

Interior Point Algorithms for Large-Scale Nonlinear Programming: Theory and Algorithmic Development

Andreas Wächter

IBM T.J. Watson Research Center

andreasw@watson.ibm.com

Industrial Optimization Seminar

Fields Institute, Toronto

March 1, 2005

Nonlinear Programming

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & d(x) \geq 0 \end{aligned}$$

x	Variables
$f(x)$: $\mathbb{R}^n \rightarrow \mathbb{R}$	Objective function
$c(x)$: $\mathbb{R}^n \rightarrow \mathbb{R}^m$	Equality constraints
$d(x)$: $\mathbb{R}^n \rightarrow \mathbb{R}^p$	Inequality constraints

- Goal: Numerical method for finding *local* solution x_*
- Local solution x_* : Exists neighborhood U of x_* so that
$$\forall x \in U : \quad x \text{ feasible} \implies f(x) \geq f(x_*)$$
- Functions $f(x)$, $c(x)$, $d(x)$ are smooth (C^2)
- The variables x are “continuous” ($x \in \mathbb{R}^n$)
(Reformulating “ $y \in \{0, 1\}$ ” as “ $y(y - 1) = 0$ ” usually doesn’t work)

Applications

- Nonlinear programming problems (NLPs) arise in many practice applications
 - Design (air planes, chemical plants, ...) (PDEs, ODEs)
 - Optimal control
 - Parameter estimation (inverse problems)
 - Stochastic optimization (portfolio analysis)
 - Tuning of digital circuits
 - ...
- see next talk, and talks by Andrew Conn/Chandu Visweswariah (May)
- Two extremes:
 - Computationally expensive function evaluations:
⇒ Derivative Free Optimization (few variables)
 - Analytic functions, can compute derivatives:
Large number of variables (*n* up to several million)

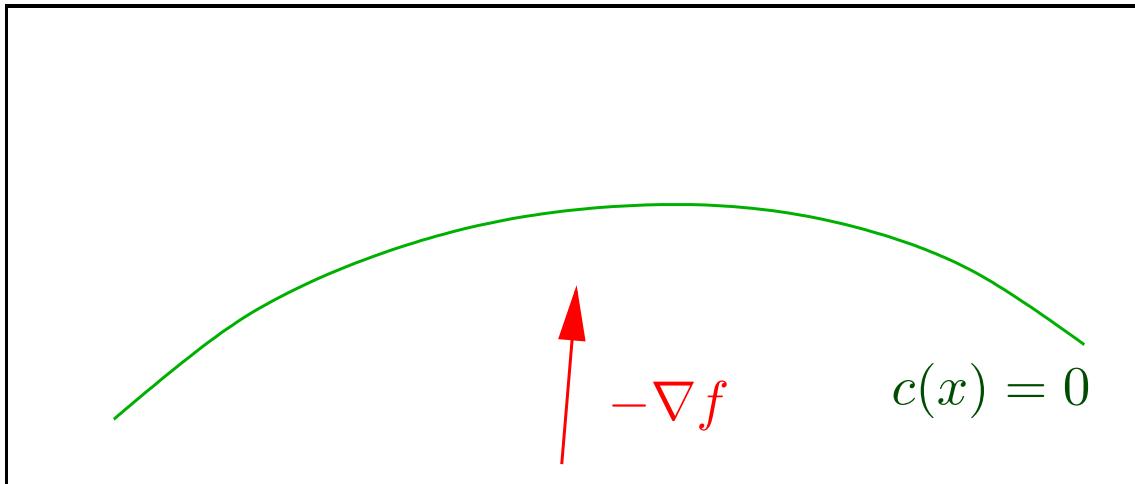
Outline

- Background on nonlinear programming (NLP)
 - Optimality conditions
- Basic primal-dual framework for interior point methods for NLP
- Globalization
 - Failure of convergence in the non-convex case
 - Filter line search procedure
- Adaptive choice of the barrier parameter
 - Quality function
- Numerical results

Optimality Conditions (Crash Course)

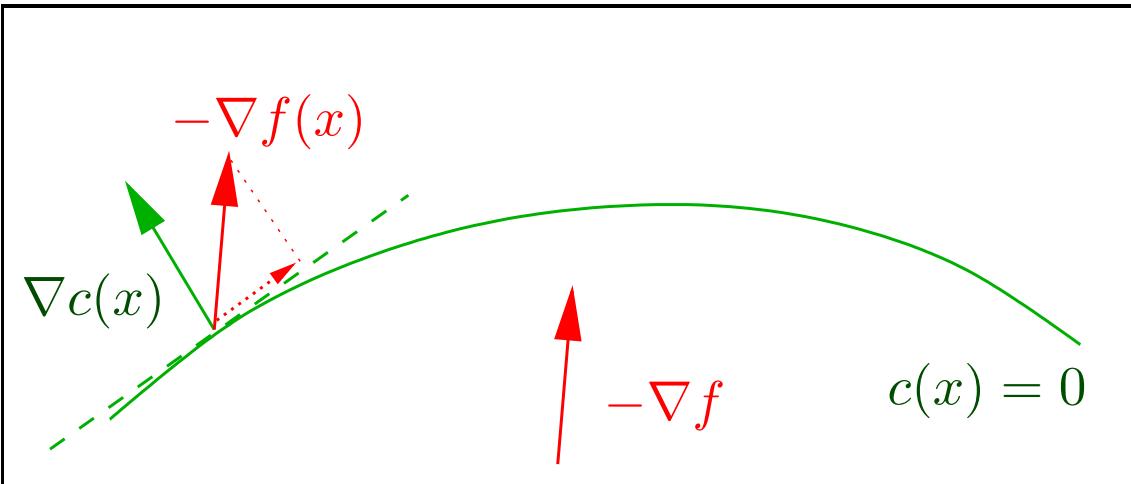
$$\min_{x \in \mathbb{R}^n} f(x)$$

$$s.t. \quad c(x) = 0$$



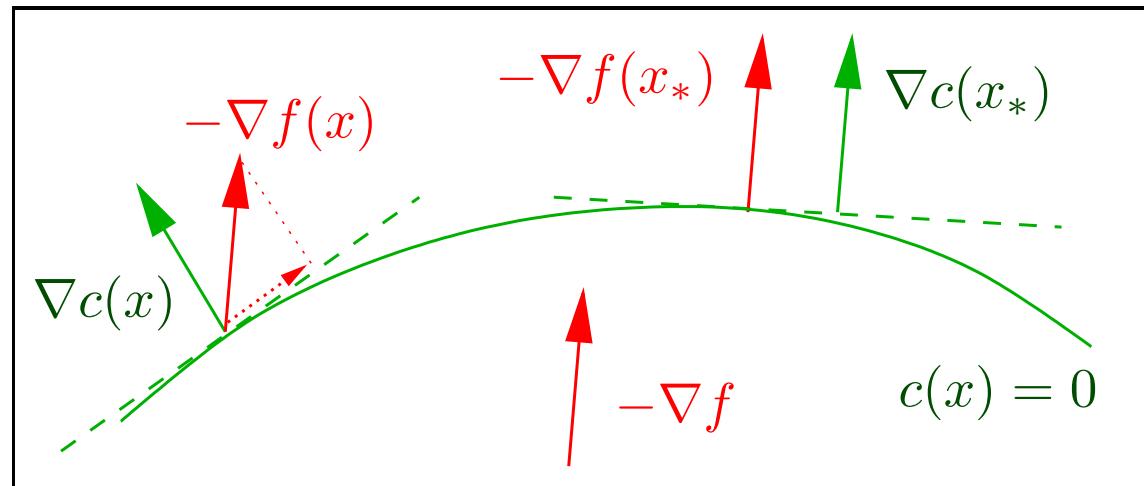
Optimality Conditions (Crash Course)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$



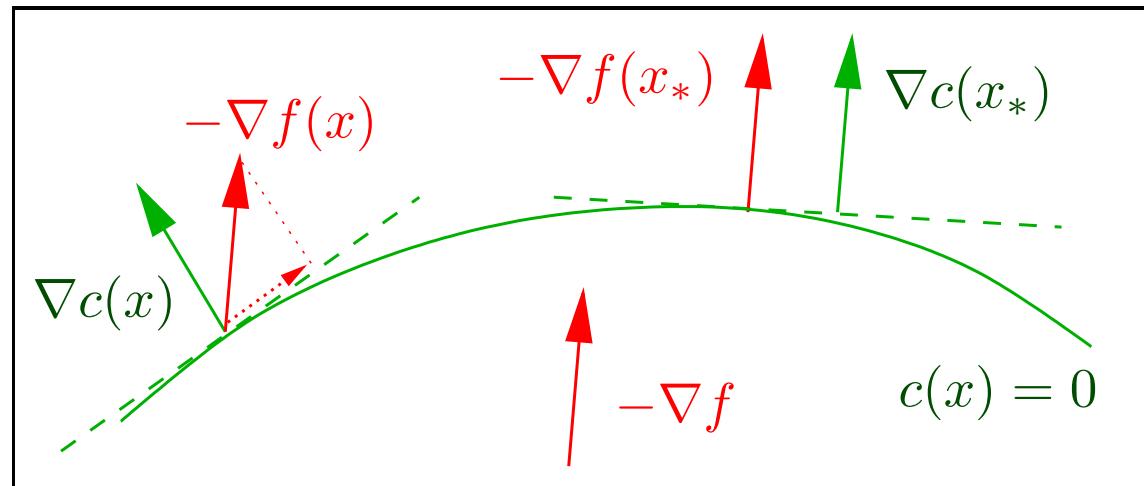
Optimality Conditions (Crash Course)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$



Optimality Conditions (Crash Course)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

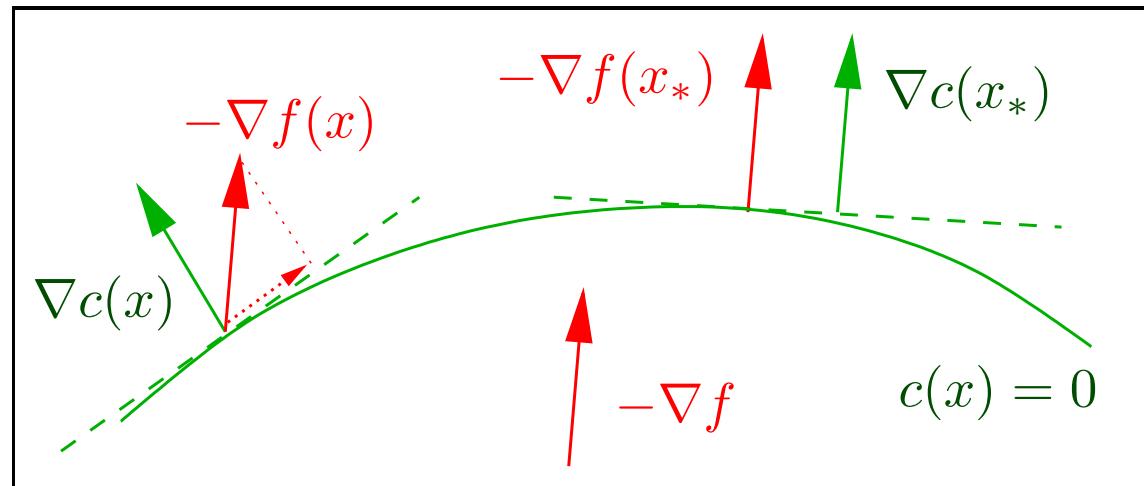


First order optimality conditions:

$$\begin{aligned} \nabla f(x_*) + \nabla c(x_*) \lambda_* &= 0 \\ c(x_*) &= 0 \end{aligned}$$

Optimality Conditions (Crash Course)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$



First order optimality conditions:

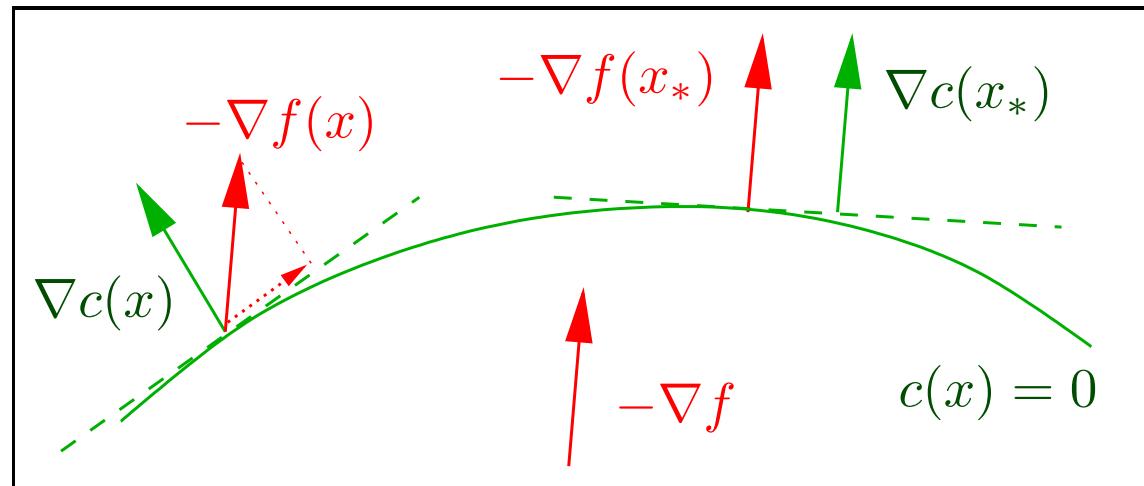
$$\begin{aligned} \nabla f(x_*) + \nabla c(x_*) \lambda_* &= 0 \\ c(x_*) &= 0 \end{aligned}$$

Constraint qualification:

$$\nabla c_1(x_*), \dots, \nabla c_m(x_*) \text{ linear ind.}$$

Optimality Conditions (Crash Course)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$



First order optimality conditions:

$$\begin{aligned} \nabla f(x_*) + \nabla c(x_*) \lambda_* &= 0 \\ c(x_*) &= 0 \end{aligned}$$

Constraint qualification:

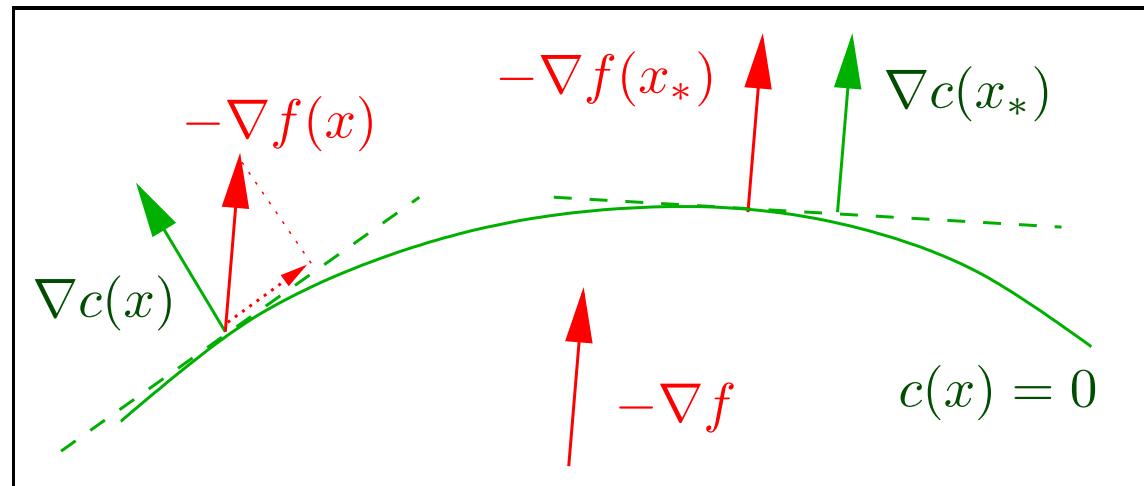
$$\nabla c_1(x_*), \dots, \nabla c_m(x_*) \text{ linear ind.}$$

Lagrangian function:

$$\mathcal{L}(x, \lambda) = f(x) + c(x)^T \lambda$$

Optimality Conditions (Crash Course)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$



First order optimality conditions:

$$\begin{aligned} \nabla_x \mathcal{L}(x_*, \lambda_*) &= 0 \\ c(x_*) &= 0 \end{aligned}$$

Constraint qualification:

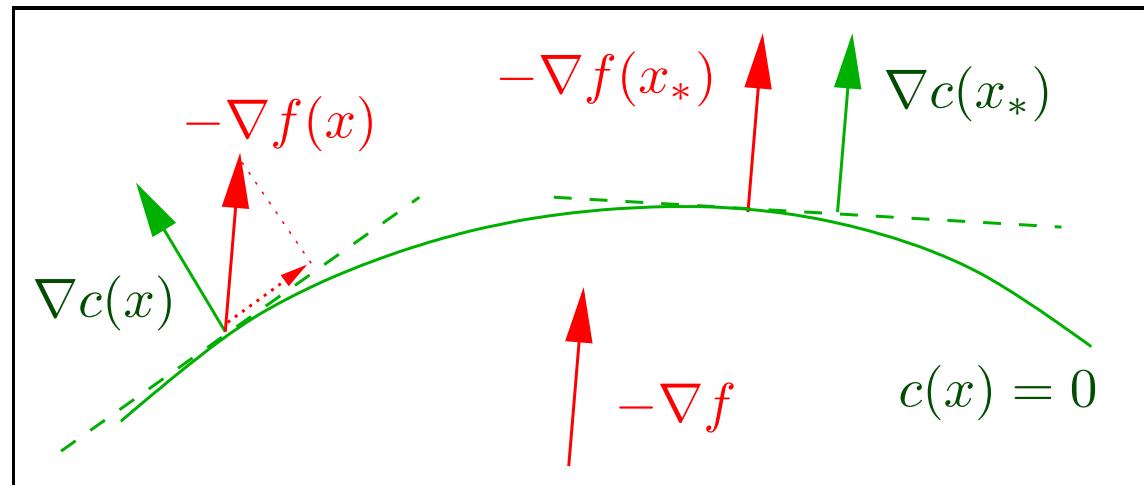
$$\nabla c_1(x_*), \dots, \nabla c_m(x_*) \text{ linear ind.}$$

Lagrangian function:

$$\mathcal{L}(x, \lambda) = f(x) + c(x)^T \lambda$$

Optimality Conditions (Crash Course)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$



First order optimality conditions:

$$\begin{aligned} \nabla_x \mathcal{L}(x_*, \lambda_*) &= 0 \\ c(x_*) &= 0 \end{aligned}$$

Constraint qualification:

$$\nabla c_1(x_*), \dots, \nabla c_m(x_*) \text{ linear ind.}$$

Lagrangian function:

