Advances in Interior-Point Methods Nonlinear Optimization

Richard A. Waltz, Northwestern University

rwaltz@ece.northwestern.edu

joint with: Richard Byrd, Jorge Nocedal and Andreas Wächter

International Conference on Continuous Optimization, ICCOPT I

Troy, New York, USA

3 August, 2004

Motivation

Want to make interior methods for NLP more robust

Difficulties caused by:

- bad starting points
- bad problem scaling
- sensitivity to barrier update procedure
- ...

Outline

- Initial Point techniques for interior methods
  - Background - LP (convex QP)
  - Extension to NLP
  - Propose new strategies
- Relation to barrier update strategy
- Numerical Results

NLP Problem

NLP:

$$\begin{array}{lll} \min\limits_{x} & f(x) & \\ \text{s.t.} & h_i(x) = 0, & i \in \mathcal{E} \\ & g_i(x) \geq 0, & i \in \mathcal{I} \end{array}$$

NLP Problem

NLP:

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = 0, \quad i \in \mathcal{E} \\ & g_i(x) \geq 0, \quad i \in \mathcal{I} \end{aligned}$$

Reformulated NLP:

$$\begin{aligned} \min_{x,s} \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = 0, \quad i \in \mathcal{E} \\ & g_i(x) - s_i = 0, \quad i \in \mathcal{I} \\ & s \geq 0 \end{aligned}$$

Barrier Sub-problem

Barrier Sub-problem:

$$\begin{aligned} \min_{x,s} \quad & f(x) - \mu \sum_{i=1}^{m} \ln s_i \\ \text{s.t.} \quad & h_i(x) = 0, \quad i \in \mathcal{E} \\ & g_i(x) - s_i = 0, \quad i \in \mathcal{I} \end{aligned}$$

Drive $\mu \to 0$.

Initial Point - Background

Initial-point selection a key heuristic for LP interior codes

Example (PCx on 93 Netlib w/wout intial-point)

- default settings: solves 89
- setting $x^0 = e, z^0 = e$: solves 28

Initial point also important for NLP barrier codes
(...but little work on initial point for NLP)

- Gertz, Nocedal, Sartenaer (2003)

Initial Point - LP

Standard LP:

$$\begin{array}{rl} \min\limits_{x} & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array}$$

Optimality conditions:

$$\begin{array}{rl} \min\limits_{x} & A^T y + z = c \\ \text{s.t.} & Ax = b \\ & x_i z_i = 0, \forall i \\ & x, z \geq 0 \end{array}$$

Initial Point - LP

Mehrotra approach:

$$\min_x \|x\|^2 \qquad \text{s.t.} \quad Ax = b$$

$$\tilde{y} = (AA^T)^{-1}Ac$$

$$\tilde{z} = c - A^T\tilde{y}$$

$$x^0 \leftarrow (\tilde{x} + \delta_x e)$$

$$y^0 \leftarrow \tilde{y}$$

$$z^0 \leftarrow (\tilde{z} + \delta_z e)$$

Initial Point - LP

Andersen etal approach: Solve:

$$\begin{array}{ll} \min\limits_{x} & c^T x + \frac{\tau}{2}\|x\|^2 \\ \text{s.t.} & Ax = b \end{array}$$

Set: $x_i = \max(1, x_i)\ \forall i$

Similarly for dual variables.

- Lustig, Marsten, Shanno (1991)
- Mehrotra (1992)
- Andersen, Gondzio, Meszaros, Xu (1996)
- S. Wright (1997)

Extension to NLP

- Want to be away from bounds and well-centered
- Not trivial to locate a feasible point
- Dual feasibility involves both primal and dual variables
- Often times want to respect user's initial point x
  - choose initial slacks s
  - choose initial multipliers (y, z)
  - choose initial barrier parameter μ
- Large initial multipliers may distort model and introduce nonconvexities
- Must be concerned with bad points later on as well

NLP: initial point strategy 1

- x^0 assume provided by user; left unchanged

NLP: initial point strategy 1

- x^0 assume provided by user; left unchanged
- choose $\tilde{s}$ such that $g(x^0) - \tilde{s} = 0$

