Primal-Dual Interior-Point methods for Linear Programming (and beyond)

Mathieu Tanneau

Georgia Institute of Technology

March 2, 2021

POLYTECHNIQUE MONTRÉAL

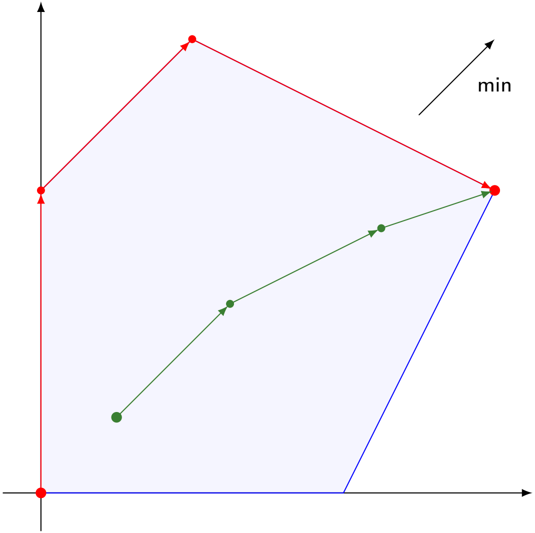
CANADA EXCELLENCE RESEARCH CHAIR DATA SCIENCE FOR REAL-TIME DECISION-MAKING

Georgia Tech

1. Foreword
2. The "barrier" approach
3. Theory of IPMs
4. Linear Algebra in IPMs
5. Conclusion
6. Beyond Linear Programming

Simplex
- Many cheap iterations
- Extreme (basic) points

Interior-Point
- Few costly iterations
- Interior points



How can we solve $\min\{\mathbf{x} \in \mathbb{R} \mid \mathbf{x} \geq 0\}$?

Define

$$f_\mu(\mathbf{x}) = \mathbf{x} - \mu \underbrace{\ln(\mathbf{x})}_{\text{"barrier"}}$$

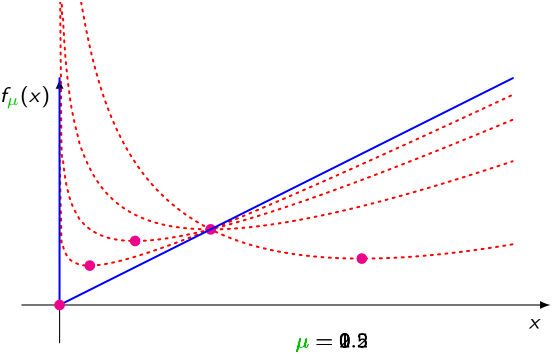
If $\mu > 0$ is small, then $f_\mu(\mathbf{x}) \simeq \mathbf{x}$.

We have (for $\mathbf{x} > 0$)

$$f'_\mu(\mathbf{x}) = 1 - \mu\frac{1}{\mathbf{x}} \quad \longrightarrow \quad f'_\mu(\mathbf{x}) = 0 \Leftrightarrow \mathbf{x} = \mu$$

$$f''_\mu(\mathbf{x}) = \frac{\mu}{\mathbf{x}^2} \quad \longrightarrow \quad f_\mu \text{ is convex!}$$

$$f_{\mu} = \mathbf{x} - \mu \ln(\mathbf{x})$$



How can we solve

$$\begin{aligned} \min_{\mathbf{x}} \quad & 2\mathbf{x}_1 + 3\mathbf{x}_2 \\ s.t. \quad & \mathbf{x}_1 + \mathbf{x}_2 = 1, \\ & \mathbf{x}_1, \mathbf{x}_2 \geq 0 \end{aligned}$$

Barrier problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & 2\mathbf{x}_1 + 3\mathbf{x}_2 - \mu \ln \mathbf{x}_1 - \mu \ln \mathbf{x}_2 \\ s.t. \quad & \mathbf{x}_1 + \mathbf{x}_2 = 1, \end{aligned}$$

$\longrightarrow$ no easy closed-form solution!