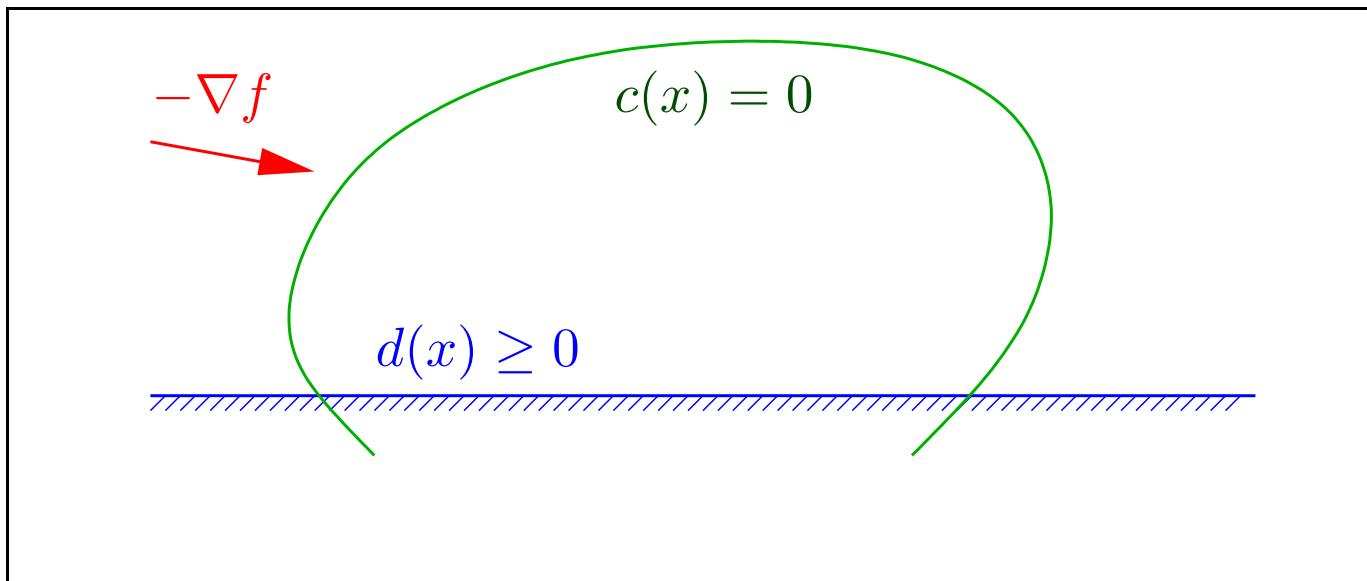
$$\mathcal{L}(x, \lambda) = f(x) + c(x)^T \lambda$$

Sufficient second order optimality conditions

$$\forall v \neq 0 : \quad \nabla c(x_*)^T v = 0 \quad \implies \quad v^T \nabla_{xx}^2 \mathcal{L}(x_*, \lambda_*) v > 0$$

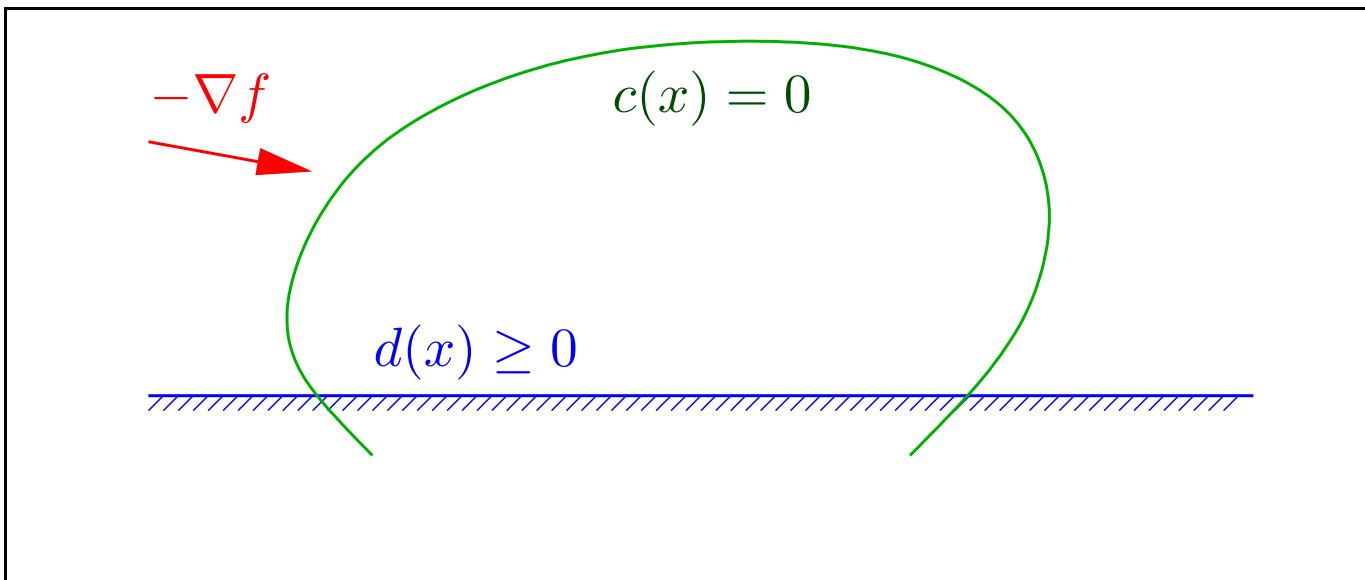
Inequality Constrained Problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & d(x) \geq 0 \end{aligned}$$



Inequality Constrained Problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & d(x) \geq 0 \end{aligned}$$

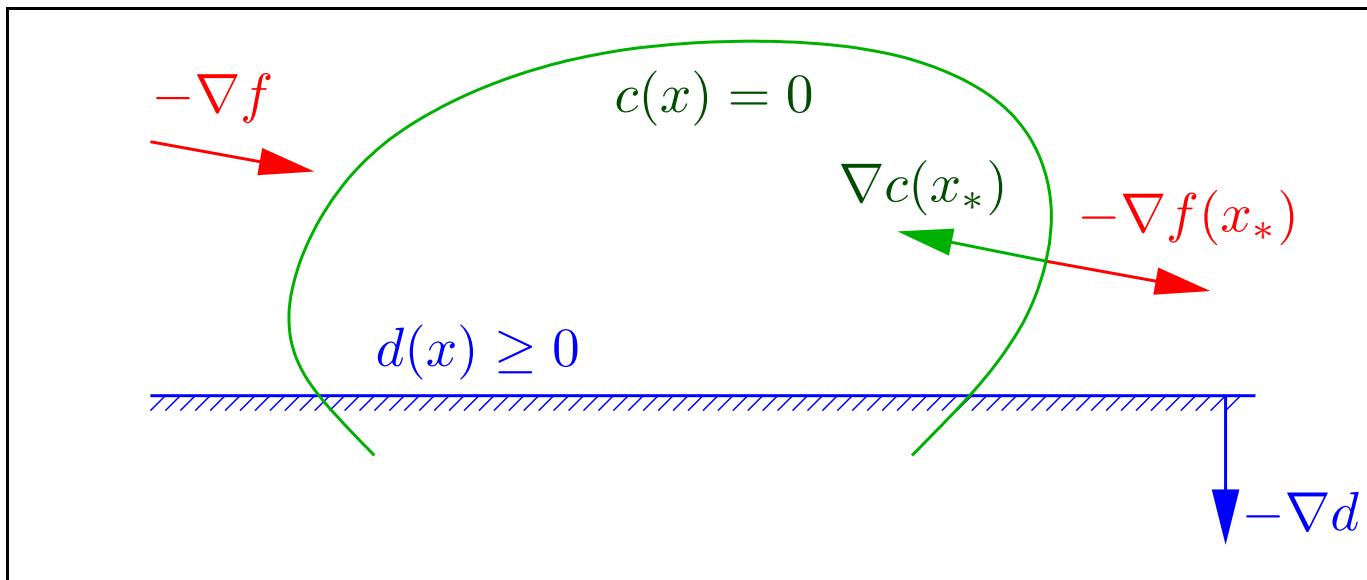


KKT conditions

$$\begin{aligned} \nabla f(x_*) + \nabla c(x_*)\lambda_* - \nabla d(x_*)z_* &= 0 \\ c(x_*) &= 0 \\ d(x_*) &\geq 0 \\ z_* &\geq 0 \\ \text{for all } i \quad d^{(i)}(x_*)z_*^{(i)} &= 0 \end{aligned}$$

Inequality Constrained Problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & d(x) \geq 0 \end{aligned}$$



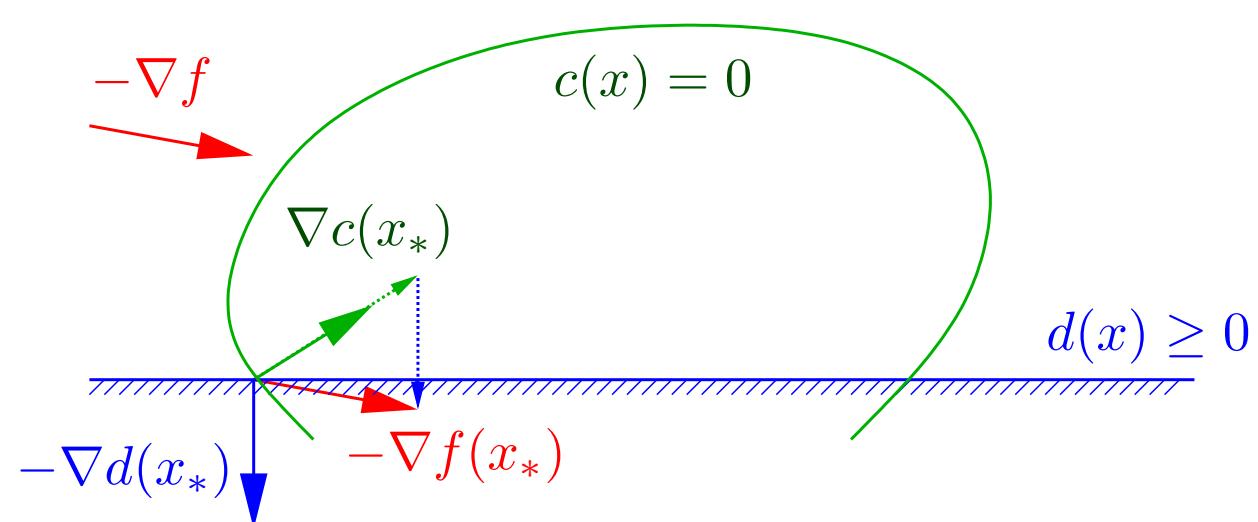
KKT conditions

$$\begin{aligned} \nabla f(x_*) + \nabla c(x_*) \lambda_* - \nabla d(x_*) z_* &= 0 \\ c(x_*) &= 0 \\ d(x_*) &\geq 0 \\ z_* &\geq 0 \\ \text{for all } i \quad d^{(i)}(x_*) z_*^{(i)} &= 0 \end{aligned}$$

$$\begin{aligned} -\nabla f(x_*) &= \nabla c(x_*) \lambda_* - \nabla d(x_*) z_* \\ c(x_*) &= 0 \\ d(x_*) &> 0 \\ z_* &= 0 \end{aligned}$$

Inequality Constrained Problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & d(x) \geq 0 \end{aligned}$$



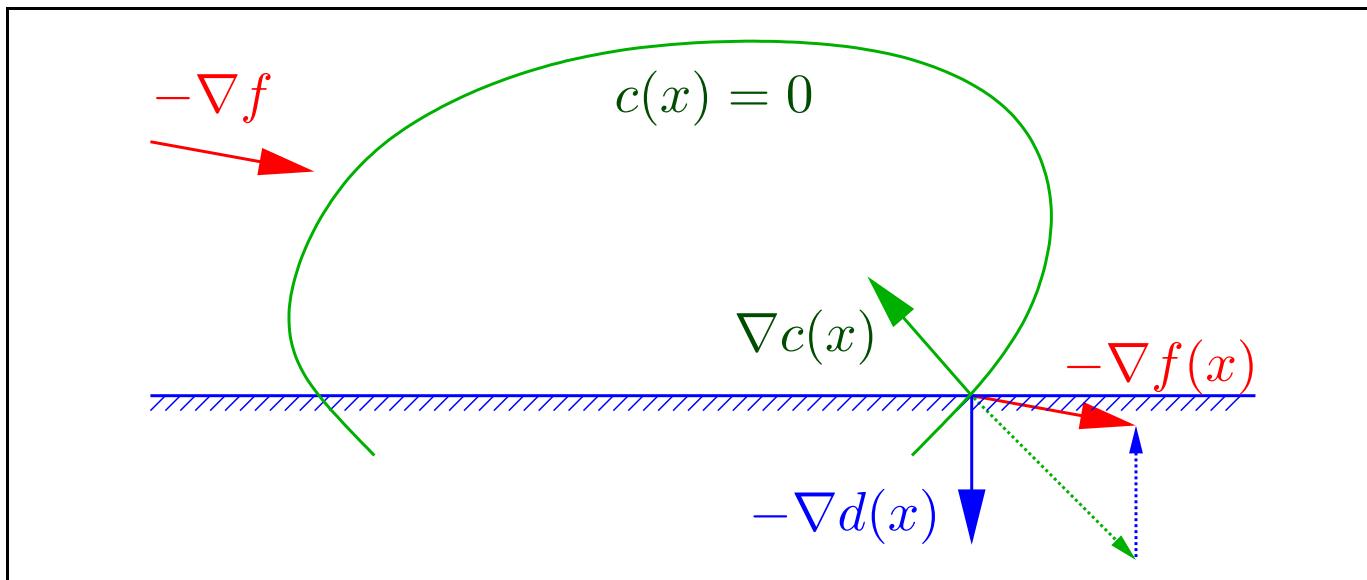
KKT conditions

$$\begin{aligned} \nabla f(x_*) + \nabla c(x_*) \lambda_* - \nabla d(x_*) z_* &= 0 \\ c(x_*) &= 0 \\ d(x_*) &\geq 0 \\ z_* &\geq 0 \\ \text{for all } i \quad d^{(i)}(x_*) z_*^{(i)} &= 0 \end{aligned}$$

$$\begin{aligned} -\nabla f(x_*) &= \nabla c(x_*) \lambda_* - \nabla d(x_*) z_* \\ c(x_*) &= 0 \\ d(x_*) &= 0 \\ z_* &> 0 \end{aligned}$$

Inequality Constrained Problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & d(x) \geq 0 \end{aligned}$$



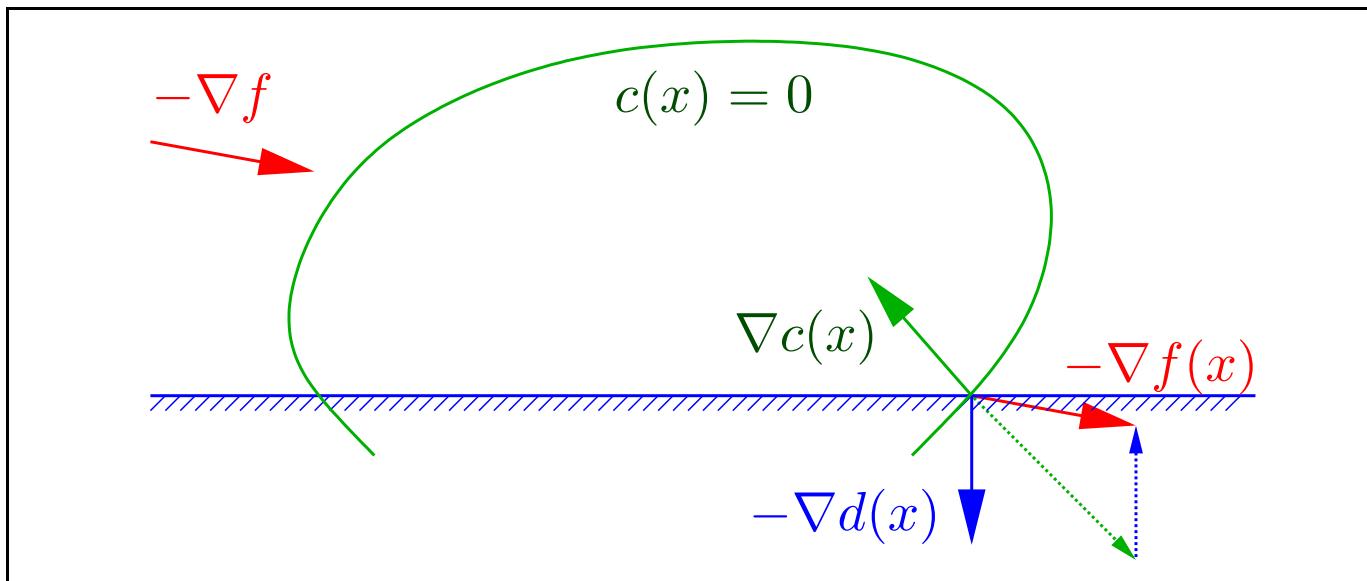
KKT conditions

$$\begin{aligned} \nabla f(x_*) + \nabla c(x_*)\lambda_* - \nabla d(x_*)z_* &= 0 \\ c(x_*) &= 0 \\ d(x_*) &\geq 0 \\ z_* &\geq 0 \\ \text{for all } i \quad d^{(i)}(x_*)z_*^{(i)} &= 0 \end{aligned}$$

$$\begin{aligned} -\nabla f(x) &= \nabla c(x)\lambda - \nabla d(x)z \\ c(x) &= 0 \\ d(x) &= 0 \\ z &< 0 \quad \text{Wrong sign!} \end{aligned}$$

Inequality Constrained Problems

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \\ & x \geq 0 \end{aligned}$$



KKT conditions

$$\begin{aligned} \nabla f(x_*) + \nabla c(x_*)\lambda_* - \nabla d(x_*)z_* &= 0 \\ c(x_*) &= 0 \\ d(x_*) &\geq 0 \\ z_* &\geq 0 \\ \text{for all } i \quad d^{(i)}(x_*)z_*^{(i)} &= 0 \end{aligned}$$

Special case

$$\begin{aligned} \nabla f(x) + \nabla c(x)\lambda - z &= 0 \\ c(x) &= 0 \\ XZe &= 0 \\ x, z &\geq 0 \\ X = \text{diag}(x), e = (1, \dots, 1)^T \end{aligned}$$

Active Set Methods

Optimality Conditions

$$\begin{aligned}\nabla f(\mathbf{x}) + \nabla c(\mathbf{x})\boldsymbol{\lambda} - \mathbf{z} &= \mathbf{0} \\ c(\mathbf{x}) &= 0 \\ \mathbf{XZ}\mathbf{e} &= \mathbf{0} \\ \mathbf{x}, \mathbf{z} &\geq \mathbf{0}\end{aligned}$$

Active Set Method

- Try to guess which ineq. are active
- Solve equality constrained problem (much easier)
- Update guess of activities and repeat

Active Set Methods

Optimality Conditions

$$\begin{aligned}\nabla f(\mathbf{x}) + \nabla c(\mathbf{x})\boldsymbol{\lambda} - \mathbf{z} &= \mathbf{0} \\ c(\mathbf{x}) &= \mathbf{0} \\ \mathbf{XZ}\mathbf{e} &= \mathbf{0} \\ \mathbf{x}, \mathbf{z} &\geq \mathbf{0}\end{aligned}$$

Active Set Method

- Try to guess which ineq. are active
- Solve equality constrained problem (much easier)
- Update guess of activities and repeat

- For LPs: Simplex method
- SQP codes for NLP:
 - SNOPT (Gill, Murray, Saunders)
 - filterSQP (Fletcher, Leyffer)
 - FSQP (Lawrence, Zhou, Tits) . . .
- Other active set NLP codes:
 - Lancelot (Conn, Gould, Toint)
 - Minos (Murtagh, Saunders, Gill, et al.) . . .

