ORF522 – Linear and Nonlinear Optimization

10. Interior-point methods for linear optimization

Bartolomeo Stellato — Fall 2020

Ed forum

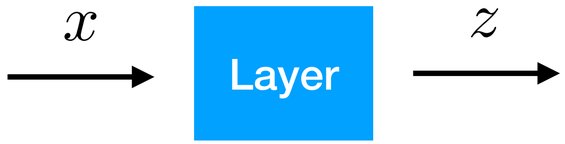
- how do we usually solve for x in f(x) = 0, where f is already given?
- we performed a global sensitivity analysis when the b vector is perturbed, but I do not think we covered the case when the constraint matrix A is changed. Can we extract information on the sensitivities of x* and y* with respect to such changes?
- The Sudoku problem is a discrete constraint satisfaction problem, while we have been looking at linear programs. What was the connection between this discrete version and the linear optimization layer? It appears to be a relaxation of the discrete case, since the constraints are relaxed. Did we discuss how to connect the solution to this relaxed problem back to the discrete case?
- if we want to add both a new variable and a new constraint, should we just add them sequentially using the above approach? Or is there a better way to do this?

Recap

Training a neural network

Single layer model

Datapoints (x^i, y^i)



$$z = f(x; \theta)$$

Training

minimize $\mathcal{L}(\theta) = \sum_{i=1}^{n} \ell(z^i, y^i)$

Gradient descent (more on this later)

$$\theta \leftarrow \theta - t\nabla_\theta \mathcal{L}(\theta)$$

Sensitivity

$$\nabla_\theta \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \theta}\right)^T = \left(\frac{\partial \ell}{\partial z}\frac{\partial z}{\partial \theta}\right)^T = \left(\frac{\partial z}{\partial \theta}\right)^T \nabla_z \mathcal{L}$$

Can f be an **optimization layer**?

Optimality conditions

minimize $c^T x$
subject to $Ax \leq b$

Parameters: $\theta = \{c, A, b\}$
Solution $x^\star(\theta)$

Solve and obtain primal-dual pair $x^\star, y^\star$ (forward-pass)

Optimality conditions

$$A^T y + c = 0$$
$$\mathbf{diag}(y)(Ax - b) = 0$$
$$y \geq 0,\ b - Ax \geq 0$$

Mapping $r(\theta, x(\theta)) = 0$

Computing derivatives

Take differentials

$$\begin{aligned} A^T y^\star + c &= 0 \\ \mathbf{diag}(y^\star)(Ax - b) &= 0 \end{aligned} \longrightarrow \begin{aligned} \mathrm{d}A^T y^\star + A^T \mathrm{d}y &= 0 \\ \mathbf{diag}(Ax - b)\mathrm{d}y + \mathbf{diag}(y^\star)(\mathrm{d}Ax^\star + A\mathrm{d}x - \mathrm{d}b) &= 0 \end{aligned}$$

Linear system

$$\begin{bmatrix} 0 & A^T \\ \mathbf{diag}(y^\star)A & \mathbf{diag}(Ax^\star - b) \end{bmatrix} \begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \end{bmatrix} = - \begin{bmatrix} \mathrm{d}A^T y^\star + \mathrm{d}c \\ \mathbf{diag}(y^\star)(\mathrm{d}Ax^\star - \mathrm{d}b) \end{bmatrix}$$

Example: How does x change with b_1?

Set $\mathrm{d}b = e_1, \mathrm{d}A = 0, \mathrm{d}c = 0$ and solve the linear system.

The solution $\mathrm{d}x$ will correspond to $\dfrac{\partial x}{\partial b}$

Example

Learning to play Sudoku

			3
1			
		4	
4			1

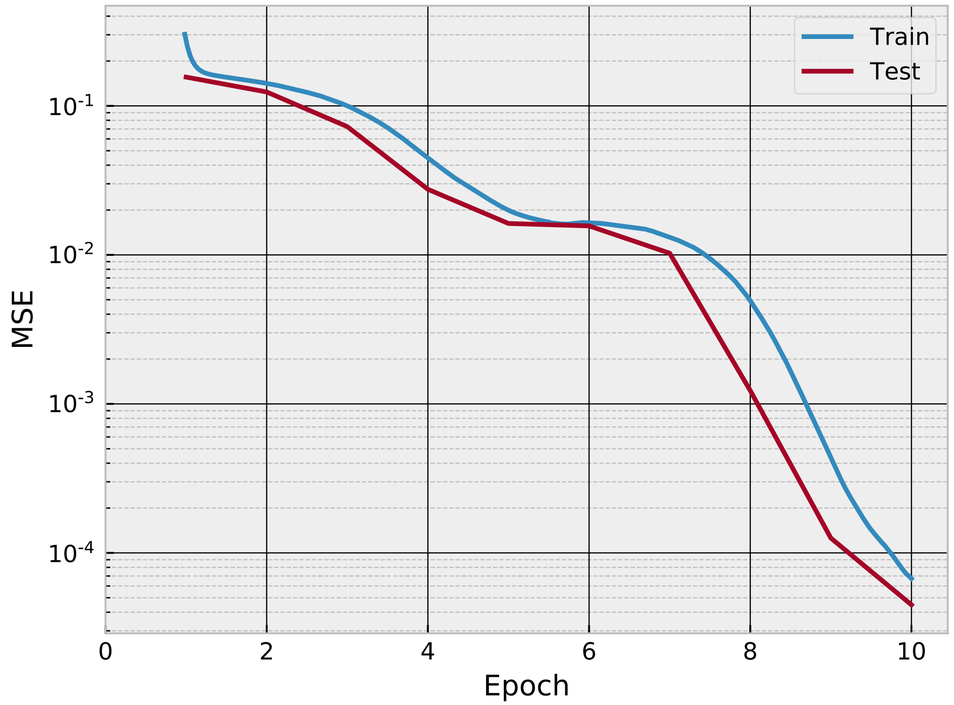
2	4	1	3
1	3	2	4
3	1	4	2
4	2	3	1

Sudoku constraint satisfaction problem

$$\begin{aligned} &\text{minimize} && 0 \\ &\text{subject to} && Ax = b \\ & && x \in \mathbf{Z}^d \end{aligned}$$