NLP: initial point strategy 1

- x^0 assume provided by user; left unchanged
- choose $\tilde{s}$ such that $g(x^0) - \tilde{s} = 0$
- compute $(\tilde{y}, \tilde{z})$ as least squares solution of
  $\nabla f(x^0) - A_h(x^0)^T y - A_g(x^0)^T z = 0$

NLP: initial point strategy 1

- x^0 assume provided by user; left unchanged
- choose $\tilde{s}$ such that $g(x^0) - \tilde{s} = 0$
- compute $(\tilde{y}, \tilde{z})$ as least squares solution of
  $\nabla f(x^0) - A_h(x^0)^T y - A_g(x^0)^T z = 0$
- $s^0 \leftarrow \max(\tilde{s}, 0.1), \quad y^0 \leftarrow \tilde{y}, \quad z^0 \leftarrow \max(\tilde{z}, 0.1)$

NLP: initial point strategy 1

- x^0 assume provided by user; left unchanged
- choose $\tilde{s}$ such that $g(x^0) - \tilde{s} = 0$
- compute $(\tilde{y}, \tilde{z})$ as least squares solution of
  $\nabla f(x^0) - A_h(x^0)^T y - A_g(x^0)^T z = 0$
- $s^0 \leftarrow \max(\tilde{s}, 0.1), \quad y^0 \leftarrow \tilde{y}, \quad z^0 \leftarrow \max(\tilde{z}, 0.1)$
- $\mu^0 = \frac{(s^0)^T z^0}{m}$

NLP: initial point strategy 2

Given x^0, algorithm chooses some s, y, z
Compute affine-scaling (Newton) step:

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & A_h & A_g \\ 0 & SZ & 0 & -S \\ A_h^T & 0 & 0 & 0 \\ A_g^T & -S & 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ S^{-1}d_s \\ -\tilde{y} \\ -\tilde{z} \end{bmatrix} = - \begin{bmatrix} \nabla f(x^0) \\ 0 \\ h(x^0) \\ g(x^0) - s \end{bmatrix}$$

$$\tilde{s} = s + d_s, \quad \delta_s = \max_i(0, -\tilde{s}_i), \quad \delta_z = \max_i(0, -\tilde{z}_i)$$

$$s^0 = \tilde{s} + 2\delta_s + 1, \quad y^0 = \tilde{y}, \quad z^0 = \tilde{z} + 2\delta_z + 1$$

$$\mu^0 = \max(\frac{(s^0)^T z^0}{m}, r_p^0, r_d^0)$$

NLP: initial point strategy 3

Given x^0, algorithm chooses some s, y, z
Compute affine-scaling (Newton) step:

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & A_h & A_g \\ 0 & SZ & 0 & -S \\ A_h^T & 0 & 0 & 0 \\ A_g^T & -S & 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ S^{-1}d_s \\ -\tilde{y} \\ -\tilde{z} \end{bmatrix} = - \begin{bmatrix} \nabla f(x^0) \\ 0 \\ h(x^0) \\ g(x^0) - s \end{bmatrix}$$

$$\tilde{s} = s + d_s, \quad \bar{\delta}_s = \max(0, -\tilde{s}), \quad \bar{\delta}_z = \max(0, -\tilde{z})$$

$$s^0 = \tilde{s} + 2\bar{\delta}_s + 1, \quad y^0 = \tilde{y}, \quad z^0 = \tilde{z} + 2\bar{\delta}_z + 1$$