Active Set Methods

Optimality Conditions

$$\begin{aligned}\nabla f(\mathbf{x}) + \nabla c(\mathbf{x})\boldsymbol{\lambda} - \mathbf{z} &= \mathbf{0} \\ c(\mathbf{x}) &= \mathbf{0} \\ \mathbf{XZ}\mathbf{e} &= \mathbf{0} \\ \mathbf{x}, \mathbf{z} &\geq \mathbf{0}\end{aligned}$$

Active Set Method

- Try to guess which ineq. are active
- Solve equality constrained problem (much easier)
- Update guess of activities and repeat

- For LPs: Simplex method
- SQP codes for NLP:
 - SNOPT (Gill, Murray, Saunders)
 - filterSQP (Fletcher, Leyffer)
 - FSQP (Lawrence, Zhou, Tits) . . .
- Other active set NLP codes:
 - Lancelot (Conn, Gould, Toint)
 - Minos (Murtagh, Saunders, Gill, et al.) . . .
- Combinatorial complexity (particular for many degrees of freedom)
- + Efficient when solving sequence of similar problems

Barrier Methods

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$s.t. \quad c(x) = 0$$

$$x \geq 0$$

Barrier Methods

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$s.t. \quad c(x) = 0$$

$$x \geq 0$$



$$\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_{i=1}^n \ln(x^{(i)})$$

$$s.t. \quad c(x) = 0$$

Barrier Parameter: $\mu > 0$

Idea: $x_*(\mu) \rightarrow x_*$ as $\mu \rightarrow 0$.

- Fiacco, McCormick (1968)

Barrier Methods

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$s.t. \quad c(x) = 0 \\ x \geq 0$$



$$\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_{i=1}^n \ln(x^{(i)})$$

$$s.t. \quad c(x) = 0$$

Outer Algorithm

1. Given initial $x_0 > 0$, $\mu_0 > 0$. Set $l \leftarrow 0$.
2. Compute (approximate) solution x_{l+1} for BP(μ_l) with error tolerance $\epsilon(\mu_l)$.
3. Decrease barrier parameter μ_l (superlinearly) to get μ_{l+1} .
4. Increase $l \leftarrow l + 1$; go to 2.

Barrier Parameter: $\mu > 0$

Idea: $x_*(\mu) \rightarrow x_*$ as $\mu \rightarrow 0$.

- Fiacco, McCormick (1968)

Barrier Methods

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$s.t. \quad c(x) = 0 \\ x \geq 0$$



$$\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_{i=1}^n \ln(x^{(i)})$$

$$s.t. \quad c(x) = 0$$

Barrier Parameter: $\mu > 0$

Idea: $x_*(\mu) \rightarrow x_*$ as $\mu \rightarrow 0$.

- Fiacco, McCormick (1968)

Outer Algorithm

1. Given initial $x_0 > 0$, $\mu_0 > 0$. Set $l \leftarrow 0$.
2. Compute (approximate) solution x_{l+1} for BP(μ_l) with error tolerance $\epsilon(\mu_l)$.
3. Decrease barrier parameter μ_l (superlinearly) to get μ_{l+1} .
4. Increase $l \leftarrow l + 1$; go to 2.

Implementations

- KNITRO (Byrd, Nocedal, Hribar, Waltz)
- LOQO (Benson, Vanderbei, Shanno)
- IPOPT (W, Biegler)
- ...

Solution of the Barrier Problem

Barrier Problem (fixed μ)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \varphi_\mu(x) := f(x) - \mu \sum \ln(x^{(i)}) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

Solution of the Barrier Problem

Barrier Problem (fixed μ)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \varphi_\mu(x) := f(x) - \mu \sum \ln(x^{(i)}) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

Optimality Conditions

$$\begin{aligned} \nabla \varphi_\mu(x) + \nabla c(x)\lambda &= 0 \\ c(x) &= 0 \\ (x > 0) \end{aligned}$$

Solution of the Barrier Problem

Barrier Problem (fixed μ)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \varphi_\mu(x) := f(x) - \mu \sum \ln(x^{(i)}) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

Optimality Conditions

$$\begin{aligned} \nabla \varphi_\mu(x) + \nabla c(x)\lambda &= 0 \\ c(x) &= 0 \\ (x > 0) \end{aligned}$$

Apply Newton's Method

$$\begin{bmatrix} W_k^\mu & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_\mu(x_k) + \nabla c(x_k)\lambda_k \\ c(x_k) \end{pmatrix}$$

Here:

- $W_k^\mu = \nabla_{xx}^2 \mathcal{L}_\mu(x_k, \lambda_k)$
- $\mathcal{L}_\mu(x, \lambda) = \varphi_\mu(x) + c(x)^T \lambda$

Solution of the Barrier Problem

Barrier Problem (fixed μ)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \varphi_\mu(x) := f(x) - \mu \sum \ln(x^{(i)}) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

Optimality Conditions

$$\begin{aligned} \nabla \varphi_\mu(x) + \nabla c(x)\lambda &= 0 \\ c(x) &= 0 \\ (x > 0) \end{aligned}$$

Apply Newton's Method

$$\begin{bmatrix} W_k^\mu & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_\mu(x_k) + \nabla c(x_k)\lambda_k \\ c(x_k) \end{pmatrix}$$

Here:

- $W_k^\mu = \nabla_{xx}^2 \mathcal{L}_\mu(x_k, \lambda_k)$
- $\mathcal{L}_\mu(x, \lambda) = \varphi_\mu(x) + c(x)^T \lambda$

$$\begin{aligned} \nabla \varphi_\mu(x) &= \nabla f(x) - \mu X^{-1} e \\ \nabla^2 \varphi_\mu(x) &= \nabla^2 f(x) + \mu X^{-2} \\ X &:= \text{diag}(x) \quad e := (1, \dots, 1)^T \end{aligned}$$

Solution of the Barrier Problem

Barrier Problem (fixed μ)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \varphi_\mu(x) := f(x) - \mu \sum \ln(x^{(i)}) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

Optimality Conditions

$$\begin{aligned} \nabla \varphi_\mu(x) + \nabla c(x)\lambda &= 0 \\ c(x) &= 0 \\ (x > 0) \end{aligned}$$

Apply Newton's Method

$$\begin{bmatrix} W_k + \mu X_k^{-2} & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_\mu(x_k) + \nabla c(x_k)\lambda_k \\ c(x_k) \end{pmatrix}$$

Here:

- $W_k = \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k)$
- $\mathcal{L}(x, \lambda) = f(x) + c(x)^T \lambda$

$$\begin{aligned} \nabla \varphi_\mu(x) &= \nabla f(x) - \mu X^{-1} e \\ \nabla^2 \varphi_\mu(x) &= \nabla^2 f(x) + \mu X^{-2} \\ X &:= \text{diag}(x) \quad e := (1, \dots, 1)^T \end{aligned}$$

Primal-Dual Approach

Primal

$$\nabla f(\textcolor{red}{x}) - \mu X^{-1}e + \nabla c(x)\lambda = 0$$

$$c(\textcolor{red}{x}) = 0$$

$$(\textcolor{red}{x} > 0)$$

Primal-Dual Approach

Primal

$$\begin{aligned}\nabla f(\mathbf{x}) - \mu X^{-1}e + \nabla c(\mathbf{x})\boldsymbol{\lambda} &= 0 \\ c(\mathbf{x}) &= 0 \\ (\mathbf{x} > 0) &\end{aligned}$$

$$z = \mu X^{-1}e$$

Primal-Dual

$$\begin{aligned}\nabla f(\mathbf{x}) + \nabla c(\mathbf{x})\boldsymbol{\lambda} - z &= 0 \\ c(\mathbf{x}) &= 0 \\ XZ e - \mu e &= 0 \\ (\mathbf{x}, z > 0) &\end{aligned}$$

Primal-Dual Approach

Primal

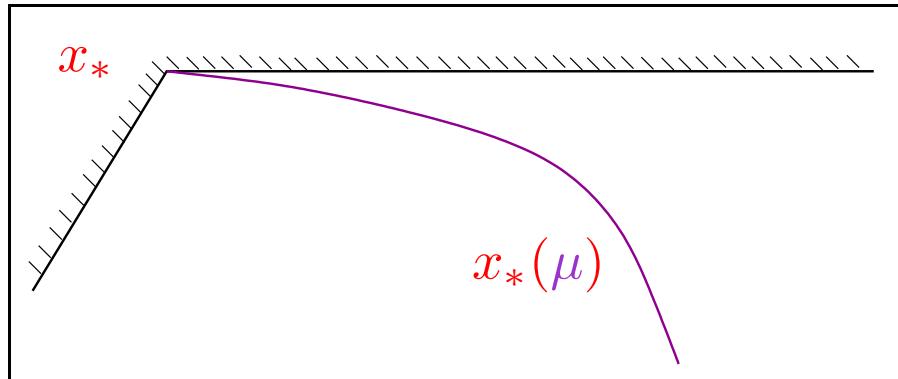
$$\begin{aligned}\nabla f(\mathbf{x}) - \mu X^{-1}e + \nabla c(\mathbf{x})\lambda &= 0 \\ c(\mathbf{x}) &= 0 \\ (\mathbf{x} > 0) &\end{aligned}$$

$$z = \mu X^{-1}e$$

Primal-Dual

$$\begin{aligned}\nabla f(\mathbf{x}) + \nabla c(\mathbf{x})\lambda - z &= 0 \\ c(\mathbf{x}) &= 0 \\ XZe - \mu e &= 0 \\ (\mathbf{x}, z > 0) &\end{aligned}$$

Central path (parameterized in μ)



“Homotopy approach”

Original KKT Conditions

$$\begin{aligned}\nabla f(\mathbf{x}) + \nabla c(\mathbf{x})\lambda - z &= 0 \\ c(\mathbf{x}) &= 0 \\ XZe &= 0 \\ (\mathbf{x}, z \geq 0) &\end{aligned}$$

Primal-Dual Approach

Primal

$$\begin{aligned}\nabla f(\mathbf{x}) - \mu X^{-1}e + \nabla c(\mathbf{x})\lambda &= 0 \\ c(\mathbf{x}) &= 0 \\ (\mathbf{x} > 0) &\end{aligned}$$

$$z = \mu X^{-1}e$$

Primal-Dual

$$\begin{aligned}\nabla f(\mathbf{x}) + \nabla c(\mathbf{x})\lambda - z &= 0 \\ c(\mathbf{x}) &= 0 \\ XZ e - \mu e &= 0 \\ (\mathbf{x}, z > 0) &\end{aligned}$$

Apply Newton's Method

$$\begin{bmatrix} W_k & \nabla c(\mathbf{x}_k) & -I \\ \nabla c(\mathbf{x}_k)^T & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{pmatrix} \Delta \mathbf{x}_k \\ \Delta \lambda_k \\ \Delta z_k \end{pmatrix} = - \begin{pmatrix} \nabla f(\mathbf{x}_k) + \nabla c(\mathbf{x}_k)\lambda_k - z_k \\ c(\mathbf{x}_k) \\ X_k Z_k e - \mu e \end{pmatrix}$$

Again: $W_k = \nabla_{xx}^2 \mathcal{L}(\mathbf{x}_k, \lambda_k)$

Primal-Dual Approach

Primal

$$\begin{aligned}\nabla f(\mathbf{x}) - \mu X^{-1}e + \nabla c(\mathbf{x})\lambda &= 0 \\ c(\mathbf{x}) &= 0 \\ (\mathbf{x} > 0) &\end{aligned}$$

$$z = \mu X^{-1}e$$

Primal-Dual

$$\begin{aligned}\nabla f(\mathbf{x}) + \nabla c(\mathbf{x})\lambda - z &= 0 \\ c(\mathbf{x}) &= 0 \\ XZ e - \mu e &= 0 \\ (\mathbf{x}, z > 0) &\end{aligned}$$

Apply Newton's Method

$$\begin{bmatrix} W_k + \Sigma_k & \nabla c(\mathbf{x}_k) \\ \nabla c(\mathbf{x}_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta \mathbf{x}_k \\ \Delta \lambda_k \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_\mu(\mathbf{x}_k) + \nabla c(\mathbf{x}_k)\lambda_k \\ c(\mathbf{x}_k) \end{pmatrix}$$

$$\Delta z_k = \mu X_k^{-1}e - z_k - \Sigma_k \Delta x_k \quad \Sigma_k = X_k^{-1}Z_k$$

Again: $W_k = \nabla_{xx}^2 \mathcal{L}(\mathbf{x}_k, \lambda_k)$

A Basic NLP Interior Point Algorithm

0. Choose $\mu_0, \epsilon_0 > 0$, (x_0, λ_0, z_0) with $x_0, z_0 > 0$. Set $k = 0$
1. If barrier problem solved to tolerance ϵ_k : Reduce μ_k, ϵ_k
2. Compute primal-dual search direction $(\Delta x_k, \Delta \lambda_k, \Delta z_k)$
3. Determine largest $\alpha_k^\tau, \alpha_k^{z,\tau} \in (0, 1]$ such that $(\tau \approx 0.99)$

$$x_k + \alpha_k^\tau \Delta x_k \geq (1 - \tau) x_k > 0$$

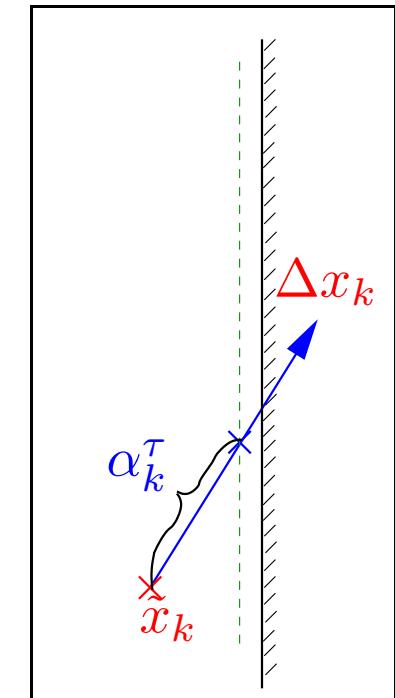
$$z_k + \alpha_k^{z,\tau} \Delta z_k \geq (1 - \tau) z_k > 0$$

4. Choose step sizes $\alpha_k \in (0, \alpha_k^\tau]$ and $\alpha_k^z \in (0, \alpha_k^{z,\tau}]$

$$(x_{k+1}, \lambda_{k+1}) = (x_k, \lambda_k) + \alpha_k (\Delta x_k, \Delta \lambda_k)$$

$$z_{k+1} = z_k + \alpha_k^z \Delta z_k$$

5. Increase $k \leftarrow k + 1$ and continue at 1.



Line Search

- Basic framework locally convergent (e.g. Byrd, Liu, Nocedal (1998))
- Need globalization approach to ensure convergence from arbitrary starting point

Line Search

- Basic framework locally convergent (e.g. Byrd, Liu, Nocedal (1998))
- Need globalization approach to ensure convergence from arbitrary starting point

In basic framework:

4. Choose step sizes $\alpha_k \in (0, \alpha_k^\tau]$ and $\alpha_k^z \in (0, \alpha_k^{z,\tau}]$

$$\begin{aligned}(x_{k+1}, \lambda_{k+1}) &= (x_k, \lambda_k) + \alpha_k (\Delta x_k, \Delta \lambda_k) \\ z_{k+1} &= z_k + \alpha_k^z \Delta z_k\end{aligned}$$

Line Search

- Basic framework locally convergent (e.g. Byrd, Liu, Nocedal (1998))
- Need globalization approach to ensure convergence from arbitrary starting point

In basic framework:

4. Choose step sizes $\alpha_k \in (0, \alpha_k^\tau]$ and $\alpha_k^z \in (0, \alpha_k^{z,\tau}]$

$$\begin{aligned}(x_{k+1}, \lambda_{k+1}) &= (x_k, \lambda_k) + \alpha_k (\Delta x_k, \Delta \lambda_k) \\ z_{k+1} &= z_k + \alpha_k^z \Delta z_k\end{aligned}$$

- Choose $\alpha_k^z = \alpha_k^{z,\tau}$
- Backtracking line search $\alpha_{k,l} = 2^{-l} \alpha_k^\tau$ to find “acceptable trial point”
 - e.g., descent in exact penalty function: $\phi_\nu(x) = \varphi_\mu(x) + \nu \|c(x)\|$

Example Problem

$$\begin{aligned} \min_{x \in \mathbb{R}^3} \quad & x^{(1)} \\ s.t. \quad & (x^{(1)})^2 - x^{(2)} - 1 = 0 \\ & x^{(1)} - x^{(3)} - 0.5 = 0 \\ & x^{(2)}, x^{(3)} \geq 0 \end{aligned}$$

W, Biegler (2000)

Example Problem

$$\begin{aligned} \min_{x \in \mathbb{R}^3} \quad & x^{(1)} \\ s.t. \quad & (x^{(1)})^2 - x^{(2)} - 1 = 0 \\ & x^{(1)} - x^{(3)} - 0.5 = 0 \\ & x^{(2)}, x^{(3)} \geq 0 \end{aligned}$$

W, Biegler (2000)

- $f(x)$ and $c(x)$ smooth
- $\nabla c(x)$ full rank everywhere
- Only one stationary point:
 - $x_* = (1, 0, 0.5)^T$
 - $\lambda_* = (-0.5, 0)^T$
 - $(z_*^{(2)}, z_*^{(3)})^T = (0.5, 0)^T$
- Sufficient 2nd order optimality cond.
- Strict complementarity

Example Problem

$$\begin{aligned} \min_{x \in \mathbb{R}^3} \quad & x^{(1)} \\ \text{s.t.} \quad & (x^{(1)})^2 - x^{(2)} - 1 = 0 \\ & x^{(1)} - x^{(3)} - 0.5 = 0 \\ & x^{(2)}, x^{(3)} \geq 0 \end{aligned}$$

W, Biegler (2000)

- $f(x)$ and $c(x)$ smooth
- $\nabla c(x)$ full rank everywhere
- Only one stationary point:
 $x_* = (1, 0, 0.5)^T$
 $\lambda_* = (-0.5, 0)^T$
 $(z_*^{(2)}, z_*^{(3)})^T = (0.5, 0)^T$
- Sufficient 2nd order optimality cond.
- Strict complementarity

This problem is well–posed!