Linear optimization layer (parameters $\theta = \{A, b\}$)

$$\begin{aligned} z = x^\star = \quad &\underset{x}{\text{argmin}} && 0 \\ &\text{subject to} && Ax = b \\ & && x \geq 0 \end{aligned}$$



[OptNet: Differentiable Optimization as a Layer in Neural Networks, B. Amos and J. Z. Kolter ICML 2017]

Reference readings for differentiable optimization

[OptNet: Differentiable Optimization as a Layer in Neural Networks, B. Amos and J. Z. Kolter, ICML 2017]

[Differentiating Through a Conic Program, A. Agrawal, Barratt S., Boyd S., Busseti E., and Moursi W. M., Journal of Applied and Numerical Optimization, 2019]

[Differentiable Convex Optimization Layers, A. Agrawal, Amos B., Barratt S., Boyd S., Diamond S., and Kolter Z. NeurIPS 2019]

[Learning Convex Optimization Control Policies, A. Agrawal, Barratt S., Boyd S., Stellato, B., Proceedings MLR 2020]

[Learning Convex Optimization Models, A. Agrawal, Amos B., Barratt S., Boyd S., arXiv:2006.04248, 2020]

And many more…

Today’s lecture

[Chapter 14, Wright][Chapter 17/18 Vanderbei]

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

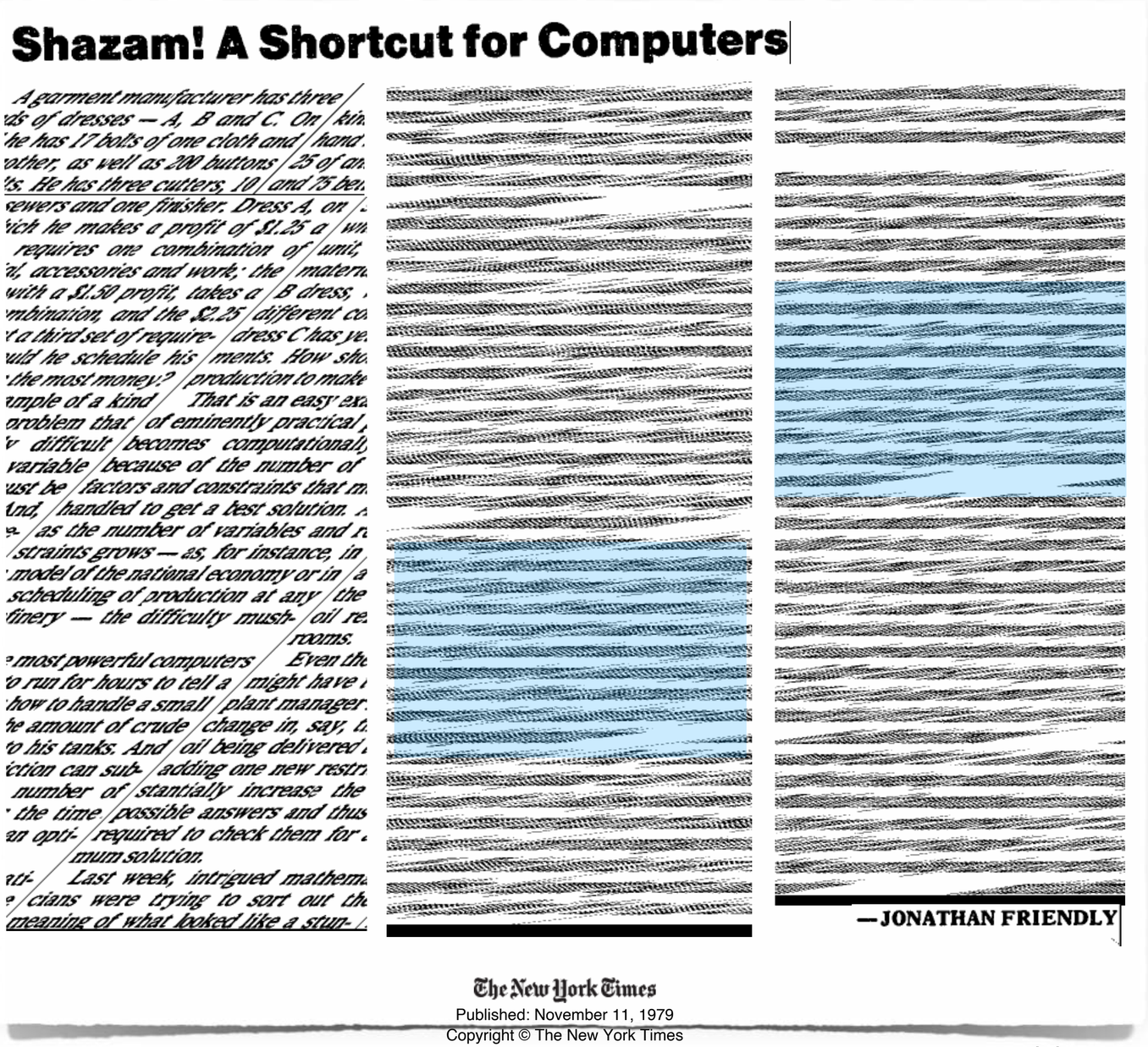
History

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 kinds of dresses — A, B and C. On kind A he has 17 bolts of one cloth and hand other, as well as 200 buttons, 25 of an It's. He has three cutters, 10 and 15 hel sewers and one finisher. Dress A, on which he makes a profit of $1.25 a requires one combination of unit, al, accessories and work; the materi with a $1.50 profit, takes a B dress, mbination, and the $2.25 different co a third set of require- dress C has ye uld he schedule his ments. How sho the most money? production to make ample of a kind That is an easy exa problem that of eminently practical difficult becomes computationall variable because of the number of ust be factors and constraints that m nd, handled to get a best solution. as the number of variables and straints grows — as, for instance, in model of the national economy or in a scheduling of production at any the finery — the difficulty mush- oil re rooms.

most powerful computers Even th to run for hours to tell a might have how to handle a small plant manager he amount of crude change in, say, t to his tanks. And oil being delivered ction can sub- adding one new restr number of stantially increase the the time possible answers and thus an opti- required to check them for mum solution.

