SQP, SLP and Interior-Point methods for large-scale nonlinear programming

Nick Gould

Rutherford Appleton Laboratory

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) \;\text{ subject to }\; c_{\mathcal{E}}(x) = 0 \;\text{ and }\; c_{\mathcal{I}}(x) \geq 0$$

IMA Workshop on Optimization in Simulation-Based Models, 9th–16th January, 2003

Joint in parts with Richard Byrd, Jorge Nocedal, Dominique Orban, Philippe Toint and Richard Waltz

NONLINEAR PROGRAMMING

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; f(x) \;\; \text{subject to} \;\; c_{\mathcal{E}}(x) = 0, \;\; c_{\mathcal{I}}(x) \geq 0$$

- f, $c_{\mathcal{E}}$, $c_{\mathcal{I}}$ smooth (preferably C^2)
- no convexity assumptions $\Longrightarrow$ content with local minimizers
- n, $m \stackrel{\text{def}}{=} |\mathcal{E}| + |\mathcal{I}|$ large, say $O(10^4)$ – $O(10^6)$
- Jacobians , Hessians sparse and/or structured
- in general, constraints may be
  - bounded on both sides: $c_{\mathcal{I}}^l \leq c_{\mathcal{I}}(x) \leq c_{\mathcal{I}}^u$
  - simple bounds on variables: $x^l \leq x \leq x^u$
  - linear (or linear network): $a_{\mathcal{E}}^T x = b_{\mathcal{E}}, \;\; a_{\mathcal{I}}^T x \geq b_{\mathcal{I}}$
  - nonlinear

GALAHAD

Aims:

- build a **threadsafe fortran 90 library** of optimization modules designed to cope with a variety of commonly-occurring problems
- in particular, produce a/some successor(s) to LANCELOT

For GALAHAD 1.0 (April 2002), concentrated on

- improvements to LANCELOT A $\Longrightarrow$ LANCELOT B
- two algorithms for (non-convex) quadratic programming:

$$\min_{x \in \mathbb{R}^n} \; q(x) = g^T x + \tfrac{1}{2} x^T H x \;\; \text{subject to} \;\; A_{\mathcal{E}} x = b_{\mathcal{E}}, \;\; A_{\mathcal{I}} x \geq b_{\mathcal{I}}$$

- auxiliary packages for pre-solving QPs, solving trust-region subproblems, sparse linear systems, sorting, . . .

QPB — an interior point trust-region QP solver

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; g^T x + \tfrac{1}{2} x^T H x \;\; \text{subject to} \;\; Ax \geq b$$

- uses a sequential minimization of the **barrier function**

  $$\Phi(x, c, \mu) = g^T x + \tfrac{1}{2} x^T H x - \mu e^T \log c \;\; \text{subject to} \;\; Ax - c = b$$

  in a trust-region framework, keeping $c > 0$
- feasibility maintained throughout — start near analytic centre
- primal-dual model approximately solved as an EQP via preconditioned projected CG (G., Hribar, Nocedal)
- strong underlying convergence theory (Conn, G., Orban, Sartenaer, Toint)

QPA — an active-set QP solver

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\;\; g^T x + \tfrac{1}{2}x^T H x \;\;\text{subject to}\;\; Ax \geq b$$

- uses a traditional active set method with basic step computation

  $$\underset{s\in\mathbb{R}^n}{\text{minimize}}\;\; g_k^T s + \tfrac{1}{2}s^T H s \;\;\text{subject to}\;\; A_k s = 0$$

  for some subset A_k of A

- EQP subproblem solved via preconditioned projected CG

  (G., Hribar, Nocedal)

- **preconditioner** changes by low-rank inertia-controlled update
  — Schur complement updating used (Gill, Murray, Saunders, Wright)

- actually use a single-phase penalty method (Conn, Sinclair)

- many technical details (G., Toint)

QPA -vs- QPB

- interior-point QPB usually better, and often far better, than active-set QPA when cold started

Name	n	m	type	its	time
QPBAND	500000	250000	C	17	**2909**
QPNBAND	500000	250000	NC	13	**181**
PORTSQP	1000000	1	C	11	**72**
PORTSNQP	1000000	2	NC	22	**210**

Compaq AlphaServer DS20
(3.5 Gbytes RAM)
QPB
C = convex
NC = nonconvex
time in CPU seconds

- when warm started (good prediction of active set known), QPA often outperforms QPB **except** on highly degenerate or ill-conditioned examples

GALAHAD 2.0 and the future

In May 2002 (SIOPT meeting, Toronto) I predicted

" . . . next release, GALAHAD 2.0, will include at least

- ⊙ SQP methods
  - ⋄ our implementation of Fletcher's Sℓ_1QP
  - ⋄ our implementation of Fletcher & Leyffer's SQP-filter approach"

I no longer am convinced of this!