Example Problem

$$\begin{aligned} \min_{x \in \mathbb{R}^3} \quad & x^{(1)} \\ \text{s.t.} \quad & (x^{(1)})^2 - x^{(2)} - 1 = 0 \\ & x^{(1)} - x^{(3)} - 0.5 = 0 \\ & x^{(2)}, x^{(3)} \geq 0 \end{aligned}$$

W, Biegler (2000)

- $f(x)$ and $c(x)$ smooth
- $\nabla c(x)$ full rank everywhere
- Only one stationary point:
 $x_* = (1, 0, 0.5)^T$
 $\lambda_* = (-0.5, 0)^T$
 $(z_*^{(2)}, z_*^{(3)})^T = (0.5, 0)^T$
- Sufficient 2nd order optimality cond.
- Strict complementarity

This problem is well–posed!

Let's try starting point $x_0 = (-2, 3, 1)^T \dots$

But...

ITER	X(1)	X(2)	X(3)
0	-0.20000000D+01	0.30000000D+01	0.10000000D+01
1	-0.18947413D+01	0.25789655D+01	0.10000000D-01
2	-0.12201587D+01	0.25789655D-01	0.23540510D-02
3	-0.12082251D+01	0.25789655D-03	0.93113038D-03
4	-0.12080095D+01	0.25789655D-05	0.15850601D-03
5	-0.12079893D+01	0.14108060D-05	0.15850601D-05
6	-0.12079884D+01	0.14108060D-07	0.10157878D-06
7	-0.12079884D+01	0.14108060D-09	0.22892307D-08
8	-0.12079884D+01	0.21513916D-11	0.22892307D-10
9	-0.12079884D+01	0.21513916D-13	0.25068098D-12
10	-0.12079884D+01	0.21513916D-15	0.25450869D-14

From starting point $x^0 = (-2, 3, 1)^T$, original IPOPT crashes into bounds:

Different starting points give similar results...

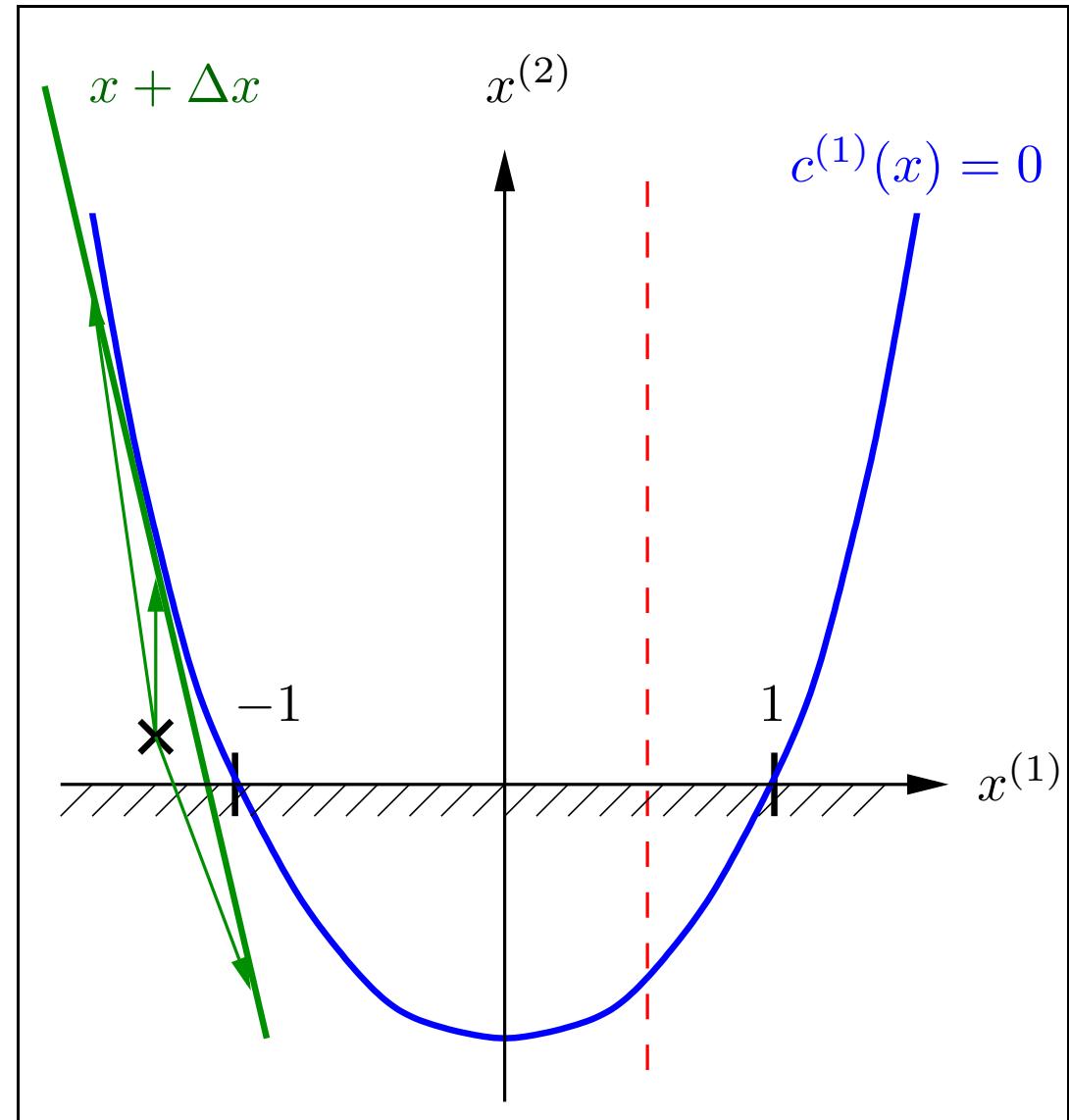
Failure of Global Convergence

$$\min_{x \in \mathbb{R}^3} f(x)$$

$$s.t. \quad (x^{(1)})^2 - x^{(2)} - 1 = 0$$

$$x^{(1)} - x^{(3)} - 0.5 = 0$$

$$x^{(2)}, x^{(3)} \geq 0$$



Failure of Global Convergence

$$\min_{x \in \mathbb{R}^3} f(x)$$

$$s.t. \quad (x^{(1)})^2 - x^{(2)} - 1 = 0$$

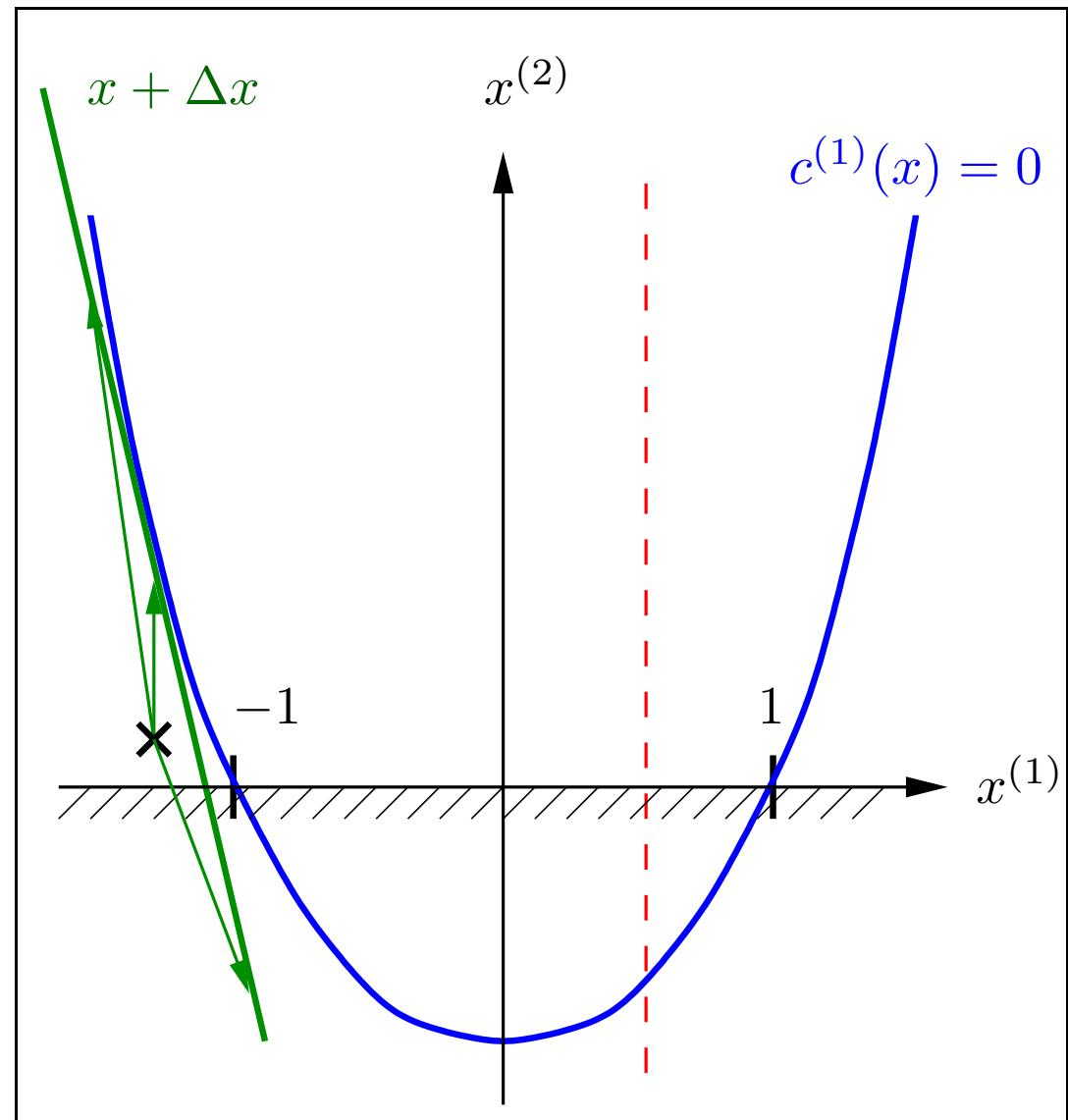
$$x^{(1)} - x^{(3)} - 0.5 = 0$$

$$x^{(2)}, x^{(3)} \geq 0$$

Step computation

$$1. \quad \nabla c(x_k)^T \Delta x_k + c(x_k) = 0$$

$$2. \quad x_k^{(i)} + \alpha_k \Delta x_k^{(i)} > 0$$



Line Search

- Basic framework locally convergent (e.g. Byrd, Liu, Nocedal (1998))
- Need globalization approach to ensure convergence from arbitrary starting point

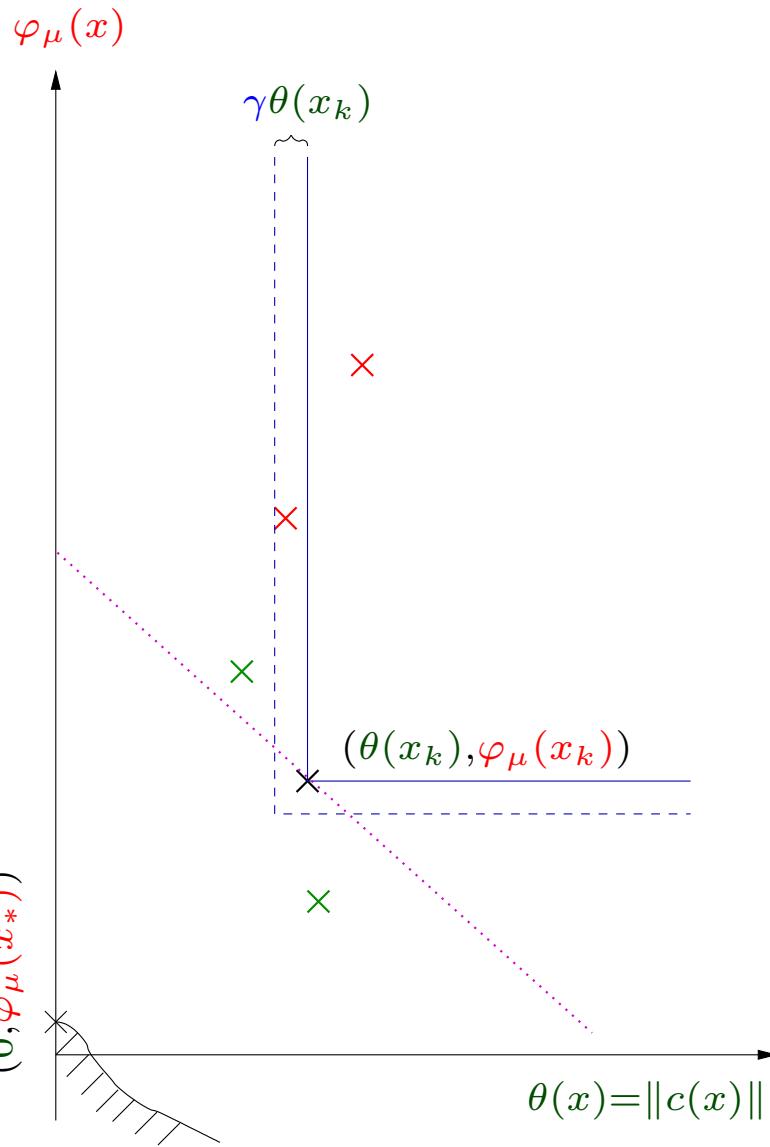
In basic framework:

4. Choose step sizes $\alpha_k \in (0, \alpha_k^\tau]$ and $\alpha_k^z \in (0, \alpha_k^{z,\tau}]$

$$\begin{aligned}(x_{k+1}, \lambda_{k+1}) &= (x_k, \lambda_k) + \alpha_k (\Delta x_k, \Delta \lambda_k) \\ z_{k+1} &= z_k + \alpha_k^z \Delta z_k\end{aligned}$$

- Choose $\alpha_k^z = \alpha_k^{z,\tau}$
- Backtracking line search $\alpha_{k,l} = 2^{-l} \alpha_k^\tau$ to find “acceptable trial point”
 - e.g., descent in exact penalty function: $\phi_\nu(x) = \varphi_\mu(x) + \nu \|c(x)\|$
 - **filter method**

A Filter Line Search Method



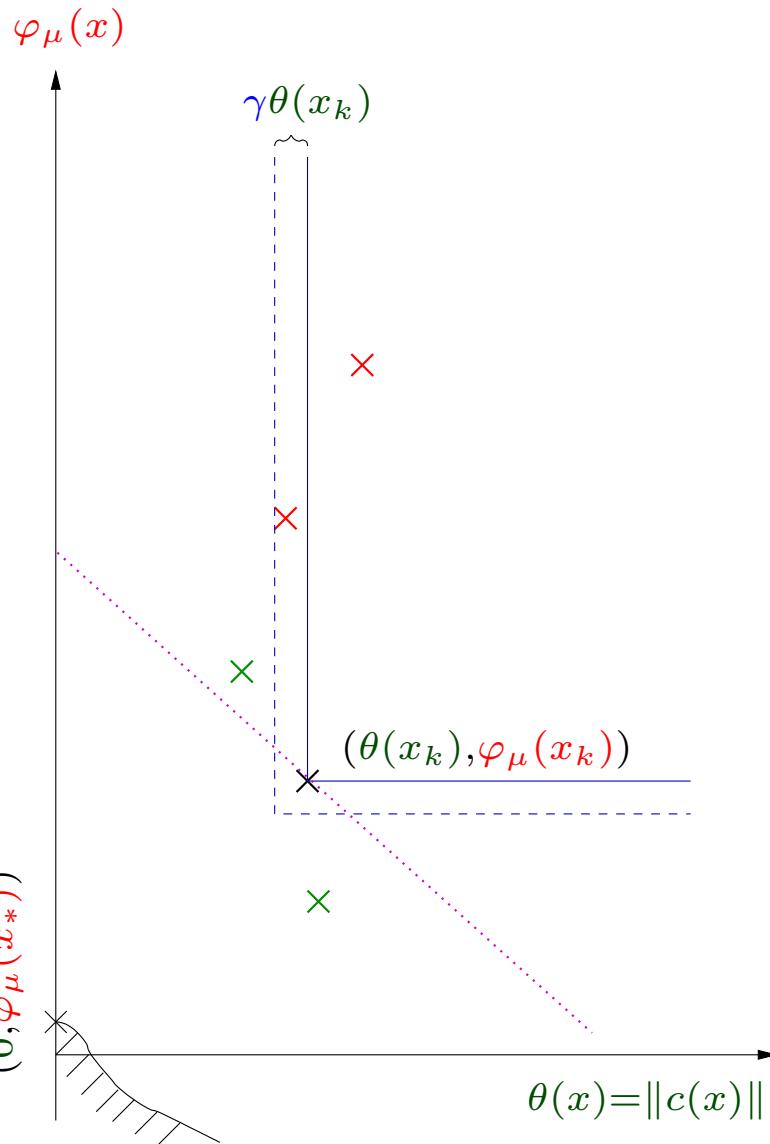
Idea: Bi-objective optimization
(Fletcher, Leyffer; 1998)

$$\begin{aligned} \min \quad & \varphi_\mu(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

$$\min \theta(x)$$

$$\min \varphi_\mu(x)$$

A Filter Line Search Method



Idea: Bi-objective optimization
(Fletcher, Leyffer; 1998)

$$\begin{aligned} \min \quad & \varphi_\mu(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

$$\min \theta(x)$$

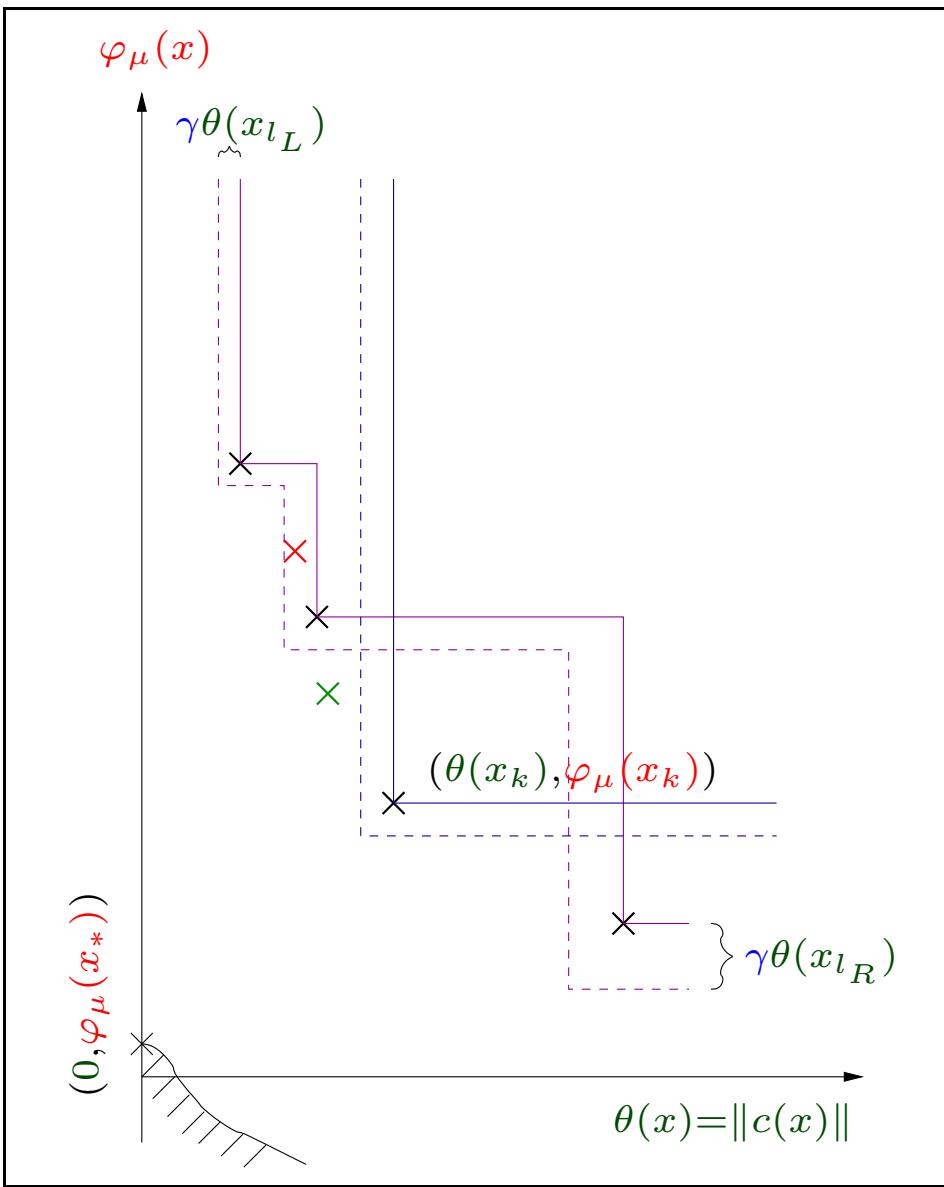
$$\min \varphi_\mu(x)$$

$$x_{\text{tr}} = x_k + \alpha \Delta x_k$$

Sufficient progress w.r.t. x_k :

$$\begin{aligned} \varphi_\mu(x_{\text{tr}}) &\leq \varphi_\mu(x_k) - \gamma_\varphi \theta(x_k) \\ \theta(x_{\text{tr}}) &\leq \theta(x_k) - \gamma_\theta \theta(x_k) \end{aligned}$$