Last week, intrigued mathem cians were trying to sort out th meaning of what looked like a stun-

— JONATHAN FRIENDLY

The New York Times

Published: November 11, 1979

Ellipsoid method

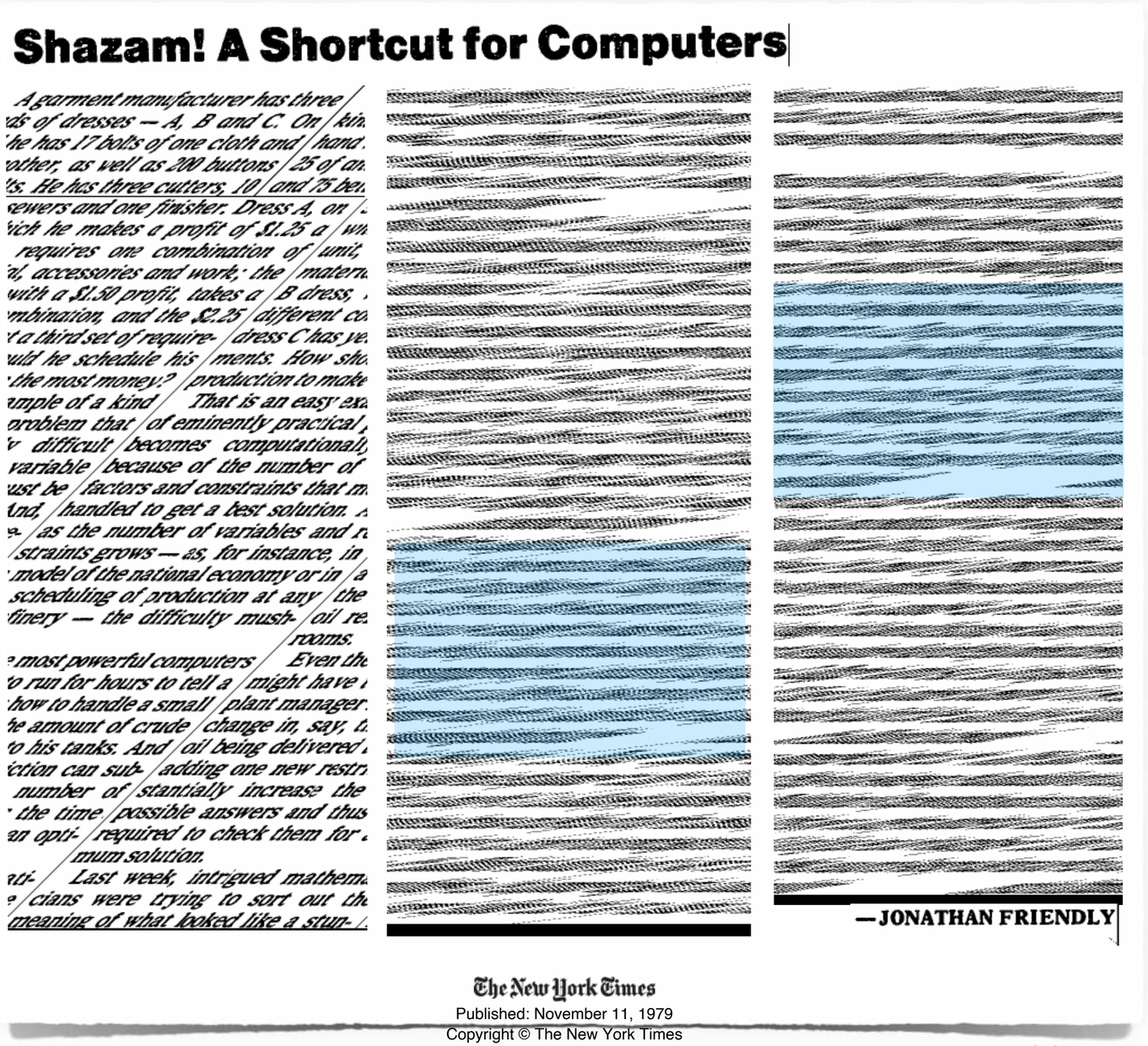
Khachian (1979)

Answer to major question

Is worst-case LP complexity polynomial? Yes!

Drawbacks

Very inefficient. Much slower than simplex!



Shazam! A Shortcut for Computers

A garment manufacturer has three kinds of dresses — A, B and C. On kind A he has 17 bolts of one cloth and 25 of another, as well as 200 buttons and 75 belts. He has three cutters, 10 and 15 sewers and one finisher. Dress A, on which he makes a profit of $1.25 a unit, requires one combination of unit, B dress, with a $1.50 profit, takes a different combination, and the $2.25 dress C has yet a third set of requirements. How should he schedule his production to make the most money?

That is an easy example of a kind of eminently practical problem that becomes computationally difficult because of the number of variables and constraints that must be handled to get a best solution. As the number of variables and constraints grows — as, for instance, in a model of the national economy or in the scheduling of production at any oil refinery — the difficulty mushrooms.

Even the most powerful computers might have to run for hours to tell a plant manager how to handle a small change in, say, the amount of crude oil being delivered to his tanks. And adding one new restriction can substantially increase the number of possible answers and thus the time required to check them for an optimum solution.