$$\mu^0 = \max\left(\frac{(s^0)^T z^0}{m}, r_p^0, r_d^0\right)$$

Relation to barrier update strategy

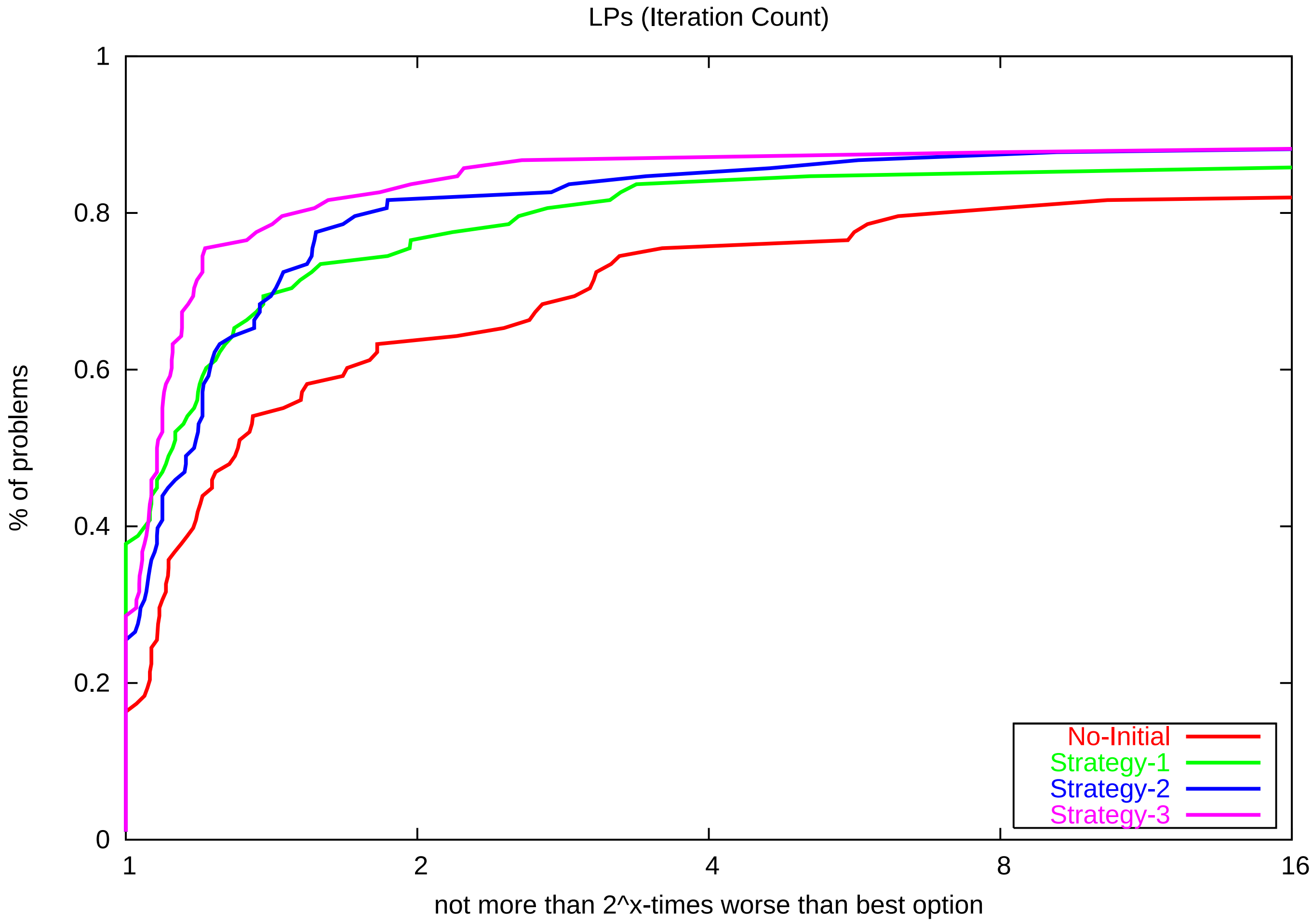
- large initial μ, requires barrier strategy which can decrease μ quickly
- the performance of barrier strategies can be very sensitive to the initial point
- may get bad *intermediate* points which require a dynamic barrier strategy to center you

(see Andreas Wächter's talk)