A Filter Line Search Method (Filter)



Need to avoid cycling



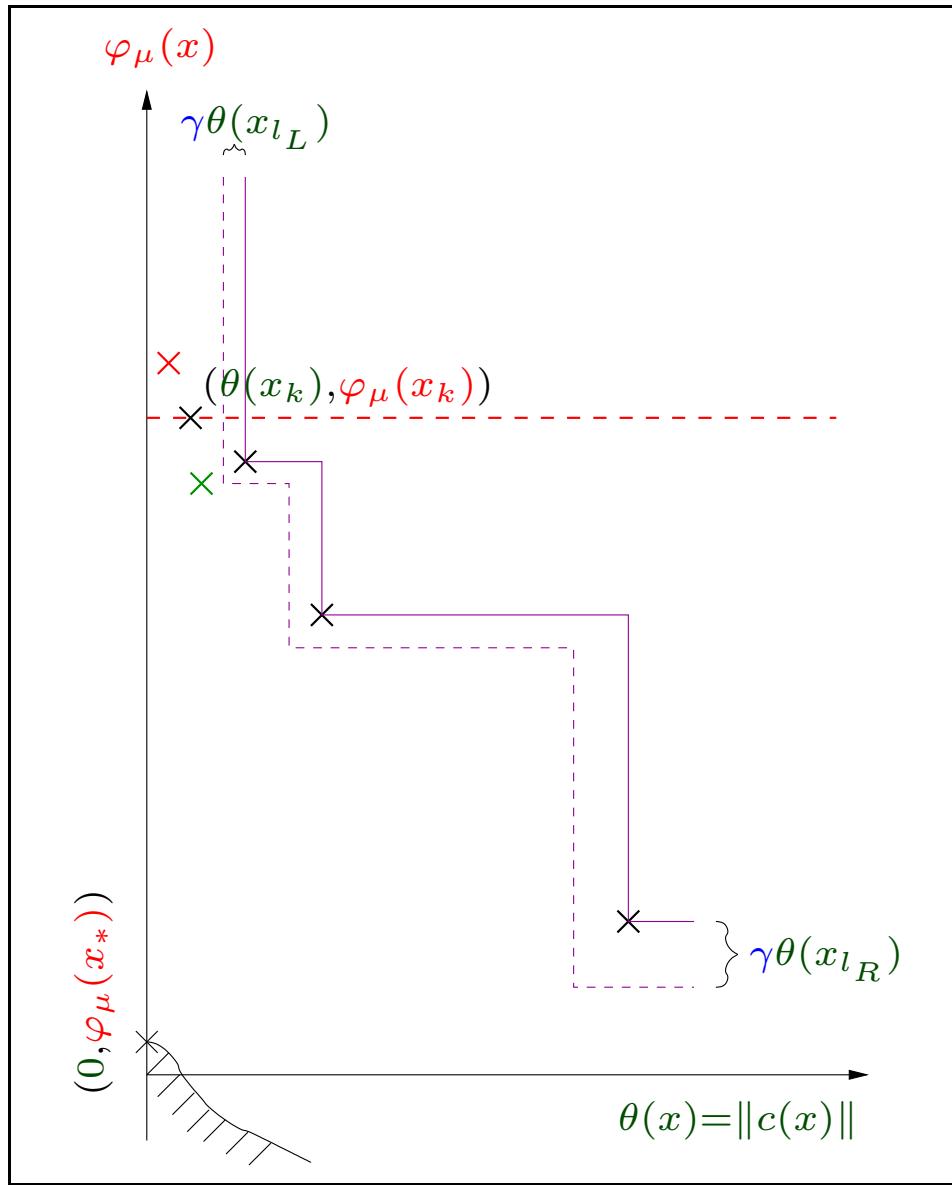
Store some previous
 $(\theta(x_l), \varphi_\mu(x_l))$ pairs in *filter* \mathcal{F}_k

Sufficient progress w.r.t. filter:

$$\begin{aligned}\varphi_\mu(x_{\text{tr}}) &\leq \varphi_\mu(x_l) - \gamma_\varphi \theta(x_l) \\ \theta(x_{\text{tr}}) &\leq \theta(x_l) - \gamma_\theta \theta(x_l)\end{aligned}$$

for $(\theta(x_l), \varphi_\mu(x_l)) \in \mathcal{F}_k$

A Filter Line Search Method (“ φ -type”)



If switching condition

$$-\alpha \nabla \varphi_\mu(x_k)^T \Delta x_k > \delta [\theta(x_k)]^{s_\theta}$$

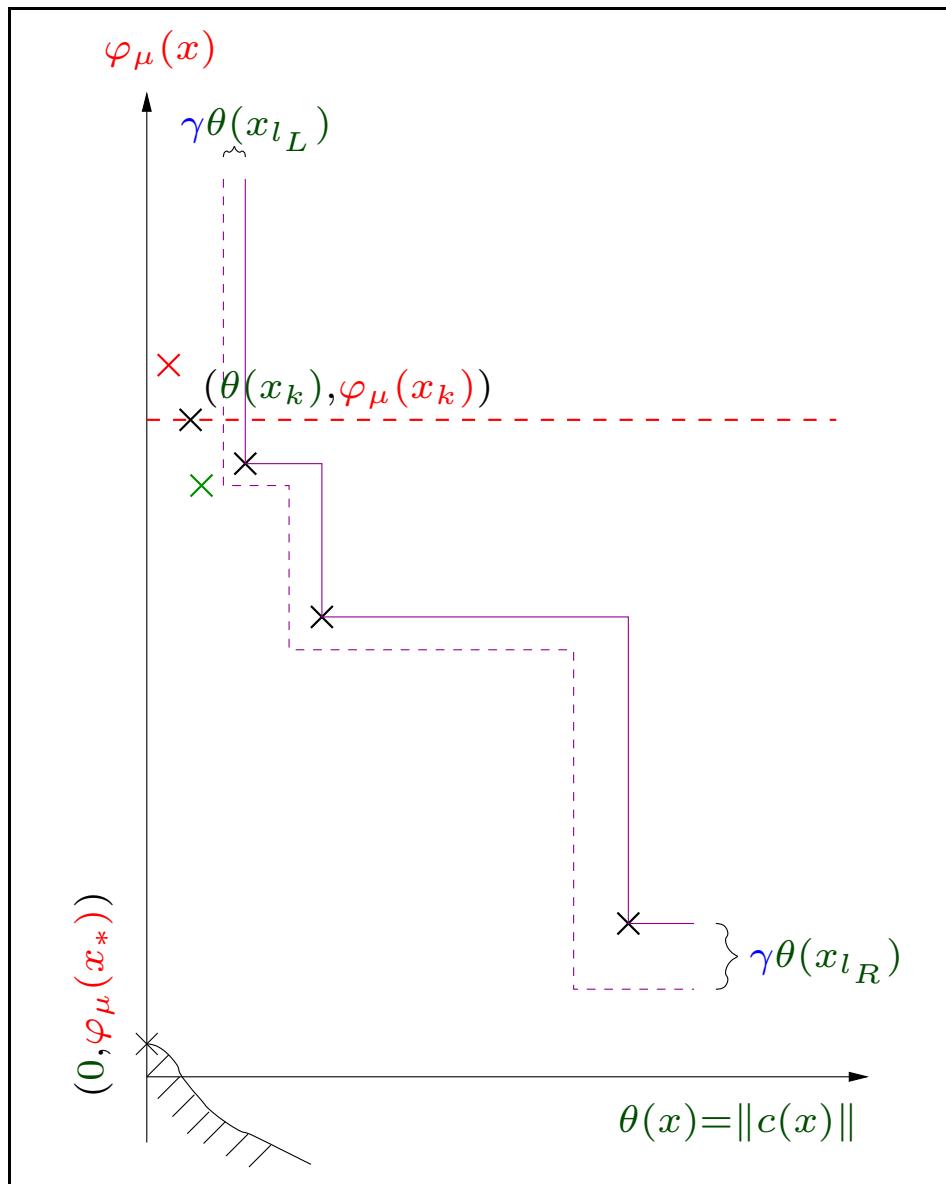
holds ($s_\theta > 1$):



Armijo-condition on $\varphi_\mu(x)$:

$$\varphi_\mu(x_{\text{tr}}) \leq \varphi_\mu(x_k) + \alpha \nabla \varphi_\mu(x_k)^T \Delta x_k$$

A Filter Line Search Method (“ φ -type”)



If switching condition

$$-\alpha \nabla \varphi_\mu(x_k)^T \Delta x_k > \delta [\theta(x_k)]^{s_\theta}$$

holds ($s_\theta > 1$):

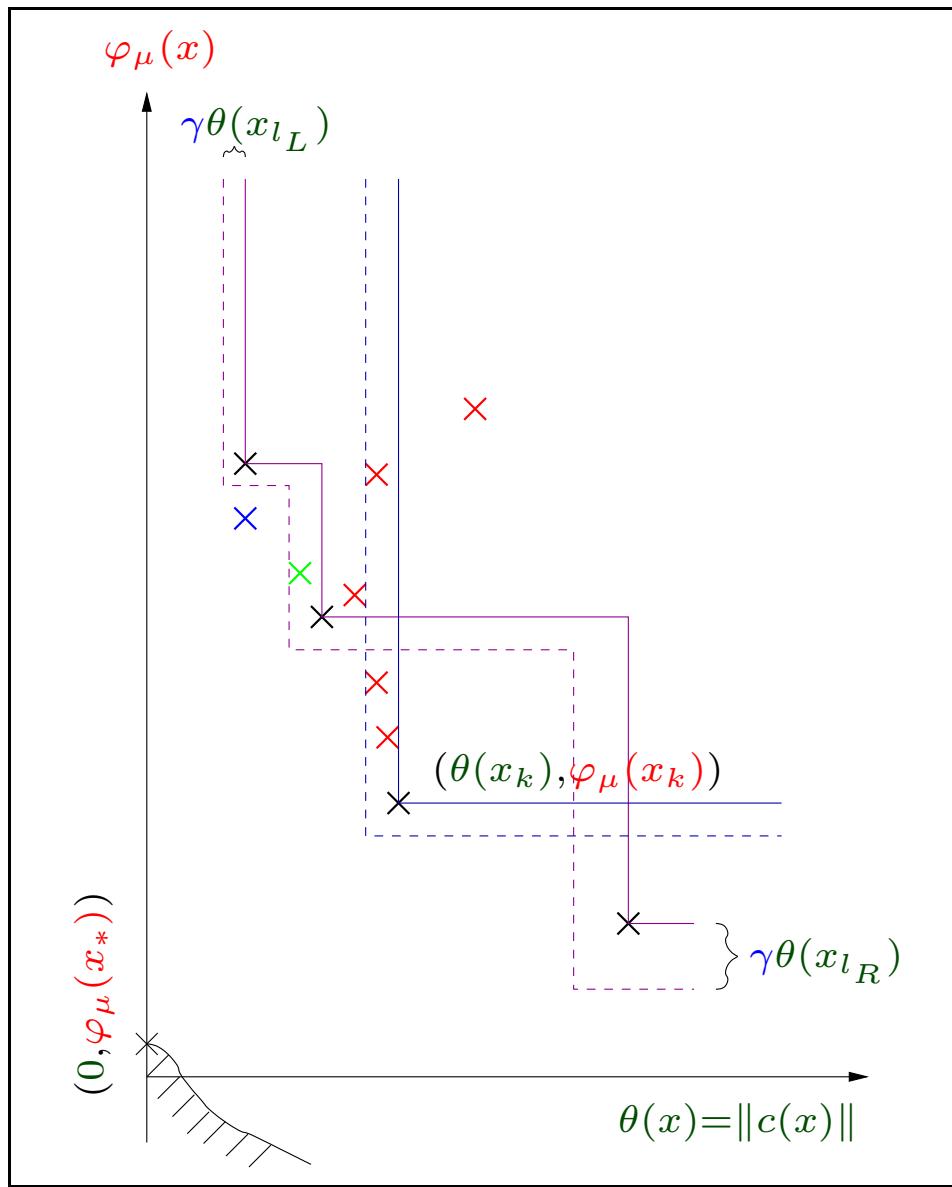


Armijo-condition on $\varphi_\mu(x)$:

$$\varphi_\mu(x_{\text{tr}}) \leq \varphi_\mu(x_k) + \alpha \nabla \varphi_\mu(x_k)^T \Delta x_k$$

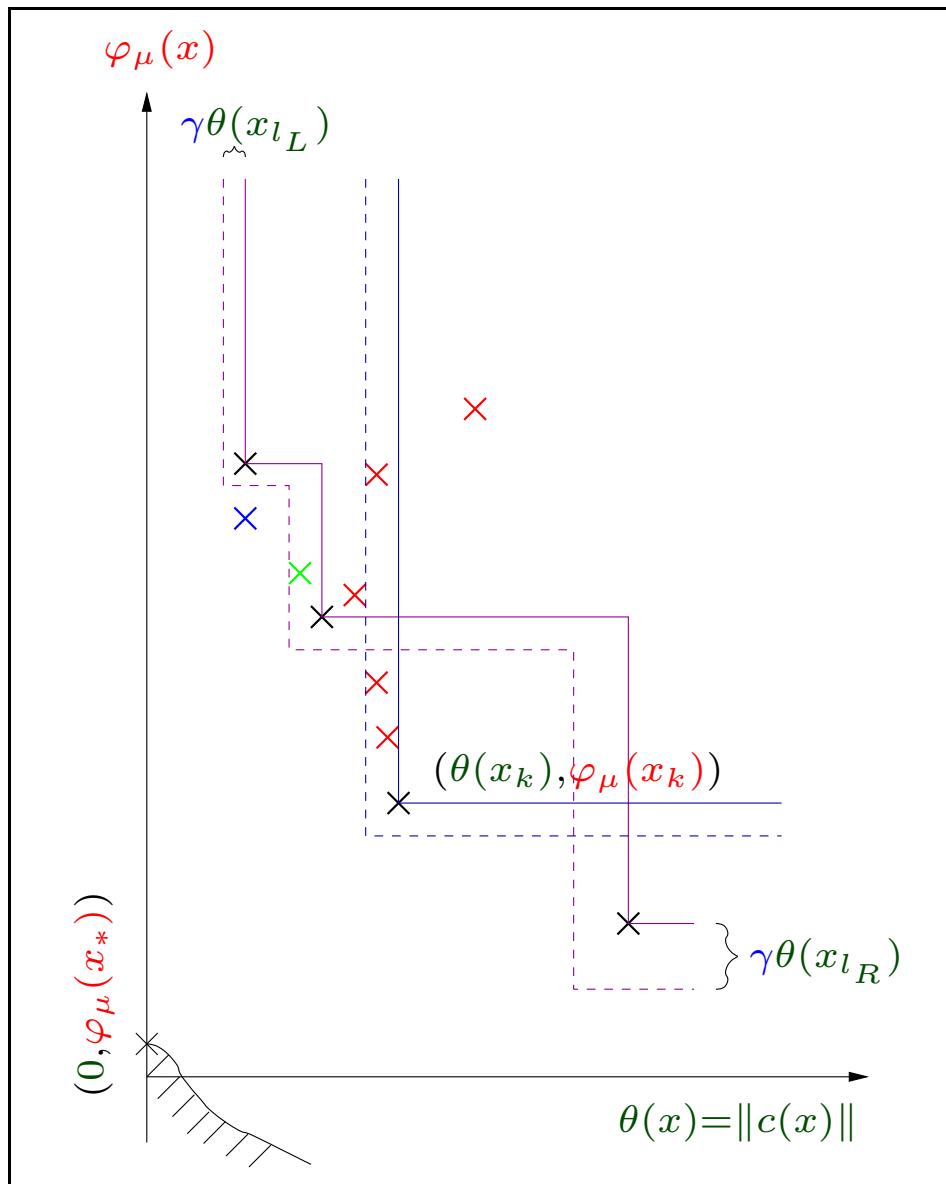
⇒ Don't augment F_k in that case

A Filter Line Search Method (Restoration)



If no admissible step size α_k can be found

A Filter Line Search Method (Restoration)



If no admissible step size α_k can be found



Revert to
feasibility restoration phase:

Decrease $\theta(x)$ until

- found acceptable new iterate $x_{k+1} := \tilde{x}_*^R > 0$, or
- converged to local minimizer of constraint violation

Primal-Dual Filter Line Search Method

- Solve barrier problems for $\mu_l \rightarrow 0$
 - Avoid combinatorial complexity of identifying active set

Primal-Dual Filter Line Search Method

- Solve barrier problems for $\mu_l \rightarrow 0$
 - Avoid combinatorial complexity of identifying active set
- Use the primal-dual formulation
 - Good convergence as $\mu \rightarrow 0$

Primal-Dual Filter Line Search Method

- Solve barrier problems for $\mu_l \rightarrow 0$
 - Avoid combinatorial complexity of identifying active set
- Use the primal-dual formulation
 - Good convergence as $\mu \rightarrow 0$
- Obtain search direction $\Delta x_k, \dots$ from linearization (Newton's method)
 - Fast local convergence
 - Usually computationally most expensive part (exploit structure)