What changed?

SQP — SEQUENTIAL QUADRATIC PROGRAMMING

$$\underset{x \in \mathbb{R}^n}{\textbf{minimize}} \; f(x) \; \textbf{subject to} \; c(x) \geq 0$$

Basic SQP method:

- from current solution estimate x, compute step s to

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \; s^T g(x) + \tfrac{1}{2} s^T H s \; \text{ subject to } \; A(x)s + c(x) \geq 0$$

  + (possibly) a trust region constraint $\|s\| \leq \Delta$,
  where

  - $g(x) \stackrel{\text{def}}{=} \nabla_x f(x)$
  - $A(x) \stackrel{\text{def}}{=} \nabla_x c(x)$
  - $H \approx \nabla_{xx}[f(x) + y^T c(x)]$ for some multipliers y

- globalize using an appropriate merit function

SQP — DRAWBACKS (I)

$$\textbf{minimize}_{s \in \mathbb{R}^n} \; s^T g(x) + \tfrac{1}{2} s^T H s \;\; \textbf{subject to} \;\; A(x)s + c(x) \geq 0$$

The SQP step computation is too coarse/expensive a calculation, especially in early iterations

- alternative cheap steps like those for KNITRO or LOQO far more cost effective . . . conjugate gradients, equality constraints
- contrasts with "cheap" truncated Newton steps for unconstrained minimization (Dembo, Steihaug)
- could truncate QP calculation . . . but how in general ? (Murray, Prieto)

$\Longrightarrow$ **inefficiency**

SQP — DRAWBACKS (II)

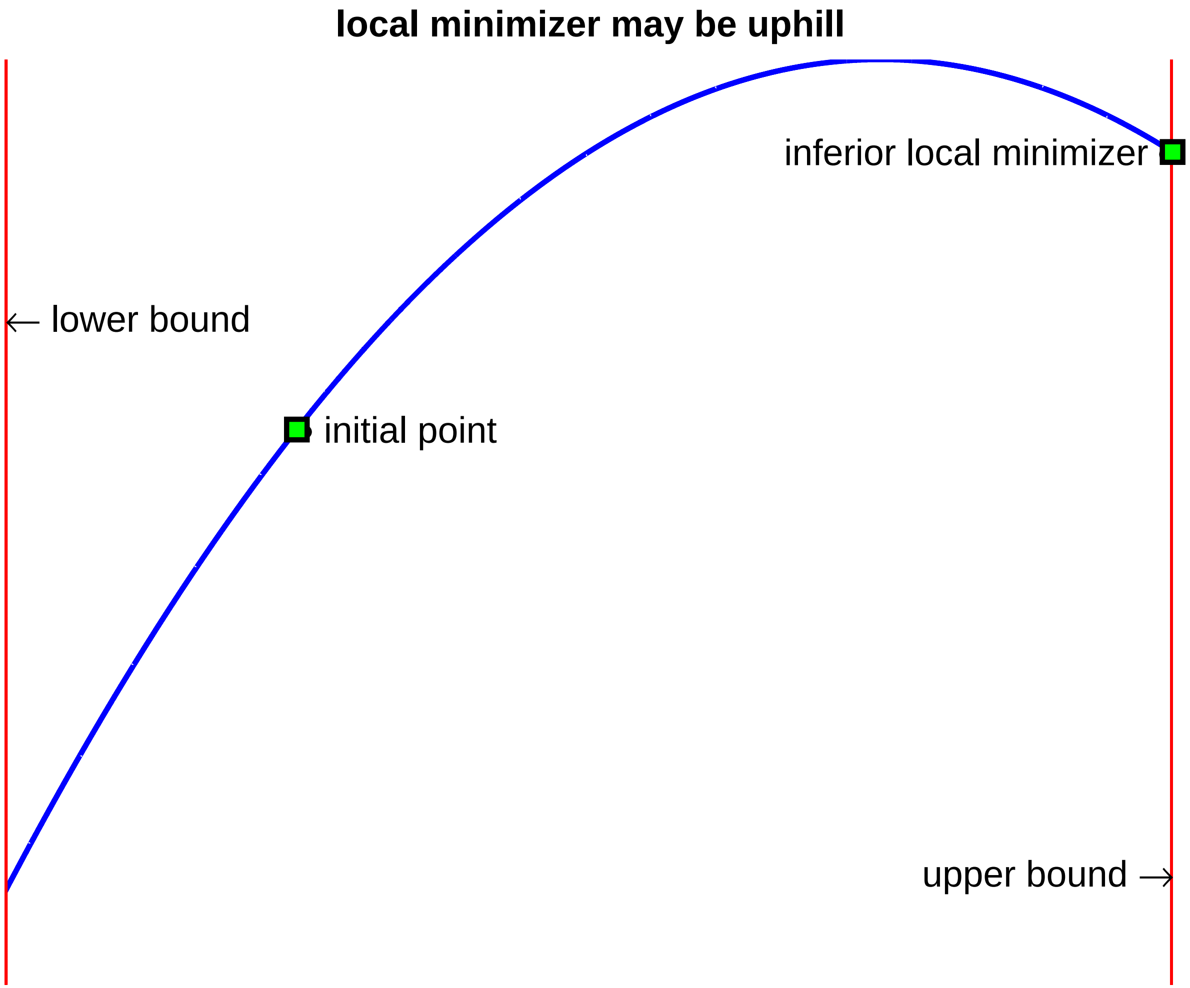
$$\underset{s \in \mathbb{R}^n}{\textbf{minimize}} \;\; s^T g(x) + \tfrac{1}{2} s^T H s \;\; \textbf{subject to} \;\; A(x)s + c(x) \geq 0$$