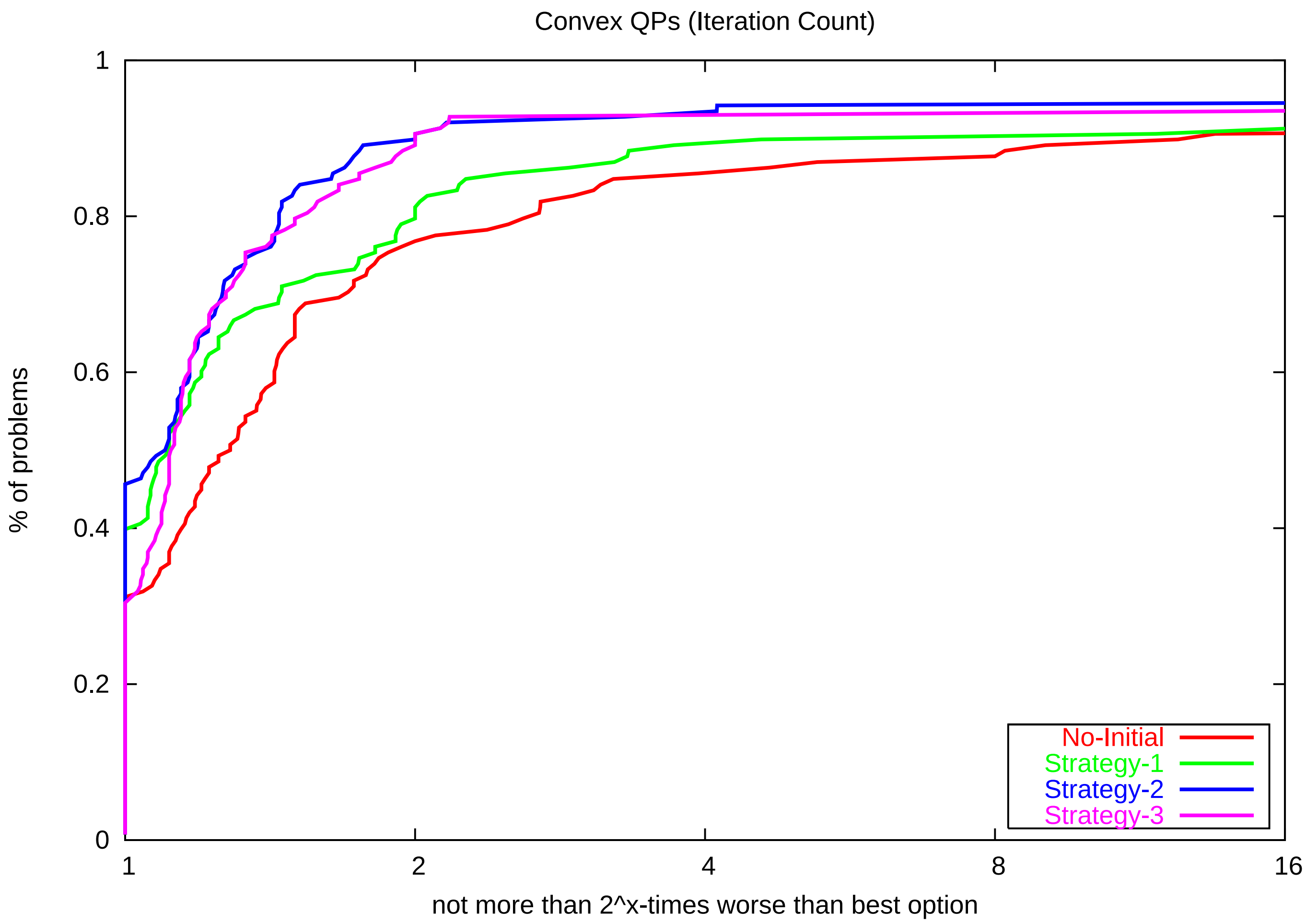
Numerical Results - Overview

- look at LPs (Netlib: 98); convex QPs (Maros-Meszaros: 138); general NLPs (CUTEr: 312)
- implemented in KNITRO-DIRECT using *probing* rule for barrier update
- probing = Mehrotra predictor+centering step (but no corrector!)
- GOAL 1: Develop general strategy which competes with current LP/convex QP strategies and doesn't hurt NLPs
- GOAL 2: competitive on LPs/QPs and significantly improves NLP performance