Primal-Dual Filter Line Search Method

- Solve barrier problems for $\mu_l \rightarrow 0$
 - Avoid combinatorial complexity of identifying active set
- Use the primal-dual formulation
 - Good convergence as $\mu \rightarrow 0$
- Obtain search direction $\Delta x_k, \dots$ from linearization (Newton's method)
 - Fast local convergence
 - Usually computationally most expensive part (exploit structure)
- Determine maximal step sizes α_k^τ from fraction-to-the-boundary rule
 - Keep variables within bounds

Primal-Dual Filter Line Search Method

- Solve barrier problems for $\mu_l \rightarrow 0$
 - Avoid combinatorial complexity of identifying active set
- Use the primal-dual formulation
 - Good convergence as $\mu \rightarrow 0$
- Obtain search direction $\Delta x_k, \dots$ from linearization (Newton's method)
 - Fast local convergence
 - Usually computationally most expensive part (exploit structure)
- Determine maximal step sizes α_k^τ from fraction-to-the-boundary rule
 - Keep variables within bounds
- Perform backtracking line search using a filter approach
 - Ensure global convergence (W, Biegler, 2001)

Primal-Dual Filter Line Search Method

- Solve barrier problems for $\mu_l \rightarrow 0$
 - Avoid combinatorial complexity of identifying active set
- Use the primal-dual formulation
 - Good convergence as $\mu \rightarrow 0$
- Obtain search direction $\Delta x_k, \dots$ from linearization (Newton's method)
 - Fast local convergence
 - Usually computationally most expensive part (exploit structure)
- Determine maximal step sizes α_k^τ from fraction-to-the-boundary rule
 - Keep variables within bounds
- Perform backtracking line search using a filter approach
 - Ensure global convergence (W, Biegler, 2001)
- If line search “gets stuck”, revert to feasibility restoration phase
 - Remedy for global convergence failure
 - Fast detection of infeasibility

Software Implementation

- Implementation in Fortran 77 (and C) as “IPOPT”
 - Interfaces to AMPL and SIF (CUTEr)
 - Available on NEOS Server
 - Currently working on object-oriented reimplementations in C++
 - Available as open source code at COIN-OR

<http://www.coin-or.org/Ipopt>

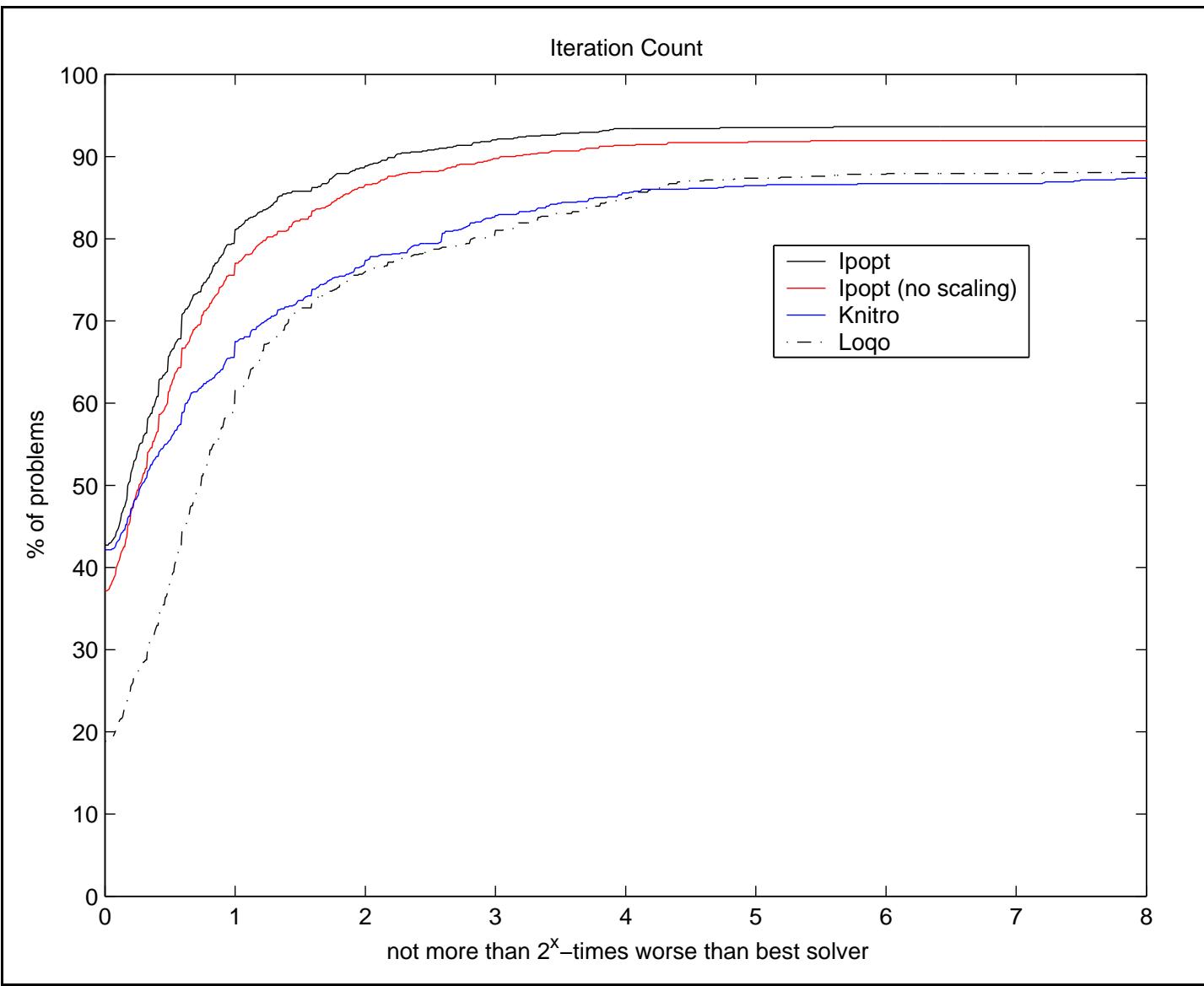
Software Implementation

- Implementation in Fortran 77 (and C) as “IPOPT”
 - Interfaces to AMPL and SIF (CUTEr)
 - Available on NEOS Server
 - Currently working on object-oriented reimplementations in C++
 - Available as open source code at COIN-OR

<http://www.coin-or.org/Ipopt>

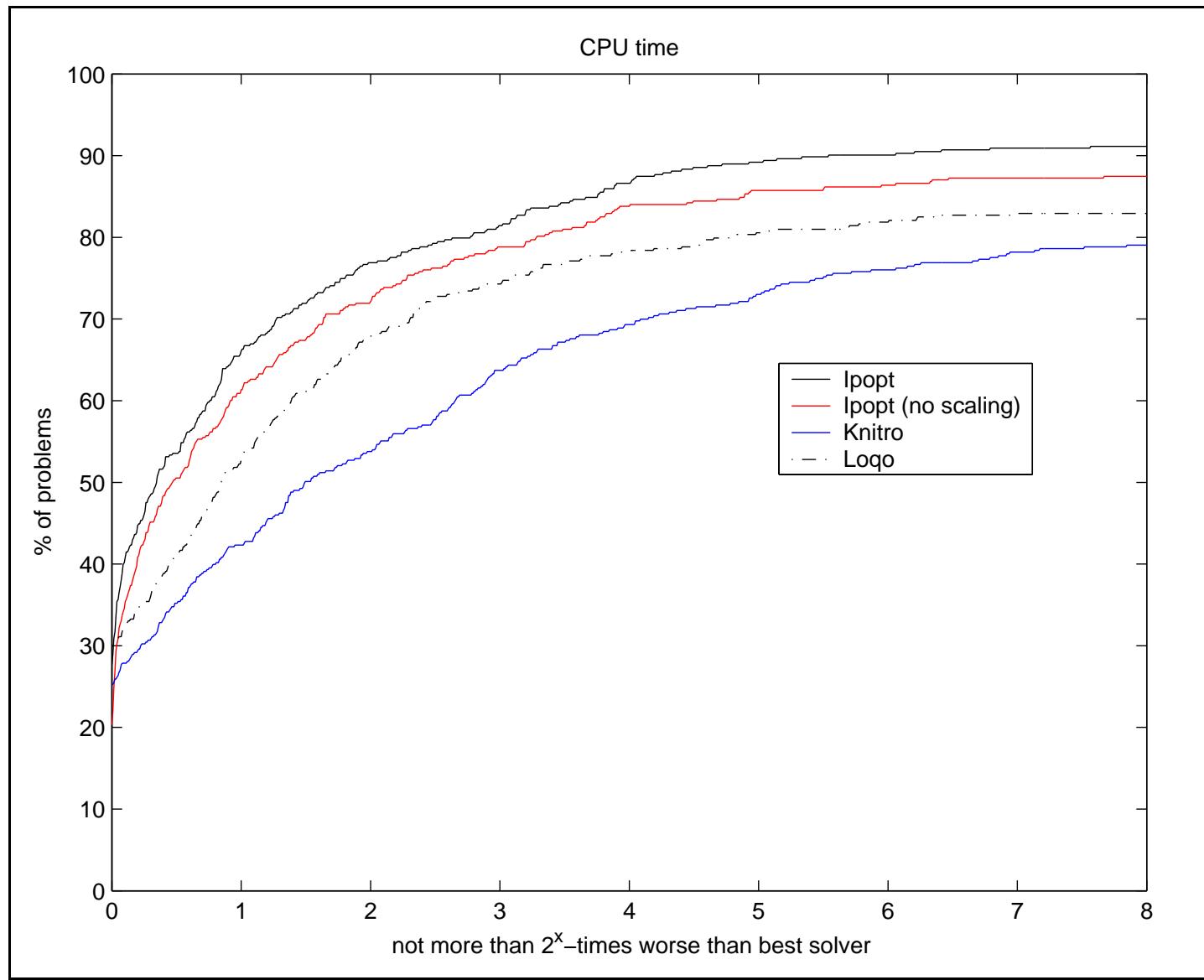
- Numerical results on CUTEr Problems (Gould, Orban, Toint)
 - 955 problems with $n = 2 \dots 125,050$ variables
 - Comparison (Feb 2004) with
 - KNITRO 3.1.1 (Byrd, Nocedal, Waltz, ...)
 - LOQO 6.06 (Benson, Shanno, Vanderbei)
 - Default options — CPU time limit 1h — max. iterations 3000
 - Performance profiles (Dolan, Moré, 2002)

Comparison Iteration Count — Solvers



955 total
75 $\Delta f^* > tol$
15 all fail

Comparison CPU Time — Solvers



511 total
48 $\Delta f^* > tol$
14 all fail

(skip problems with
 $t \leq 0.05s$)

Adaptive Barrier Parameter

Barrier Methods for NLP

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & c(x) = 0 \\ & x \geq 0\end{array}$$

→

$$\begin{array}{ll}\min & f(x) - \mu \sum \ln(x^{(i)}) \\ \text{s.t.} & c(x) = 0\end{array}$$

BP(μ)

- Fiacco-McCormick approach:
Solve barrier problem BP(μ_l) (approximately) for monotone sequence
of barrier parameters $\{\mu_l\} \rightarrow 0$
- Many different variants
(primal-dual, different handling of constraints)

Barrier Methods for NLP

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & c(x) = 0 \\ & x \geq 0\end{array}$$

→

$$\begin{array}{ll}\min & f(x) - \mu \sum \ln(x^{(i)}) \\ \text{s.t.} & c(x) = 0\end{array}$$

BP(μ)

- Fiacco-McCormick approach:
Solve barrier problem BP(μ_l) (approximately) for monotone sequence
of barrier parameters $\{\mu_l\} \rightarrow 0$
- Many different variants
(primal-dual, different handling of constraints)
- Global convergence: Ensure global convergence for each BP(μ_l)

Barrier Methods for NLP

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & c(x) = 0 \\ & x \geq 0\end{array}$$

→

$$\begin{array}{ll}\min & f(x) - \mu \sum \ln(x^{(i)}) \\ \text{s.t.} & c(x) = 0\end{array}$$

BP(μ)

- Fiacco-McCormick approach:
Solve barrier problem BP(μ_l) (approximately) for monotone sequence of barrier parameters $\{\mu_l\} \rightarrow 0$
- Many different variants
(primal-dual, different handling of constraints)
- Global convergence: Ensure global convergence for each BP(μ_l)
- Possible problems:
 - For given large μ , problem might be unbounded or difficult
 - Difficult to find good starting value μ_0 (scale dependent)
 - Fixing μ for several iterations might not be most efficient option
(don't really want to solve BP(μ_l))

Solving Primal-Dual Equations

$$\nabla f(x) - \nabla c(x)\lambda - z = 0$$

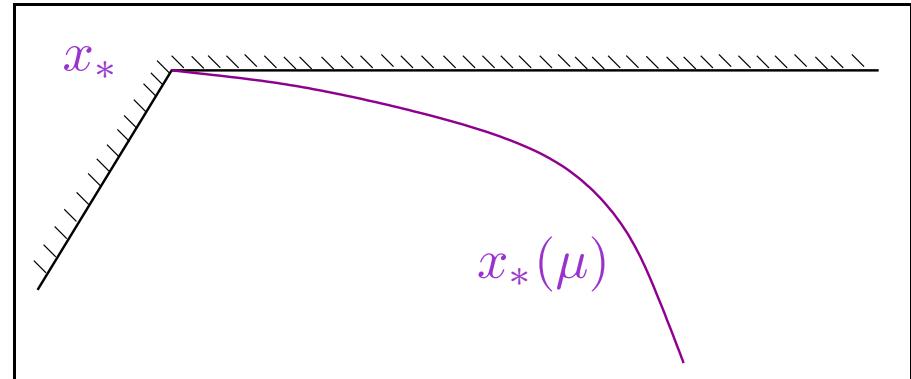
$$c(x) = 0$$

$$XZe = 0$$

$$x, z \geq 0$$

Solving Primal-Dual Equations

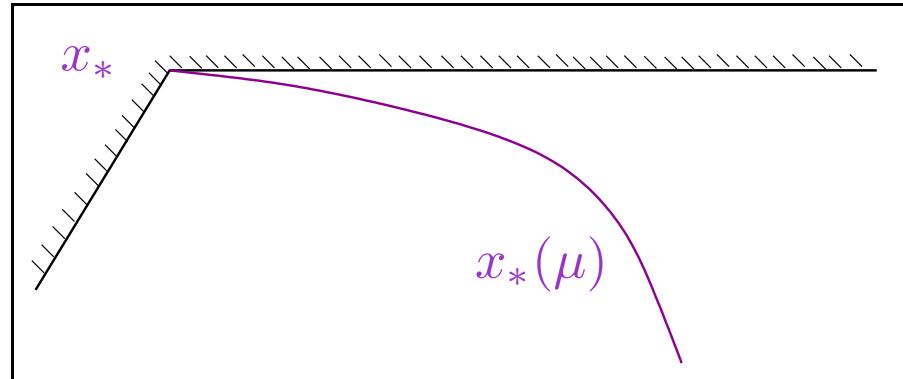
$$\begin{aligned}\nabla f(x) - \nabla c(x)\lambda - z &= 0 \\ c(x) &= 0 \\ XZe &= \mu e \\ x, z &\geq 0\end{aligned}$$



- Concentrate on solving the primal-dual equations
- Follow the central path $x_*(\mu)$ as $\mu \rightarrow 0$
- Methods adapt μ (non-monotonically) in every iteration
- Very effective for LP, QP, convex programming

Solving Primal-Dual Equations

$$\begin{aligned}\nabla f(x) - \nabla c(x)\lambda - z &= 0 \\ c(x) &= 0 \\ XZe &= \mu e \\ x, z &\geq 0\end{aligned}$$

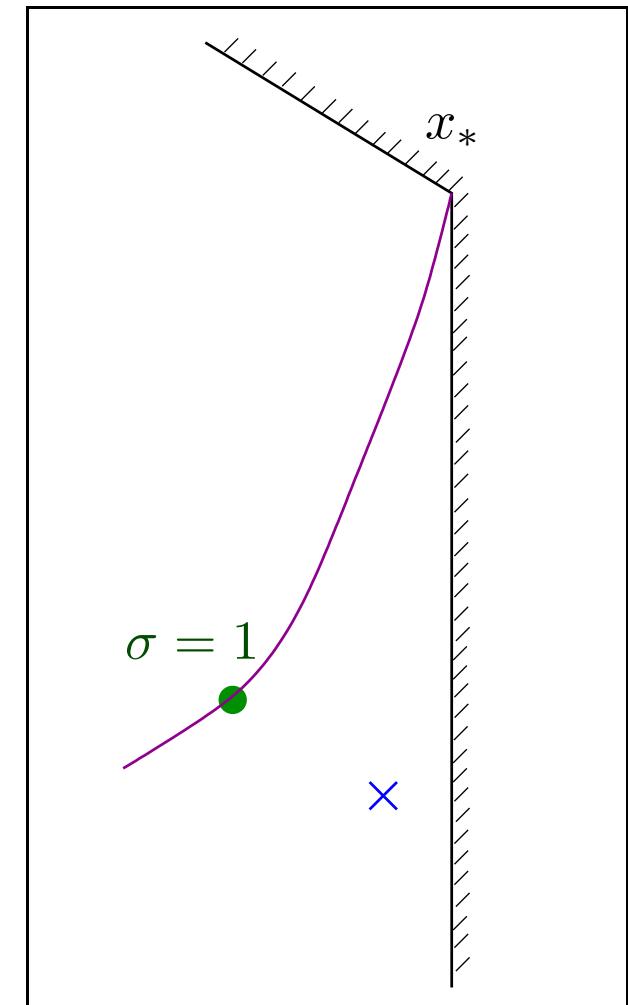


- Concentrate on solving the primal-dual equations
- Follow the central path $x_*(\mu)$ as $\mu \rightarrow 0$
- Methods adapt μ (non-monotonically) in every iteration
- Very effective for LP, QP, convex programming
- Global convergence issues:
 - Solving PD equations is good enough for convex problems
 - In nonconvex case: Ignoring $f(x)$ can lead more easily to convergence to saddle points or maxima