Last week, intrigued mathematicians were trying to sort out the meaning of what looked like a stun-

— JONATHAN FRIENDLY

The New York Times
Published: November 11, 1979

Ellipsoid method

Khachian (1979)

Answer to major question

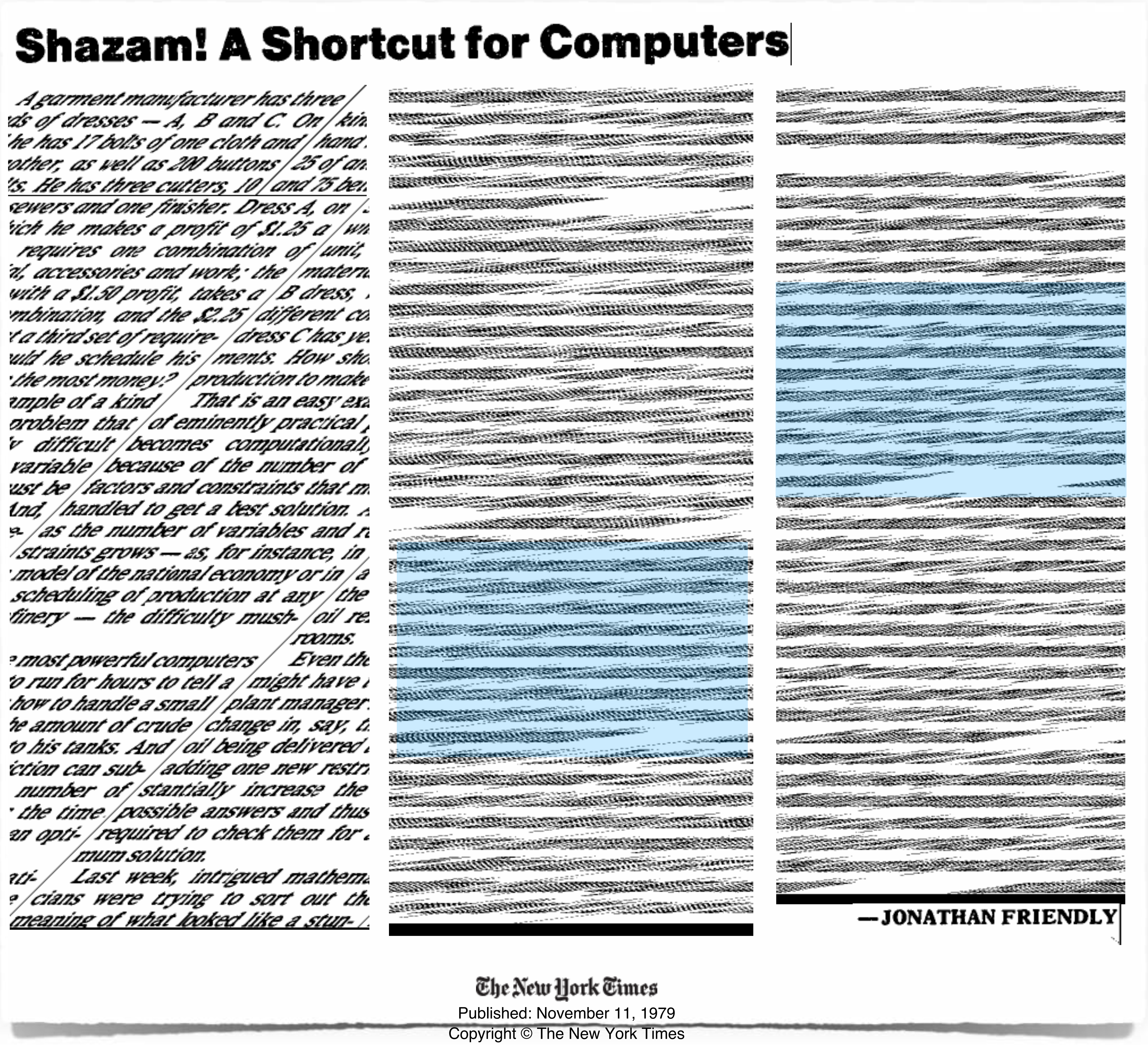
Is worst-case LP complexity polynomial? Yes!

Drawbacks

Very inefficient. Much slower than simplex!

Benefits

Motivated new research directions


Shazam! A Shortcut for Computers

—JONATHAN FRIENDLY

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

"All the News That's Fit to Print"

The New York Times

Late Edition

VOL. CXXXIV . . . No. 46,233 NEW YORK, MONDAY, NOVEMBER 19, 1984 30 CENTS

AMERICANS IN POLL VIEW GOVERNMENT MORE CONFIDENTLY

But Postelection Inquiry Also Finds Most Think Reagan Will Ask Rise in Taxes

By ADAM CLYMER

TREASURY MAY ASK INCREASES IN TAXES OF SOME CONCERNS

OTHERS COULD PAY LESS

Depreciation Setup Would Be Changed — Little Support in Congress Expected

By DAVID E. ROSENBAUM

Albany Leaders Predicting a Cut In Income Taxes

By JOHN HARMANES

Cocaine Traffickers Kill 17 in Peru Raid On Antidrug Team

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

By ALAN COWELL

Breakthrough in Problem Solving

By JAMES GLEICK

Homeless Spend Nights in City Welfare Office

By SARA RIMER

INSIDE

Tug With 5 Aboard Missing

Musicals for City Opera

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

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

Goal: solve

$$h(x) = 0$$

Derivative

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

First-order approximation

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

$\longrightarrow$

Iteratively set to zero

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

Newton's method for nonlinear equations

Goal: solve
$h(x) = 0$

Derivative

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

First-order approximation

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

$\longrightarrow$

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

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 $x^\star$

Symmetric primal-dual problems

minimize $\quad c^T x$

subject to $\quad Ax \leq b$

Symmetric primal-dual problems

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

Symmetric primal-dual problems

minimize $c^T x$
subject to $Ax \leq b$

Primal

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

Dual

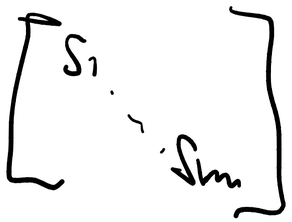
maximize $-b^T y$
subject to $A^T y + c = 0$
$y \geq 0$

Optimality conditions

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

Main idea

$SY\mathbf{1} = s \circ y$



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"

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

Newton's method for optimality conditions

Linear system

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

Residuals

$$r_p = Ax + s - b$$

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

Newton's method for optimality conditions

Linear system

$$\begin{matrix} \\ \\ \end{matrix} \begin{matrix} Dh \\ \\ \end{matrix} \qquad \begin{matrix} \\ \\ \end{matrix} \begin{matrix} -h \\ \\ \end{matrix}$$