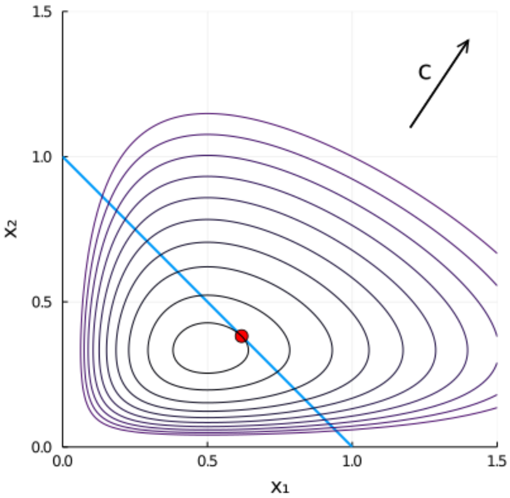


Figure: $\mu = 1.0$

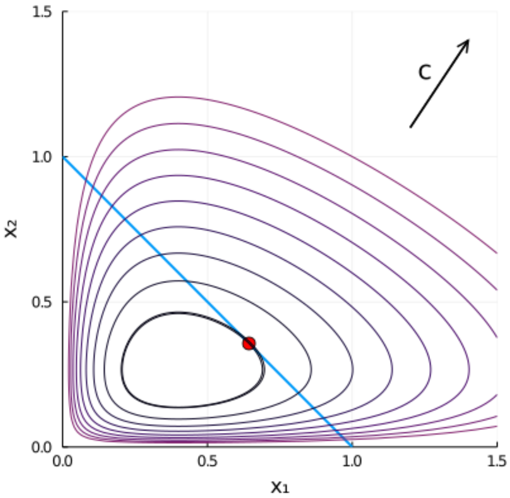


Figure: $\mu = 0.8$

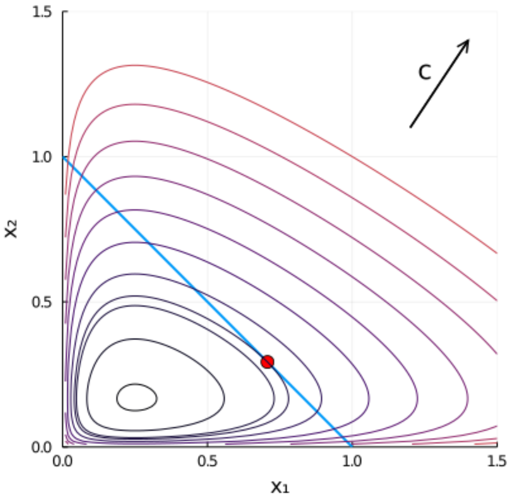


Figure: $\mu = 0.5$

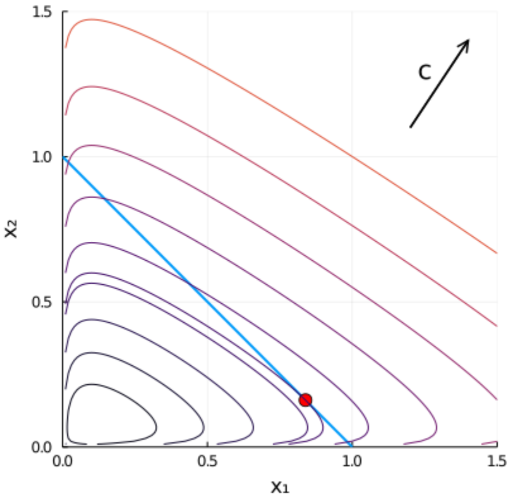


Figure: $\mu = 0.2$

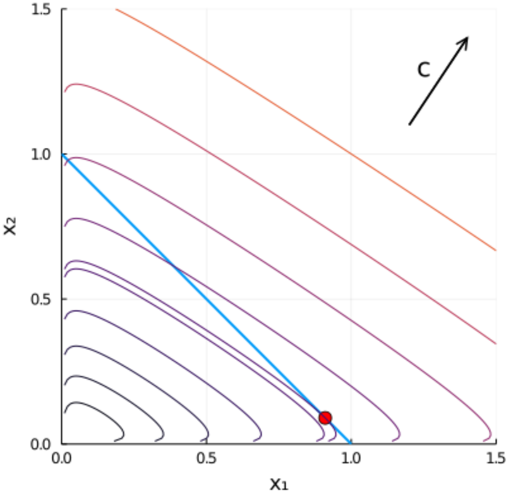


Figure: $\mu = 0.1$

Original problem (LP)

$$\begin{aligned} \min_{\mathbf{x}} \quad & c^T \mathbf{x} \\ s.t. \quad & A\mathbf{x} = b \\ & \mathbf{x} \geq 0 \end{aligned}$$