Choosing Adaptive Barrier Parameter

$$\mu = \sigma \frac{x^T z}{n}$$

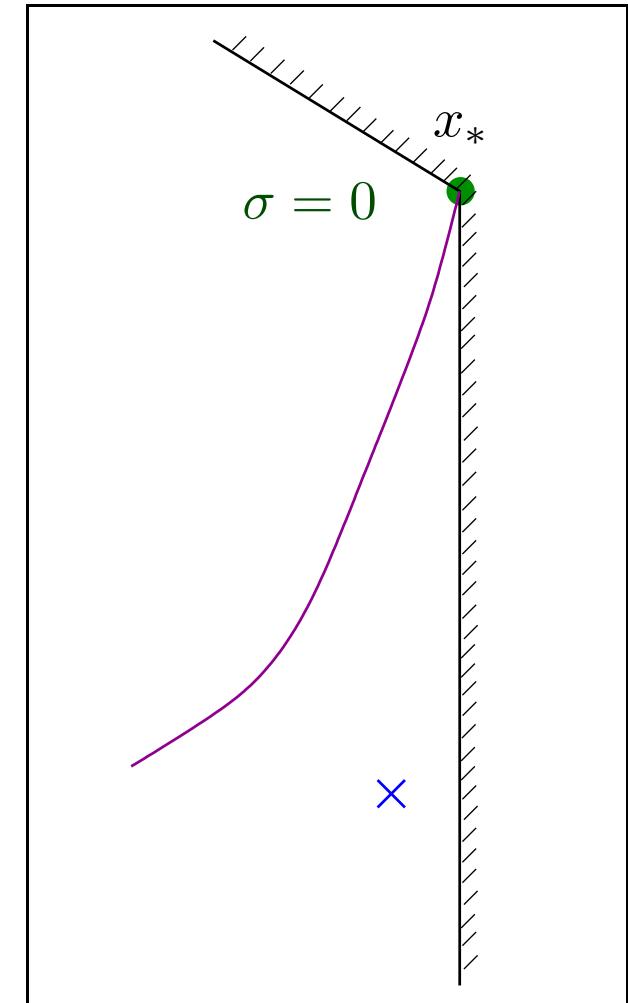
with centering parameter σ



Choosing Adaptive Barrier Parameter

$$\mu = \sigma \frac{x^T z}{n}$$

with centering parameter σ



Choosing Adaptive Barrier Parameter

$$\mu = \sigma \frac{x^T z}{n}$$

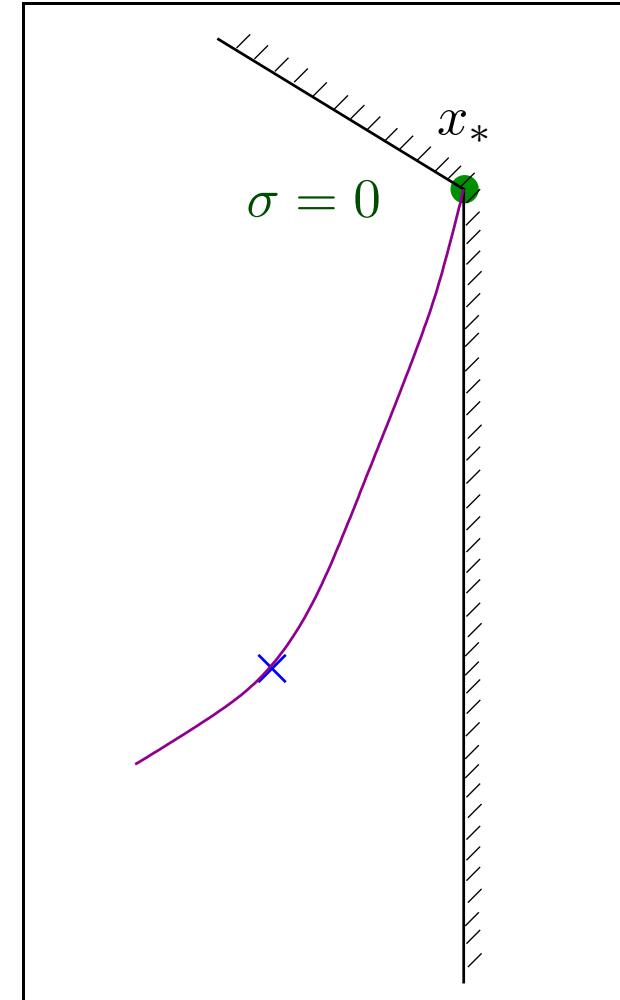
with centering parameter σ

Methods previously proposed:

1. LOQO centrality heuristic (formula)

(Vanderbei, Shanno, 1999)

- iterate central ($x^{(i)} z^{(i)} \equiv \frac{x^T z}{n}$) $\implies \sigma = 0$



Choosing Adaptive Barrier Parameter

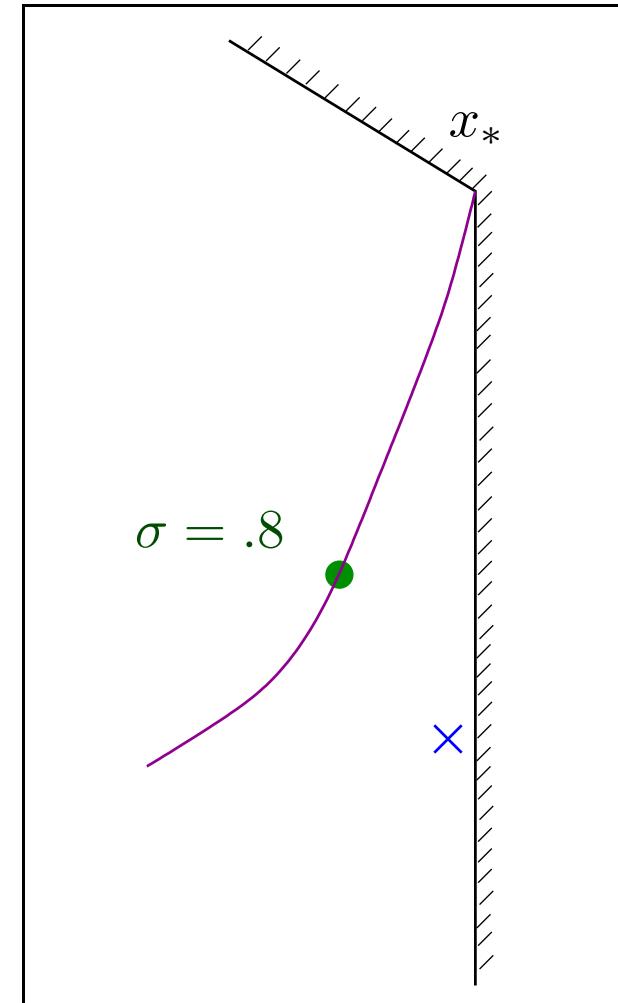
$$\mu = \sigma \frac{x^T z}{n}$$

with centering parameter σ

Methods previously proposed:

1. LOQO centrality heuristic (formula)
(Vanderbei, Shanno, 1999)

- iterate central ($x^{(i)} z^{(i)} \equiv \frac{x^T z}{n}$) $\implies \sigma = 0$
- iterate not central ($x^{(\tilde{i})} z^{(\tilde{i})} \ll \frac{x^T z}{n}$) $\implies \sigma = .8$



Choosing Adaptive Barrier Parameter

$$\mu = \sigma \frac{x^T z}{n}$$

with centering parameter σ

Methods previously proposed:

1. LOQO centrality heuristic (formula)

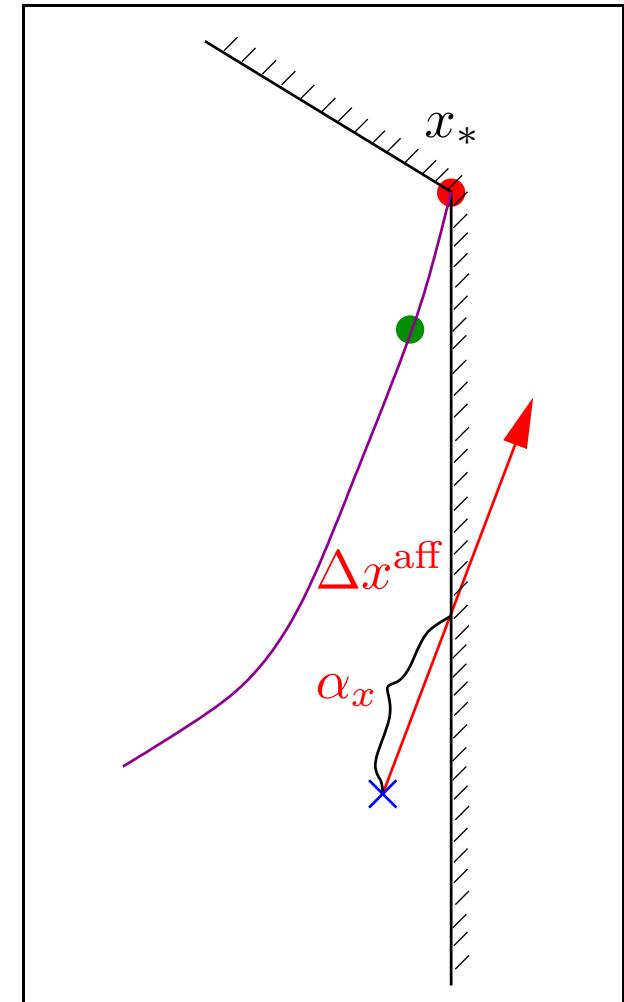
(Vanderbei, Shanno, 1999)

- iterate central ($x^{(i)} z^{(i)} \equiv \frac{x^T z}{n}$) $\implies \sigma = 0$
- iterate not central ($x^{(\tilde{i})} z^{(\tilde{i})} \ll \frac{x^T z}{n}$) $\implies \sigma = .8$

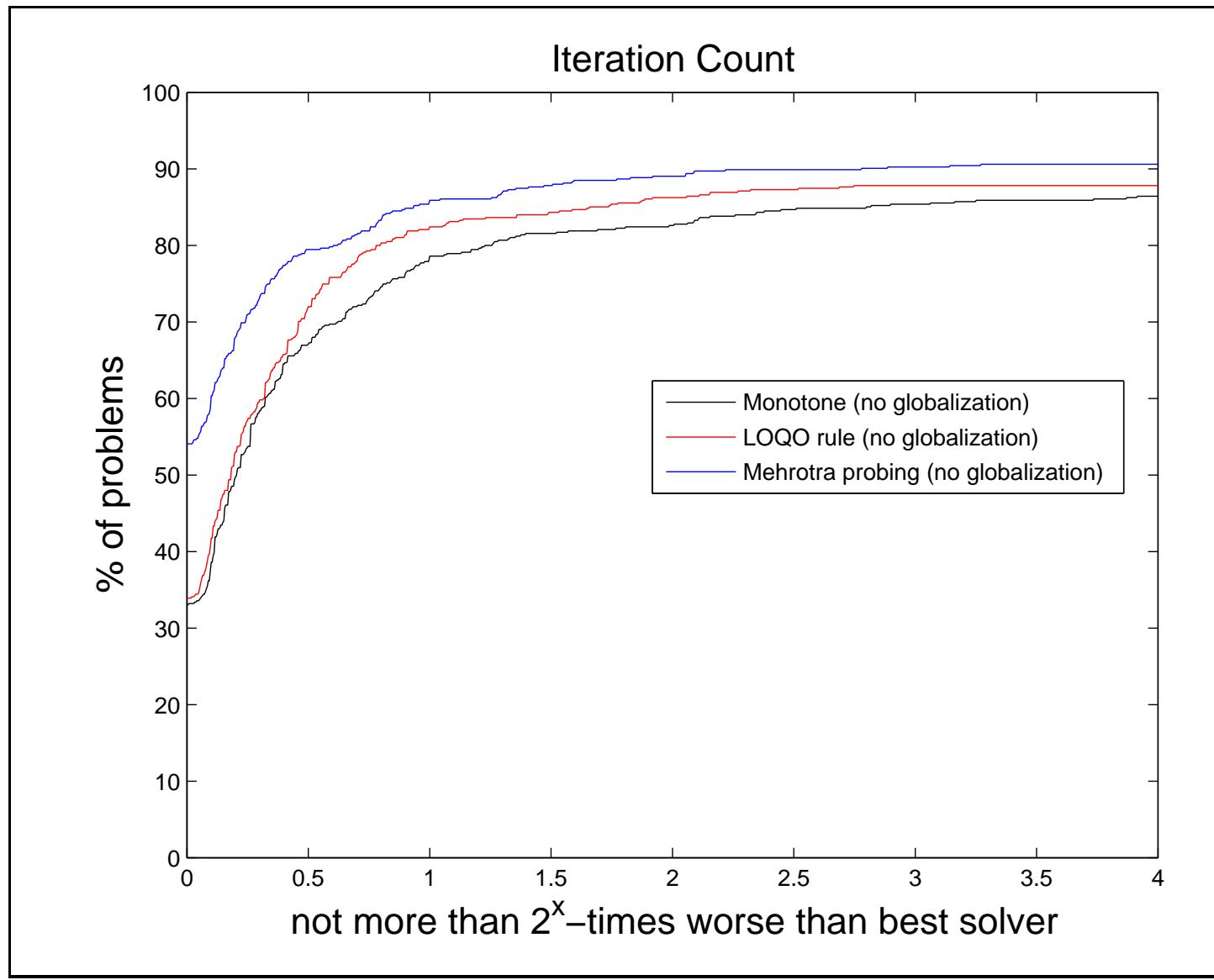
2. Mehrotra's probing heuristic for LP
(Mehrotra, 1992)

$$\sigma = \left(\frac{(x + \alpha_x \Delta x^{\text{aff}})^T (z + \alpha_z \Delta z^{\text{aff}})}{x^T z} \right)^3$$

(requires extra step computation)



Comparsion Adaptive Strategies (CUTEr)



599 total
24 $\Delta f^* > tol$

How to Find Best Centering Parameter?

$$\begin{bmatrix} W & A(x) & -I \\ A(x)^T & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{pmatrix} \Delta x_{\sigma}^{\text{pd}} \\ \Delta \lambda_{\sigma}^{\text{pd}} \\ \Delta z_{\sigma}^{\text{pd}} \end{pmatrix} = - \begin{pmatrix} \nabla f(x) + A(x)\lambda - z \\ c(x) \\ XZe - \sigma \frac{x^T z}{n} e \end{pmatrix}$$

How to Find Best Centering Parameter?

$$\begin{bmatrix} W & A(x) & -I \\ A(x)^T & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{pmatrix} \Delta x_{\sigma}^{\text{pd}} \\ \Delta \lambda_{\sigma}^{\text{pd}} \\ \Delta z_{\sigma}^{\text{pd}} \end{pmatrix} = - \begin{pmatrix} \nabla f(x) + A(x)\lambda - z \\ c(x) \\ XZe \end{pmatrix} + \sigma \begin{pmatrix} 0 \\ 0 \\ \frac{x^T z}{n} e \end{pmatrix}$$

- Compute Δ^{aff} and Δ^{cen} in two backsolves
- Then, $\Delta_{\sigma}^{\text{pd}} = \Delta^{\text{aff}} + \sigma \Delta^{\text{cen}}$ cheaply available for all $\sigma \geq 0$

How to Find Best Centering Parameter?

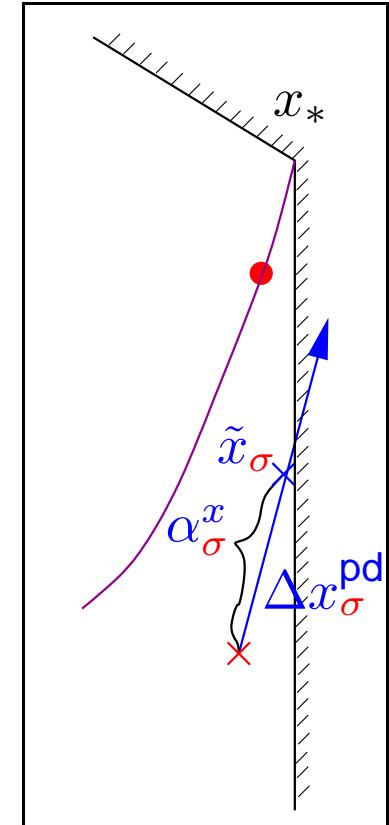
$$\begin{bmatrix} W & A(x) & -I \\ A(x)^T & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{pmatrix} \Delta x_{\sigma}^{\text{pd}} \\ \Delta \lambda_{\sigma}^{\text{pd}} \\ \Delta z_{\sigma}^{\text{pd}} \end{pmatrix} = - \begin{pmatrix} \nabla f(x) + A(x)\lambda - z \\ c(x) \\ XZe \end{pmatrix} + \sigma \begin{pmatrix} 0 \\ 0 \\ \frac{x^T z}{n} e \end{pmatrix}$$

- Compute Δ^{aff} and Δ^{cen} in two backsolves
- Then, $\Delta_{\sigma}^{\text{pd}} = \Delta^{\text{aff}} + \sigma \Delta^{\text{cen}}$ cheaply available for all $\sigma \geq 0$
- Given $\sigma \geq 0$, examine probing points

$$\begin{aligned}\tilde{x}_{\sigma} &= x + \alpha_{\sigma}^x \Delta x_{\sigma}^{\text{pd}} \\ \tilde{\lambda}_{\sigma} &= \lambda + \alpha_{\sigma}^z \Delta \lambda_{\sigma}^{\text{pd}} \\ \tilde{z}_{\sigma} &= z + \alpha_{\sigma}^z \Delta z_{\sigma}^{\text{pd}}\end{aligned}$$

where

$$\begin{aligned}\alpha_{\sigma}^x &= \max\{\alpha \in (0, 1] : x + \alpha \Delta x_{\sigma}^{\text{pd}} \geq (1 - \tau)x\} \\ \alpha_{\sigma}^z &= \max\{\alpha \in (0, 1] : z + \alpha \Delta z_{\sigma}^{\text{pd}} \geq (1 - \tau)z\}\end{aligned}$$



Quality Function For Centering Parameter

- Search for the “best” σ that minimizes some “quality function” $\Phi(\sigma)$

E.g.

$$\begin{aligned}\Phi(\sigma) = & \|\nabla f(\tilde{x}_\sigma) + A(\tilde{x}_\sigma)\tilde{\lambda}_\sigma - \tilde{z}_\sigma\|_2^2 + \\ & \|c(\tilde{x}_\sigma)\|_2^2 + \\ & \|\tilde{X}_\sigma \tilde{Z}_\sigma e\|_2^2\end{aligned}$$

- Do not want to evaluate functions and gradients

Quality Function For Centering Parameter

- Search for the “best” σ that minimizes some “quality function” $\Phi(\sigma)$

E.g.

$$\begin{aligned}\Phi(\sigma) = & \|\nabla f(\tilde{x}_\sigma) + A(\tilde{x}_\sigma)\tilde{\lambda}_\sigma - \tilde{z}_\sigma\|_2^2 + \\ & \|c(\tilde{x}_\sigma)\|_2^2 + \\ & \|\tilde{X}_\sigma \tilde{Z}_\sigma e\|_2^2\end{aligned}$$

- Do not want to evaluate functions and gradients
- “Linear quality function”