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

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

Linear system

$$\begin{matrix} \\ Dh \end{matrix} \qquad \begin{matrix} \\ -h \end{matrix}$$

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

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

$$Ax + s - b = 0$$

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

$$s_i y_i = \tau \quad \longleftarrow \text{ Same } \tau \text{ for every pair}$$

$$s, y \geq 0$$

Same optimality conditions for a “smoothed” version of our problem

Newton's method for smoothed optimality conditions

Smoothed optimality conditions

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

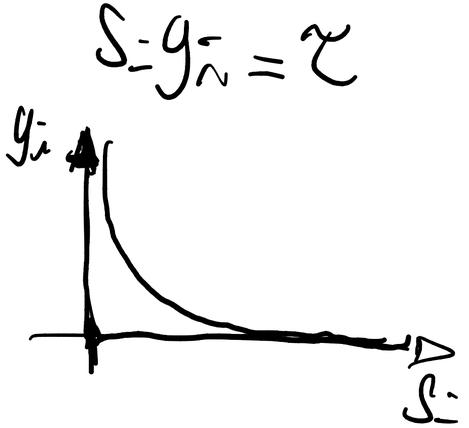
$$s, y \geq 0$$

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

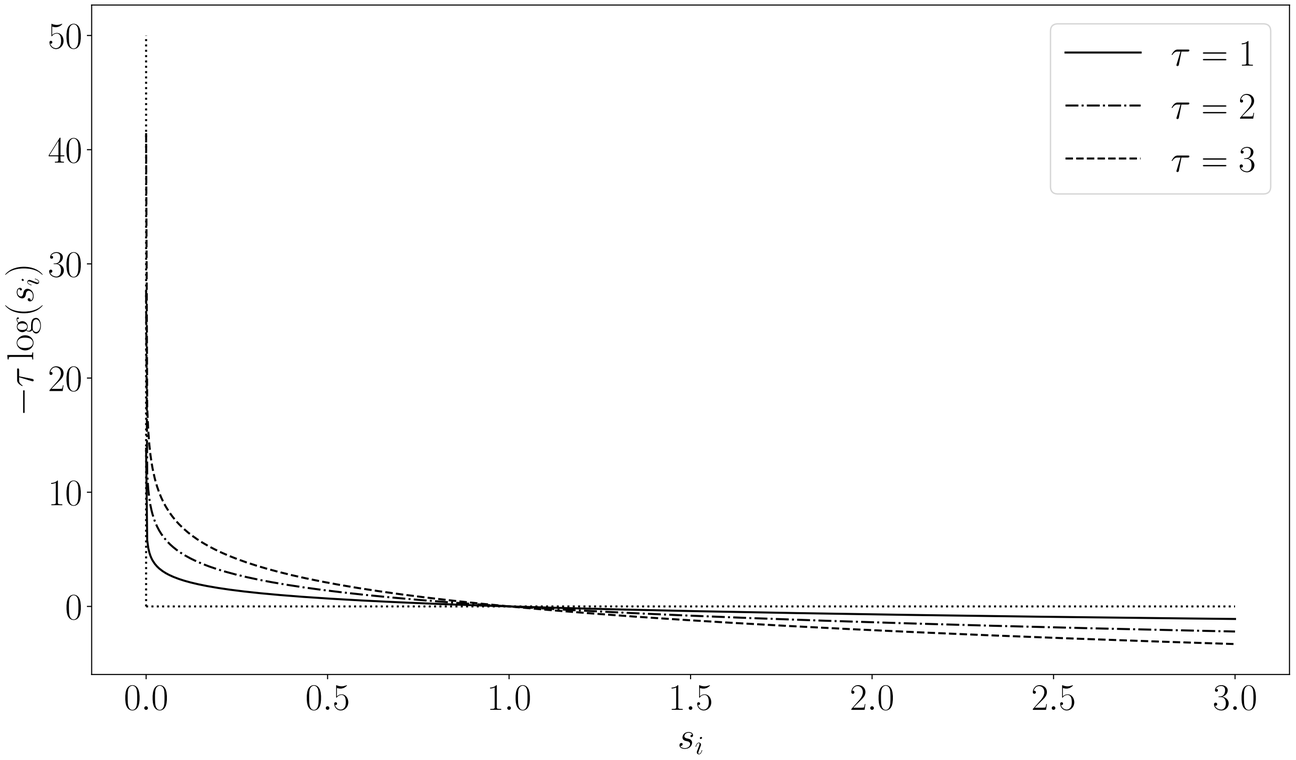
$$\begin{bmatrix} 0 & A & I \\ A^T & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY + \tau\mathbf{1} \end{bmatrix}$$