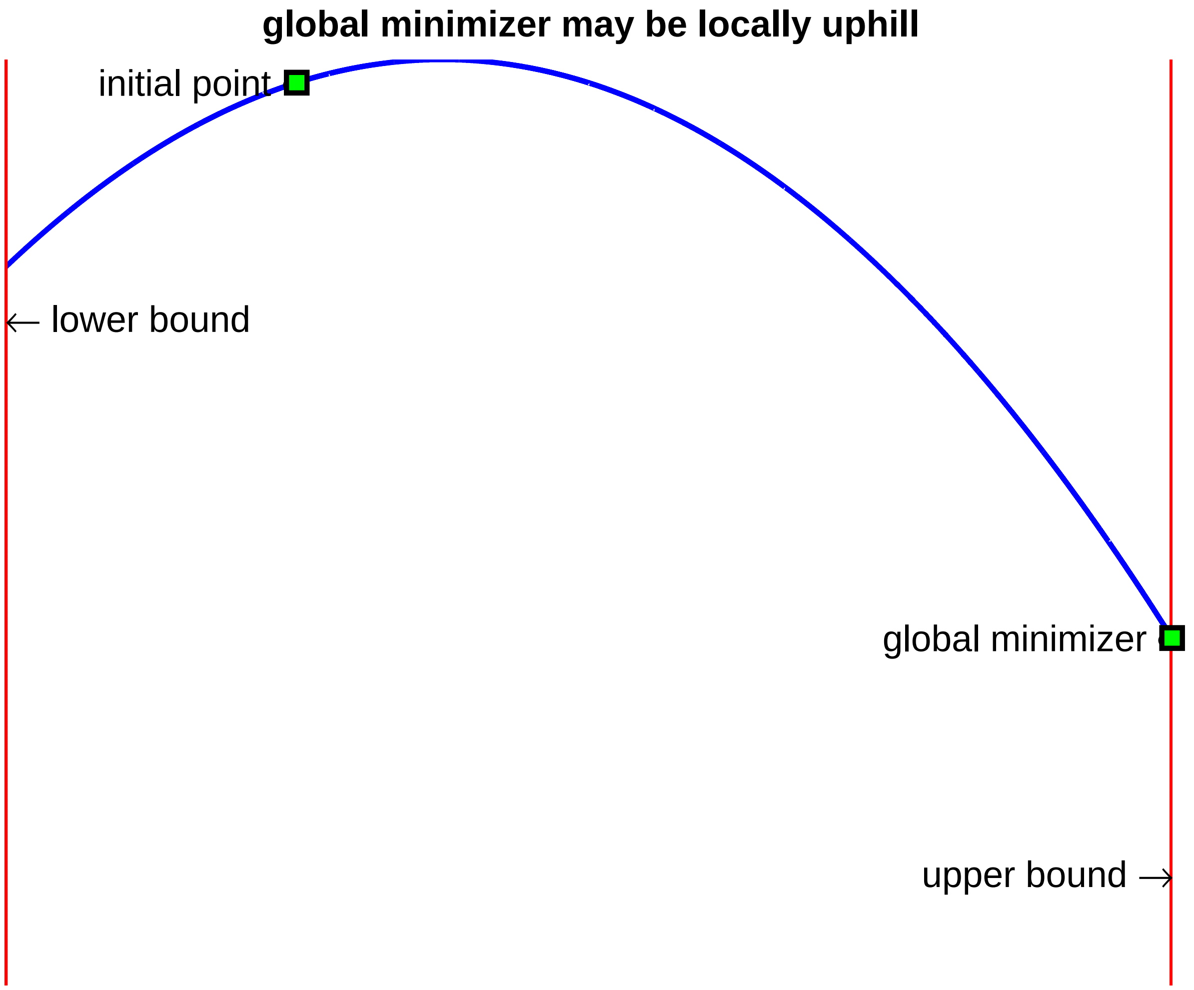
In the best possible case, would like to use exact 2nd derivatives, but ...