Barrier problem (convex)

$$\begin{aligned} \min_{\mathbf{x}} \quad & c^T \mathbf{x} - \mu \underbrace{\sum_{j=1}^{n} \ln(\mathbf{x}_j)}_{barrier} \\ s.t. \quad & A\mathbf{x} = b \end{aligned}$$

As μ goes to zero: the sequence of $\mathbf{x}_\mu$ defines a *central path*

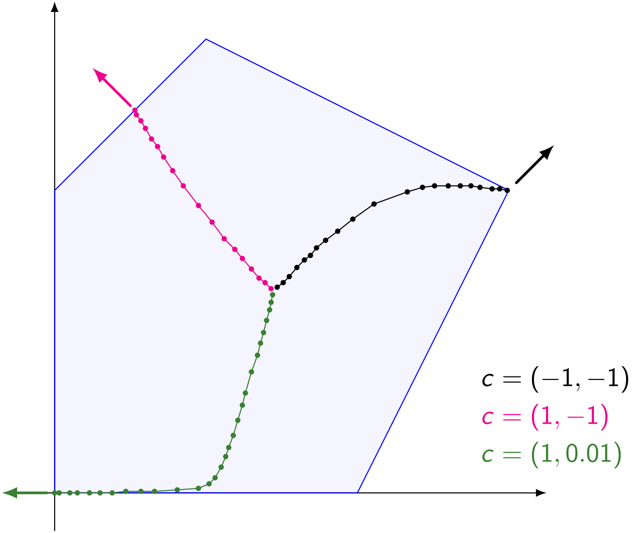


Figure: Central path - examples

Primal-dual pair of LPs

$$
(P) \quad \min_{\mathbf{x}} \quad c^T \mathbf{x} \quad s.t. \quad A\mathbf{x} = b, \quad \mathbf{x} \geq 0
\qquad\qquad
(D) \quad \max_{\mathbf{y},\mathbf{s}} \quad b^T \mathbf{y} \quad s.t. \quad A^T\mathbf{y} + \mathbf{s} = c, \quad \mathbf{s} \geq 0
$$

KKT optimality conditions:

$$A\mathbf{x} = b \quad \text{[primal feas.]} \tag{1}$$

$$A^T\mathbf{y} + \mathbf{s} = c \quad \text{[dual feas.]} \tag{2}$$

$$\forall i, \ \mathbf{x}_i \cdot \mathbf{s}_i = 0 \quad \text{[slackness]} \tag{3}$$

$$\mathbf{x}, \mathbf{s} \geq 0 \tag{4}$$

Barrier problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_{\mu}(\mathbf{x}) \\ s.t. \quad & A\mathbf{x} = b \end{aligned}$$

At the optimum: $\nabla f_{\mu}(\mathbf{x}) = c - \mu \frac{1}{\mathbf{x}} = A^T \mathbf{y}$ for some $\mathbf{y} \in \mathbb{R}^m$.
(∇f_{μ} and gradient of constraints are linearly dependent)

Let $\mathbf{s}_j = \mu / \mathbf{x}_j$, we have

$$\begin{aligned} A\mathbf{x} &= b, \\ A^T \mathbf{y} + \mathbf{s} &= c, \\ \mathbf{x}_j \cdot \mathbf{s}_j &= \mu, \quad j = 1, ..., n \\ \mathbf{x}, \mathbf{s} &\geq 0. \end{aligned}$$

Newton's method for solving $f(x) = 0$

Start from x_0

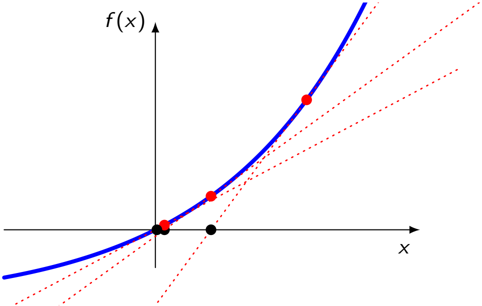
- $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$
- ...
- $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$

until $f(x_n) \simeq 0$.

At each iteration:

$$0 = f(x_n) + \underbrace{(x_{n+1} - x_n)}_{\Delta x} \times f'(x_n)$$

Newton's method for solving $f(x) = e^x - 1 = 0$



Idea: apply Newton method to the KKT system (1)-(4)

$$F(\mathbf{x}, \mathbf{y}, \mathbf{s}) := \begin{bmatrix} A\mathbf{x} - b \\ A^T\mathbf{y} + \mathbf{s} - c \\ XSe - \mu e \end{bmatrix}$$

so that $(1)-(4) \Leftrightarrow F(\mathbf{x}, \mathbf{y}, \mathbf{s}) = 0$.