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

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

Smoothed problem

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

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

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

Dual cost

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

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

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

Central path

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

subject to $Ax + s = b$

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

$$Ax + s - b = 0$$

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

$$s_i y_i = \tau$$

$$s, y \geq 0$$

Central path

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

subject to $Ax + s = b$

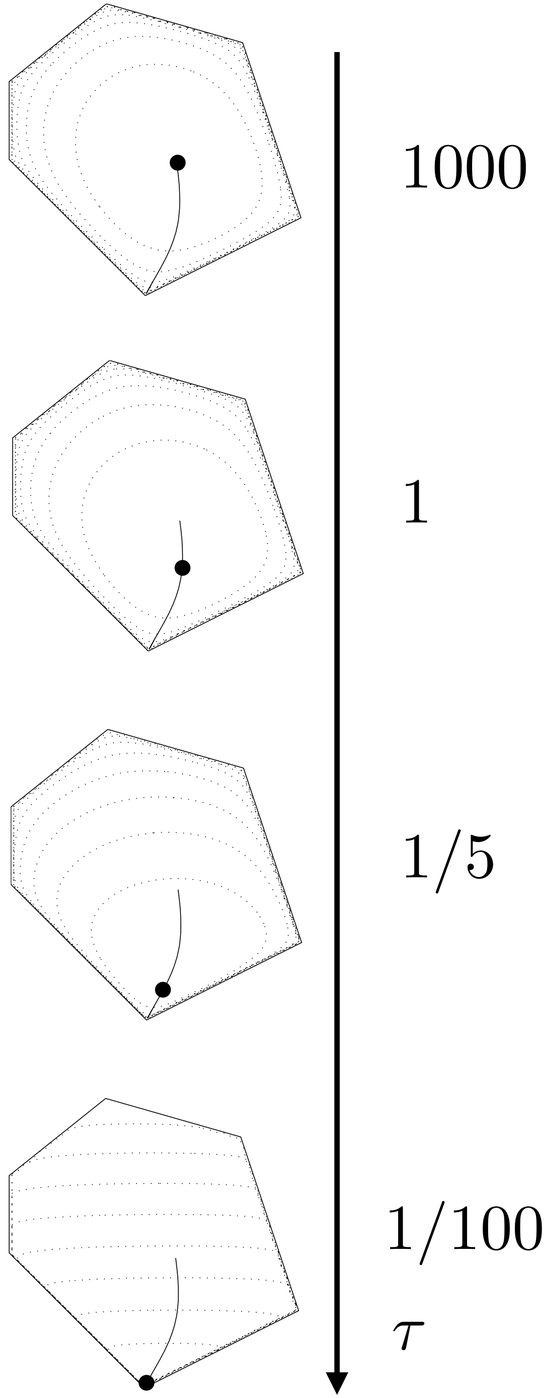
Set of points $(x^\star(\tau), s^\star(\tau), y^\star(\tau))$ with $\tau > 0$ such that

$$Ax + s - b = 0$$

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

$$s_i y_i = \tau$$

$$s, y \geq 0$$



Main idea

Follow central path as $\tau \to 0$

Primal-dual path-following method

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

$\sigma \in [0, 1]$

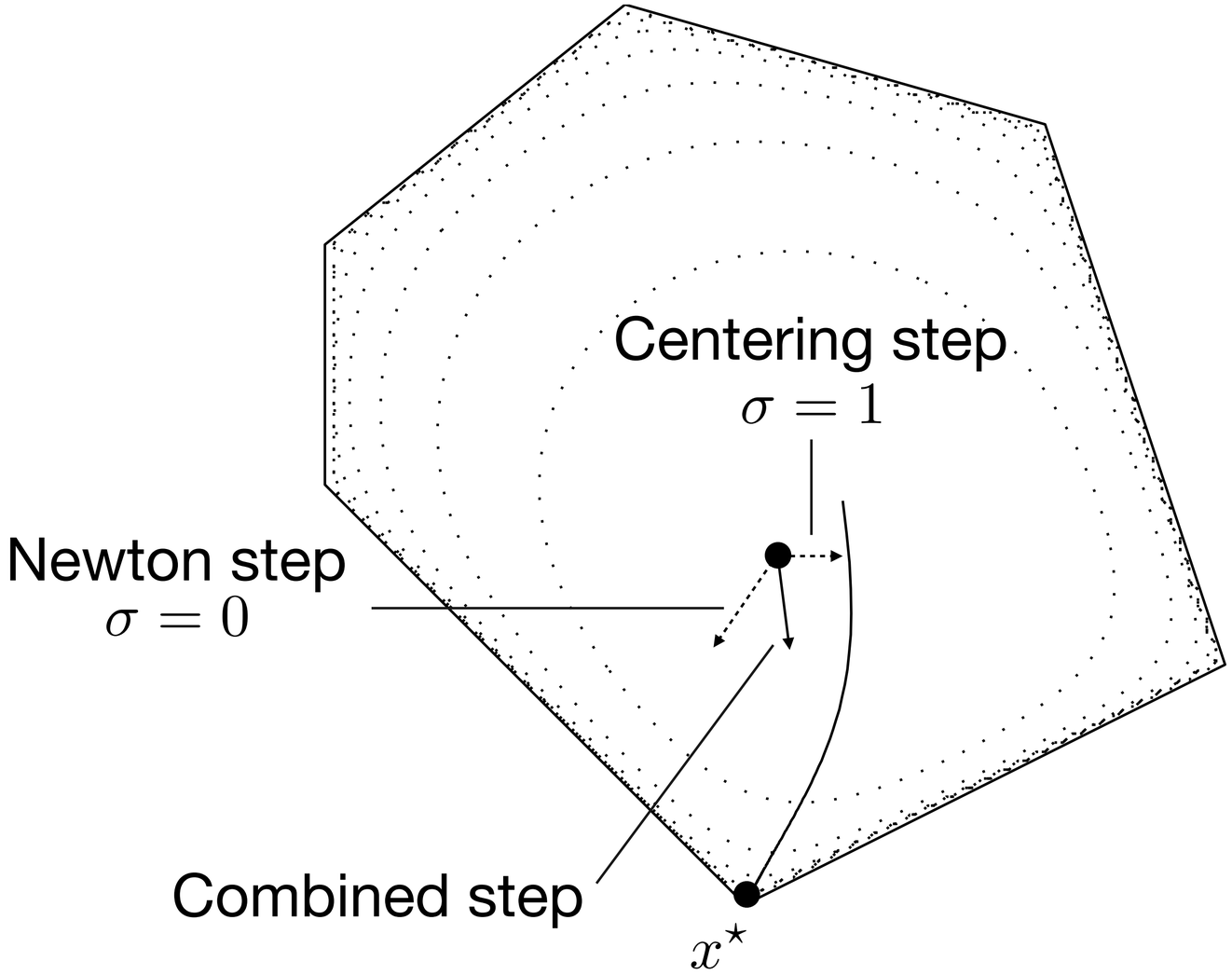
$\sigma = 0 \quad \Rightarrow$ Newton step