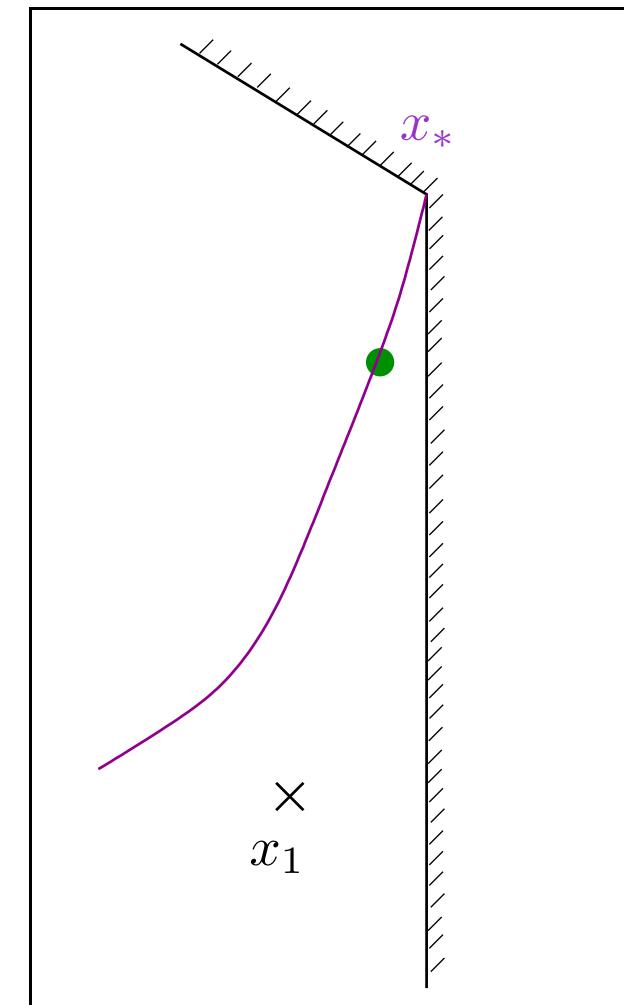
$$\begin{aligned}\Phi(\sigma) = & (1 - \alpha_\sigma^z)^2 \|\nabla f(x_k) + A(x_k)\lambda_k - z_k\|_2^2 + \\ & (1 - \alpha_\sigma^x)^2 \|c(x_k)\|_2^2 + \\ & \|\tilde{X}_\sigma \tilde{Z}_\sigma e\|_2^2\end{aligned}$$

- Accurately predicts primal-dual error at probing point for LPs
- Inexpensive to compute
- $\Phi(\sigma)$ is non-smooth, non-convex; use golden bisection

A General Globalization Framework

For each iteration k :

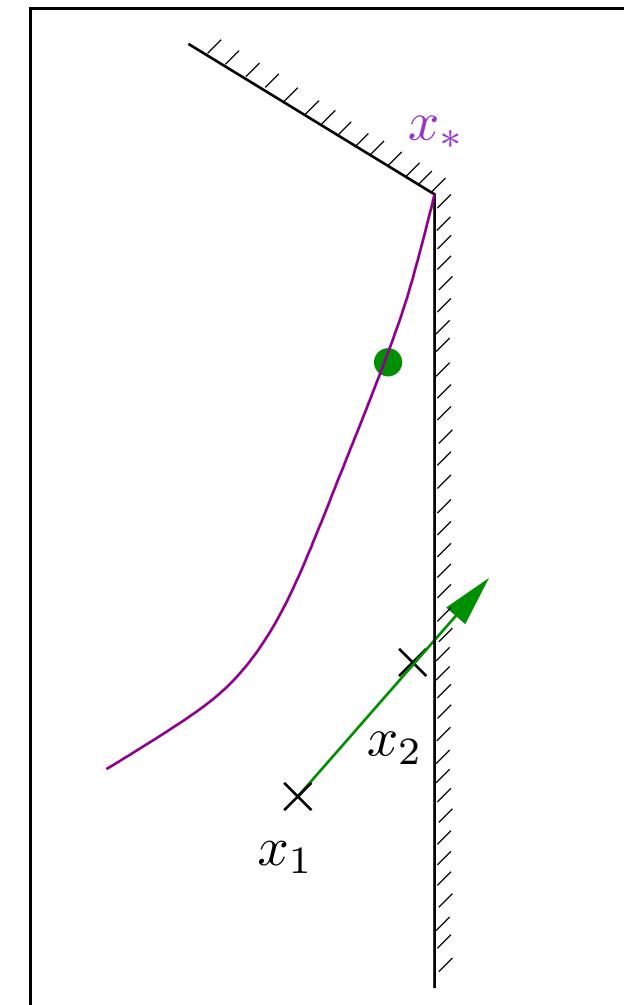
1. Choose μ_k by adaptive rule (*free mode*)



A General Globalization Framework

For each iteration k :

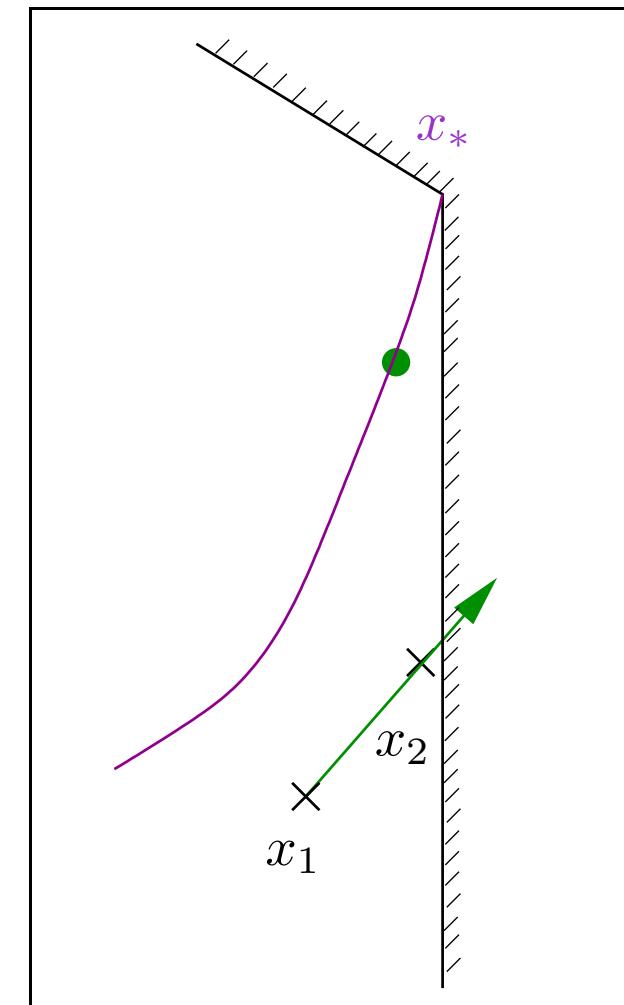
1. Choose μ_k by adaptive rule (*free mode*)
2. Compute step Δ^{pd} for $\text{BP}(\mu_k)$, and x_{k+1}



A General Globalization Framework

For each iteration k :

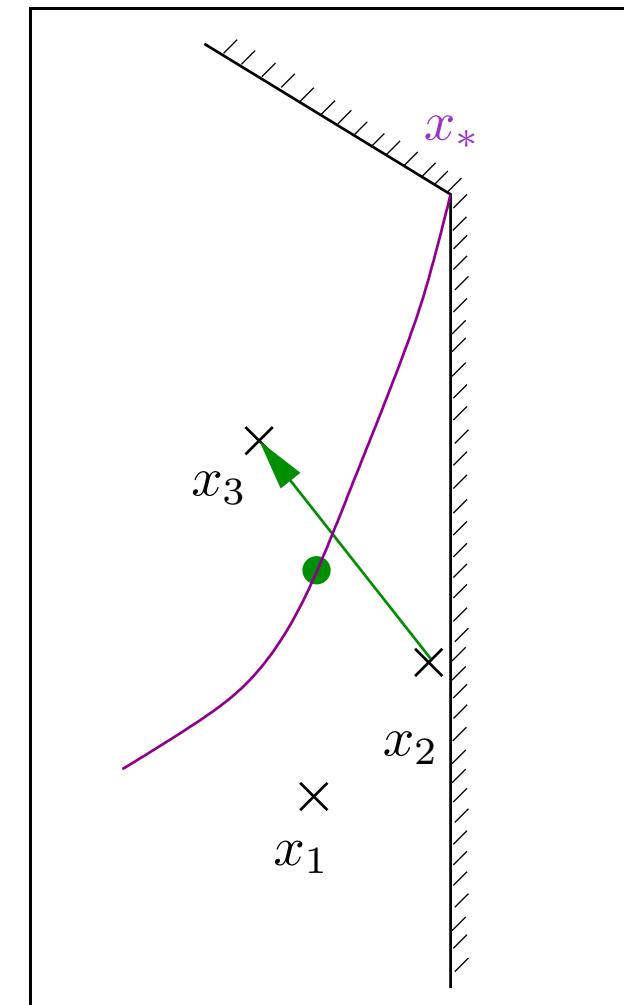
1. Choose μ_k by adaptive rule (*free mode*)
2. Compute step Δ^{pd} for $\text{BP}(\mu_k)$, and x_{k+1}
3. Progress for original NLP?
(monitor $\|\text{KKT_error}_k\|$)



A General Globalization Framework

For each iteration k :

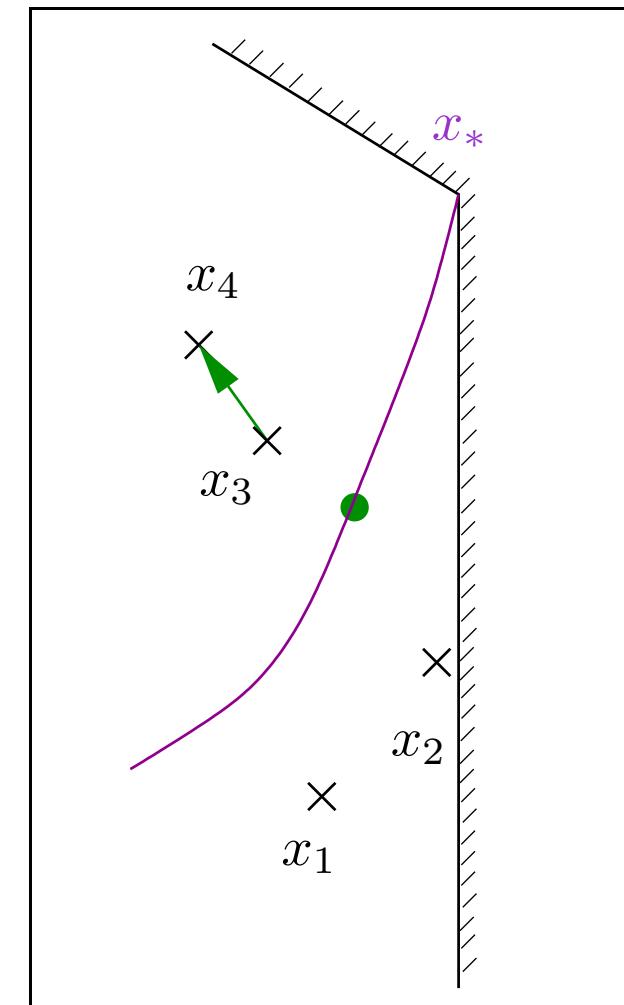
1. Choose μ_k by adaptive rule (*free mode*)
2. Compute step Δ^{pd} for $\text{BP}(\mu_k)$, and x_{k+1}
3. Progress for original NLP?
(monitor $\|\text{KKT_error}_k\|$)
 - Yes: Continue with next iteration



A General Globalization Framework

For each iteration k :

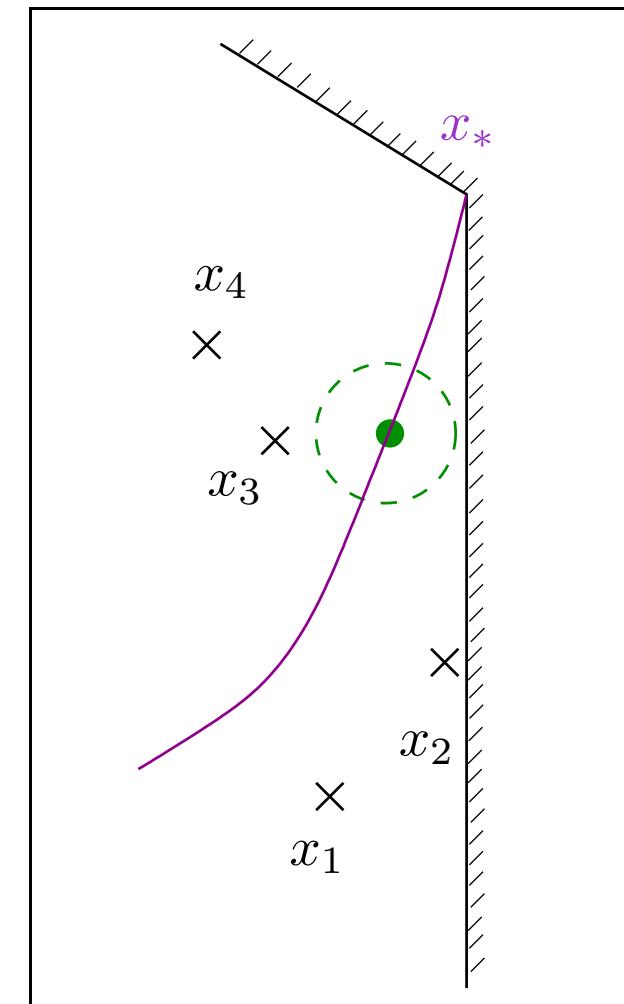
1. Choose μ_k by adaptive rule (*free mode*)
2. Compute step Δ^{pd} for $\text{BP}(\mu_k)$, and x_{k+1}
3. Progress for original NLP?
(monitor $\|\text{KKT_error}_k\|$)
 - Yes: Continue with next iteration
 - No:



A General Globalization Framework

For each iteration k :

1. Choose μ_k by adaptive rule (*free mode*)
2. Compute step Δ^{pd} for $\text{BP}(\mu_k)$, and x_{k+1}
3. Progress for original NLP?
(monitor $\|\text{KKT_error}_k\|$)
 - Yes: Continue with next iteration
 - No: (*monotone mode*)
 - (a) execute robust, monotone Fiacco-McCormick algorithm
(approx. solve $\text{BP}(\bar{\mu})$ problems)
 - (b) if suff. progress for original NLP,
return to *free mode*



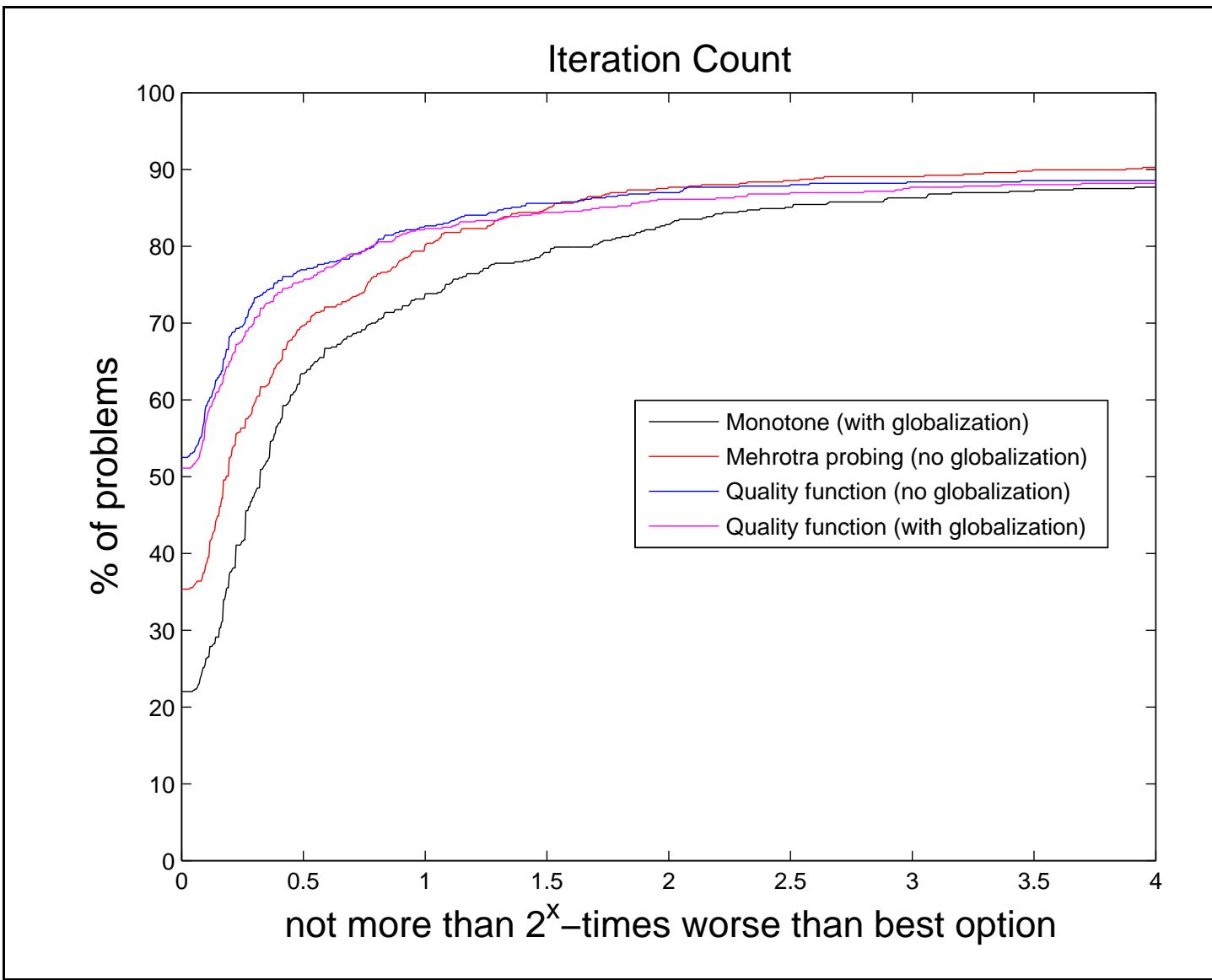
Comparison Iteration Count (Netlib LPs)



98 total
0 $\Delta f^* > tol$

(No starting point
heuristic)

Comparison Iteration Count (CUTEr)



599 total
22 $\Delta f^* > tol$

COIN-OR (www.coin-or.org)

- COmputational INfrastructure for Operations Research
- Open source repository for optimization software
- Initiated by IBM at ISMP 2000
- Since 2004 non-profit foundation (hosted by INFORMS)
- Many projects, e.g.
 - CLP (Coin LP Solver)
 - OSI (Open Solver Interface) [CPLEX, OSL, Xpress MP, CLP, GLPK]
 - CGL (Cut Generation Library)
 - SMI (Stochastic Modeling Interface)
 - CBC (Coin Branch and Cut Library)
 - IPOPT, DFO, NLP-API, Multifario, Tabu-Search, ...
- Also models/data sets (for benchmarking, allow recreating results from papers), frameworks
- Contribution are WELCOME!

Summary

- Optimality conditions for nonlinear optimization
- Basic interior point framework
 - Primal vs. primal-dual steps
 - Monotone, Fiacco-McCormick algorithm
- Globalization
 - Example for failure of basic method
 - Filter line search procedure
- Adaptive choice of the barrier parameter
 - Linear quality function
 - General globalization framework
- Numerical results <http://www.coin-or.org/Ipopt>
- Collaborations: - Larry Biegler, Carl Laird
 - Richard Byrd, Jorge Nocedal, Richard Waltz