Newton direction Δ given by

$$\underbrace{\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix}}_{\nabla F(\mathbf{x},\mathbf{y},\mathbf{s})} \cdot \underbrace{\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix}}_{\Delta} = \underbrace{\begin{bmatrix} b - A\mathbf{x} \\ c - A^T\mathbf{y} - \mathbf{s} \\ \mu e - XSe \end{bmatrix}}_{-F(\mathbf{x},\mathbf{y},\mathbf{s})} \quad (5)$$

Affine-scaling direction (predictor) given by

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} b - A\mathbf{x} \\ c - A^T\mathbf{y} - \mathbf{s} \\ -XSe \end{bmatrix}$$

Centering direction towards μ-center given by

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} b - A\mathbf{x} \\ c - A^T\mathbf{y} - \mathbf{s} \\ -XSe + \mu e \end{bmatrix}$$

Predictor-Corrector algorithm: combine the two

- Predictor step: make progress towards optimality (decrease μ)
- Corrector step: improve centrality (stay close to central path)

Mehrotra's Predictor-Corrector algorithm [Mehrotra, 1992]

Predictor step Δ^{aff}

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x^{aff} \\ \Delta y^{aff} \\ \Delta s^{aff} \end{bmatrix} = \begin{bmatrix} b - A\mathbf{x} \\ c - A^T\mathbf{y} - \mathbf{s} \\ -XSe \end{bmatrix}$$

Centering-corrector step Δ^{cc}

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x^{cc} \\ \Delta y^{cc} \\ \Delta s^{cc} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma\mu e - \Delta X^{aff} \Delta S^{aff} \end{bmatrix}$$

where:

$$\mu = \frac{1}{n}\mathbf{x}^T\mathbf{s}, \quad \mu^{aff} = \frac{1}{n}(\mathbf{x} + \alpha^{aff}\Delta x^{aff})^T(\mathbf{s} + \alpha^{aff}\Delta s^{aff}), \quad \sigma = (\mu^{aff}/\mu)^3$$

Combined direction:

$$\Delta = \Delta^{aff} + \Delta^{cc}$$

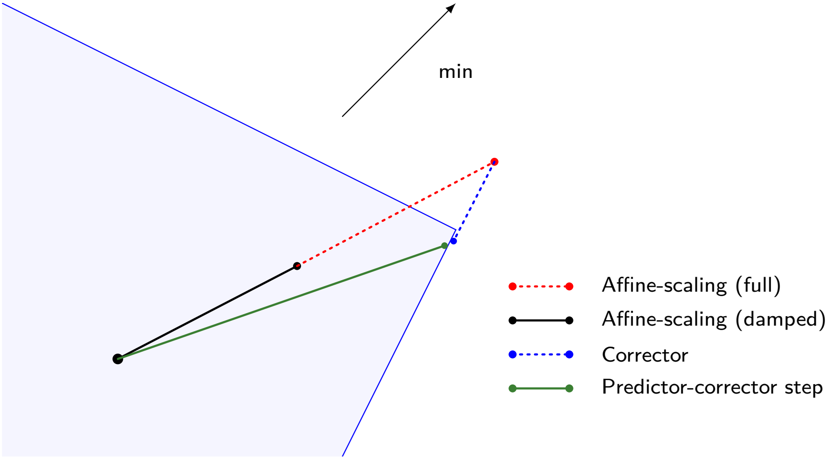


Figure: Mehrotra's Predictor-Corrector, in **x** space

- LP in standard Primal-Dual form

$$(P) \quad \min_{\mathbf{x}} \quad c^T\mathbf{x} \qquad s.t. \quad A\mathbf{x} = b, \quad \mathbf{x} \geq 0$$

$$(D) \quad \max_{\mathbf{y},\mathbf{s}} \quad b^T\mathbf{y} \qquad s.t. \quad A^T\mathbf{y} + \mathbf{s} = c, \quad \mathbf{s} \geq 0$$

- IPMs apply Newton's method to the KKT conditions

$$\begin{aligned} A\mathbf{x} &= b, \quad \mathbf{x} \geq 0, & \text{[primal feas.]} \\ A^T\mathbf{y} + \mathbf{s} &= c, \quad \mathbf{s} \geq 0, & \text{[dual feas.]} \\ \mathbf{x}_j \cdot \mathbf{s}_j &= \mu, \quad \forall j & \text{[slackness]} \end{aligned}$$

$\rightarrow$ solve a few linear systems

- Polynomial-time algorithm (see [Wright, 1997])
- Very efficient on large-scale problems

At each IPM iteration: solve a (large) linear system

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_d \\ \xi_p \\ \xi_{xs} \end{bmatrix}$$

$\longrightarrow$ typically 60−90% of total time!