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

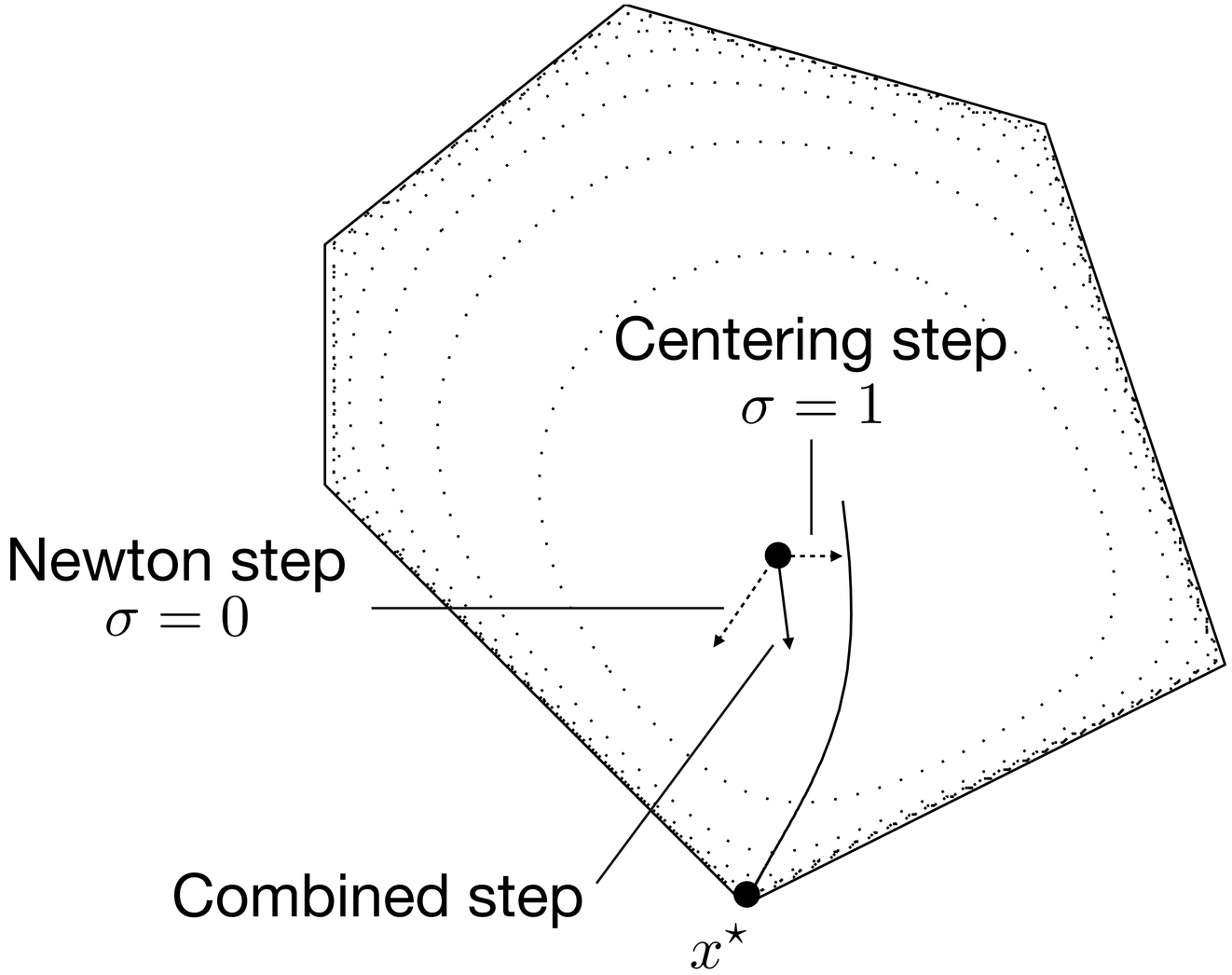
Line search to enforce $x, s > 0$

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

Path-following algorithm idea

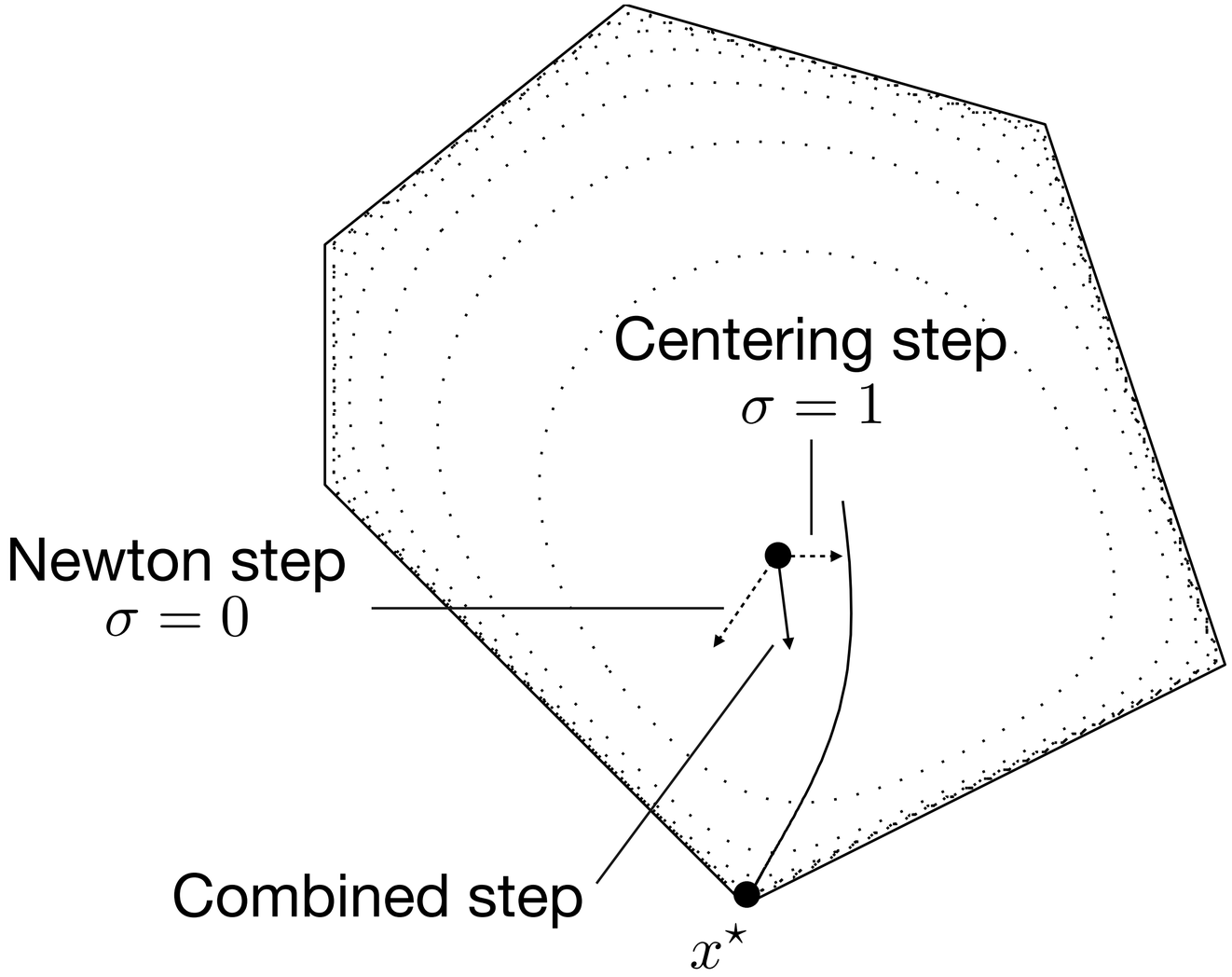


Path-following algorithm idea



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



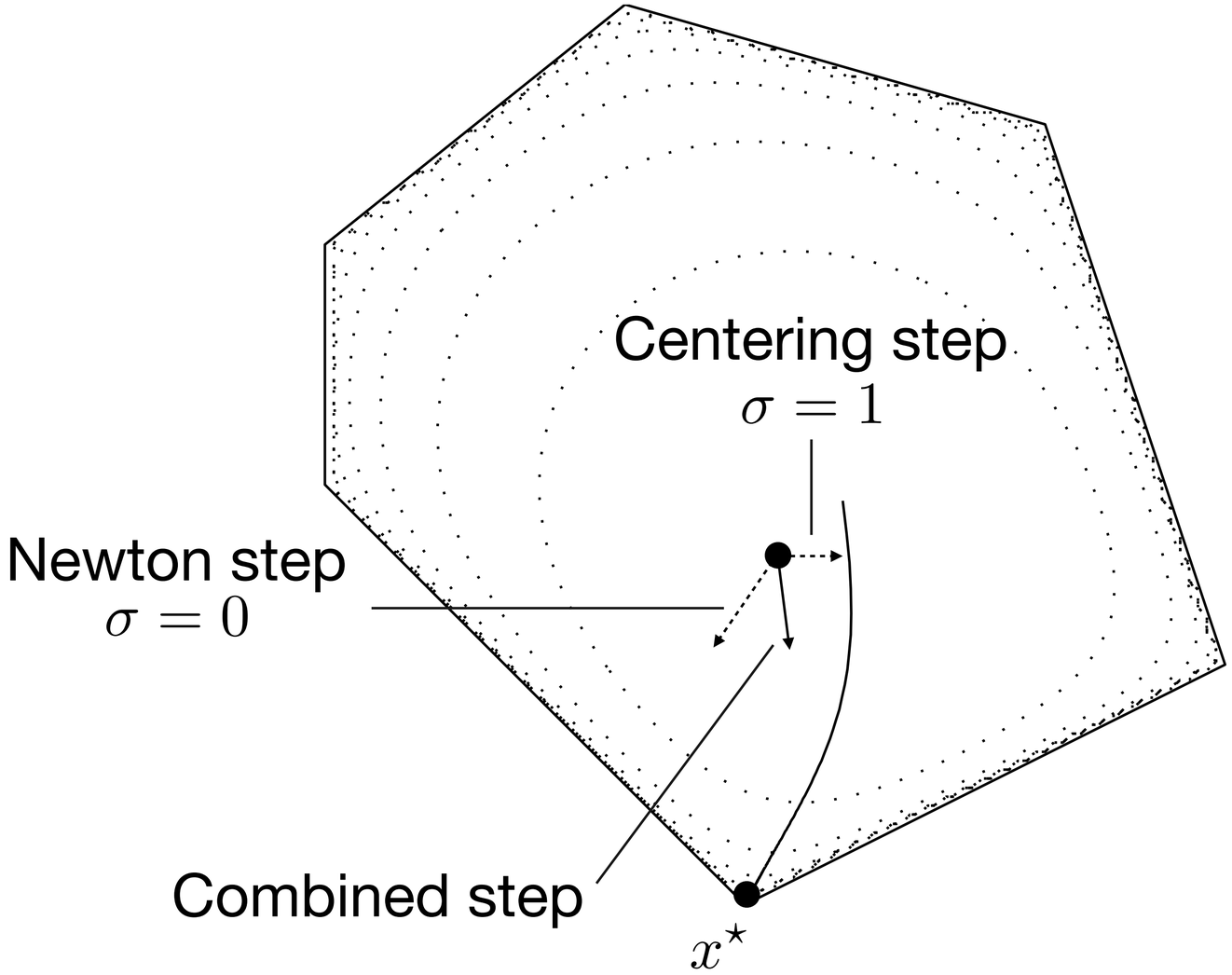
Centering step

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

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

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

Theorem

[Page 402-406, Nocedal Wright]

Smallest decrement

$$\mu_{k+1} \leq \left(1 - \frac{\delta}{m}\right) \mu_k \qquad \text{with } \delta > 0$$

Theorem

[Page 402-406, Nocedal Wright]

Smallest decrement

$$\mu_{k+1} \leq \left(1 - \frac{\delta}{m}\right) \mu_k \qquad \text{with } \delta > 0$$

Iteration complexity

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

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

Theorem

[Page 402-406, Nocedal Wright]

Smallest decrement

$$\mu_{k+1} \leq \left(1 - \frac{\delta}{m}\right) \mu_k \qquad \text{with } \delta > 0$$

Iteration complexity

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

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

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

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