu_0$$

Since $\log(1+\beta) \leq \beta, \quad \forall \beta > -1$ $\longrightarrow$ $\log(\mu_k/\mu_0) \leq k(-\delta/n)$ ≤ log(ε)

If $k(-\delta/n) \leq \log(\epsilon)$, then $\log(\mu_k/\mu_0) \leq \log(\epsilon)$. Therefore, $\mu_k/\mu_0 \leq \epsilon$

Iteration complexity proof

[Page 402-406, NO]

$$\mu_{k+1} \leq (1 - \delta/n)\, \mu_k$$

(take logarithm)

$$\log \mu_{k+1} \leq \log(1 - \delta/n) + \log \mu_k$$

→ (apply iteratively)

$$\log \mu_k \leq k \log(1 - \delta/n) + \log \mu_0$$

Since $\log(1+\beta) \leq \beta, \quad \forall \beta > -1$ → $\log(\mu_k/\mu_0) \leq k(-\delta/n)$

If $k(-\delta/n) \leq \log(\epsilon)$, then $\log(\mu_k/\mu_0) \leq \log(\epsilon)$. Therefore, $\mu_k/\mu_0 \leq \epsilon$

$k(\delta/n) \geq -\log(\varepsilon)$

Rewriting the inequality: $k \geq (n/\delta) \log(1/\epsilon)$ $\blacksquare$

Interior-point methods for linear optimization

Today, we learned to:

- **Apply** Newton's method to solve optimality conditions
- **Analyze** the central path and the smoothed optimality conditions
- **Develop** a prototype primal-dual path-following algorithm

Next lecture

- Practical interior-point method (Mehrotra predictor-corrector algorithm)
- Linear algebra implementation details
- Linear optimization recap