Two ways to make an Interior-Point faster:

- Reduce the number of iterations (better algorithm)
- Reduce the time per iteration (better linear algebra)

Initial Newton system:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_d \\ \xi_p \\ \xi_{xs} \end{bmatrix}$$

Substitute Δs to obtain the **Augmented system**

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_{xs} \\ \xi_p \end{bmatrix}$$

$$\Delta s = X^{-1}(\xi_{xs} - S\Delta x)$$

where $\Theta := XS^{-1}$

:(Left-hand matrix is indefinite (though regularization can be used)

:(Still costly to solve

:) More handy if free variables and/or non-linear terms

Substitute Δx to obtain the **Normal equations**

$$(A\Theta A^T)\Delta y = \xi_p + A\Theta(\xi_d - X^{-1}\xi_{\mathbf{xs}})$$
$$\Delta x = \Theta(A^T \Delta y - \xi_d + X^{-1}\xi_{\mathbf{xs}})$$
$$\Delta s = X^{-1}(\xi_{\mathbf{xs}} - S\Delta x)$$

:) $A\Theta A^T$ is symmetric positive-definite...

:(... but dense if A has dense columns

$\Longrightarrow$ We now focus on solving $(A\Theta A^T)\Delta y = \xi$

We want to solve

$$\underbrace{(A\Theta A^T)}_{S} \Delta y = \xi$$

S is symmetric positive definite,
we can compute its **Cholesky factorization**

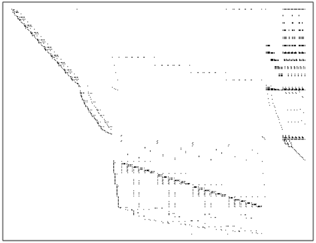
$$S = L \times L^T$$

where L is lower triangular with positive diagonal.

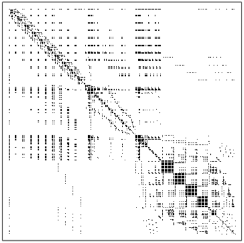
André Cholesky (X1895)

Available libraries: SuiteSparse, MUMPS, Pardiso, etc.

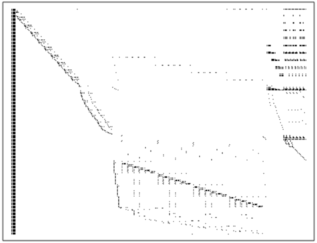
Sparsity pattern of A



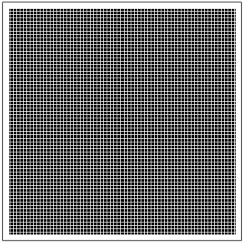
Sparsity pattern of S



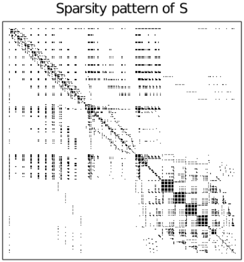
Sparsity pattern of A

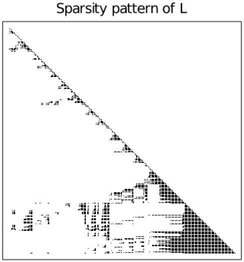


Sparsity pattern of S



One dense column in $A \Longrightarrow A\Theta A^T$ is fully dense





Cholesky factor L has more non-zeros than $S \longrightarrow$ fill-in

Linear algebra in IPMs

Newton system $\longrightarrow$ Normal equations $\longrightarrow$ Cholesky

- :) Numerically stable, parallelizable
- :) Readily available implementations
- :(Can consume a lot of memory

Other options:

- Exploit structure in A
- Iterative methods (e.g. conjugate gradient)

Hands-on: going through a barrier log

Takeaway:

- Polynomial complexity, efficient in practice
- Robust behaviour: $20 - 30$ iterations, regardless of problem size
- Costly linear algebra, but well-suited to parallelism

Ongoing research:

- Warm-start [Gondzio and Gonzalez-Brevis, 2015], [Engau and Anjos, 2017]
- Mixed Integer Programming
- Parallelization [Gondzio and Grothey, 2009]

Questions?