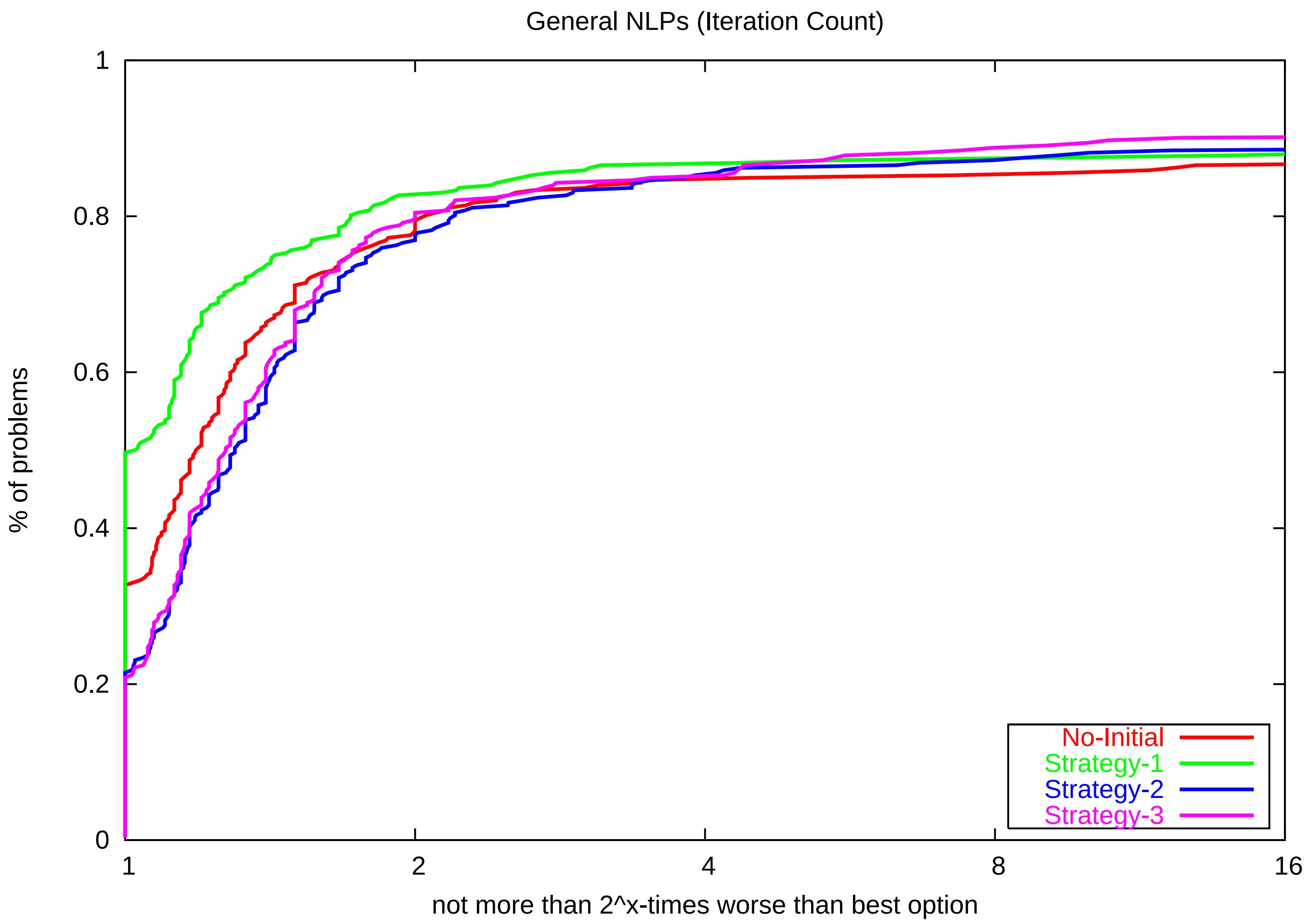
Numerical Results - LPs



Numerical Results - convex QPs



Numerical Results - general NLPs



Conclusions/Future Work

- generalized initial point strategies for LPs to general NLPs
- generalized versions seem to work well on LPs/convex QPs
- moderate success with one strategy on NLPs but poor performance with others
- negative curvature issue requires more investigation
- look at strategies which change the initial *primal* variables x (but not too much)