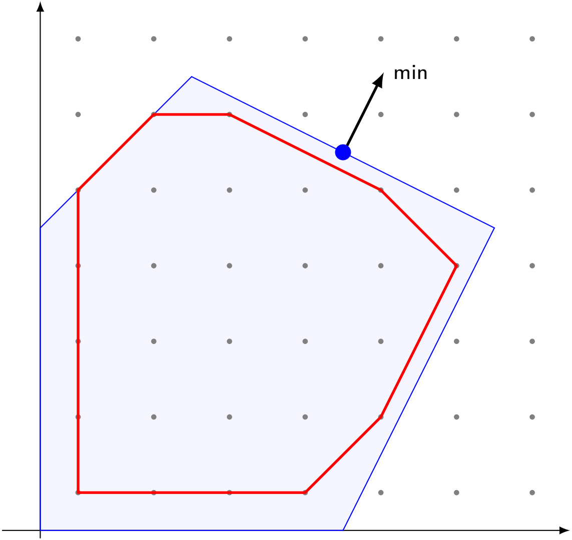


Figure: Generate cut from centred point?

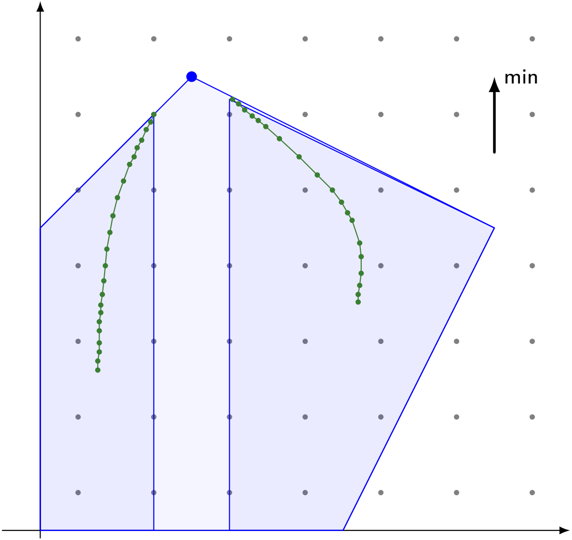


Figure: Central path is heavily affected by branching

(Convex) Quadratic Program:

$$(QP) \quad \min_{\mathbf{x}} \quad \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + c^T \mathbf{x}$$
$$s.t. \quad A\mathbf{x} = b$$
$$\mathbf{x} \geq 0$$

Augmented system for QP:

$$\begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_{\mathbf{xs}} \\ \xi_p \end{bmatrix}$$

(Convex) Non-Linear Program:

$$(NLP) \quad \min_{\mathbf{x}} \quad f(\mathbf{x}) \qquad s.t. \quad g(\mathbf{x}) \leq 0$$

Augmented system for NLP:

$$\begin{bmatrix} Q(\mathbf{x}, \mathbf{y}) & A(\mathbf{x})^T \\ A(\mathbf{x}) & -ZY^{-1} \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(\mathbf{x}) - A(\mathbf{x})^T \mathbf{y} \\ -g(\mathbf{x}) - \mu Y^{-1} e \end{bmatrix}$$

where

$$A(\mathbf{x}) = \nabla g(\mathbf{x})$$

$$Q(\mathbf{x}, \mathbf{y}) = \nabla^2 f(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{y}_i \nabla^2 g(\mathbf{x})$$

Engau, A. and Anjos, M. F. (2017).
Convergence and polynomiality of primal-dual interior-point algorithms for linear programming with selective addition of inequalities.
Optimization, 66(12):2063–2086.

Gondzio, J. and Gonzalez-Brevis, P. (2015).
A new warmstarting strategy for the primal-dual column generation method.
Mathematical Programming, 152(1):113–146.

Gondzio, J. and Grothey, A. (2009).
Exploiting structure in parallel implementation of interior point methods for optimization.
Computational Management Science, 6(2):135–160.

Mehrotra, S. (1992).
On the implementation of a primal-dual interior point method.
SIAM Journal on Optimization, 2(4):575–601.

Wright, S. (1997).
Primal-Dual Interior-Point Methods.
Society for Industrial and Applied Mathematics.