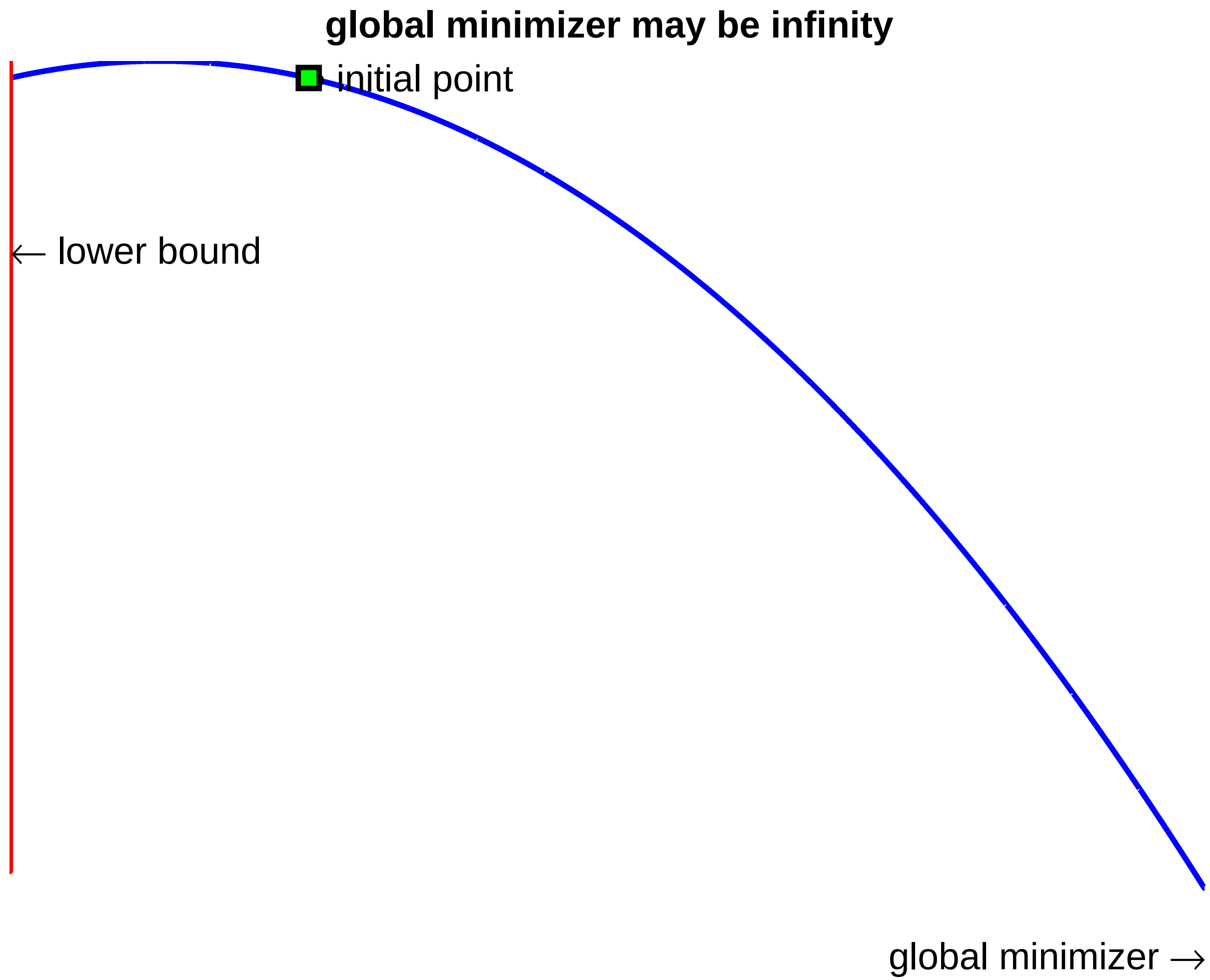
the SQP step may be inappropriate if H is indefinite

- ⊙ local minimizers may be uphill — bad with IP methods
- ⊙ ultimately lower local (or global) minimizers may initially lead uphill if the step is a direction of negative curvature (Goldsmith)
- ⊙ QP may be unbounded from below

$\Longrightarrow$ **inefficiency** or even **catastrophe**


local minimizer may be uphill
inferior local minimizer
← lower bound
initial point
upper bound →


global minimizer may be locally uphill
initial point
← lower bound
global minimizer
upper bound →


global minimizer may be infinity
initial point
← lower bound
global minimizer →

WHAT ARE THE ALTERNATIVES?

What kind of problems can we solve efficiently?

- **unconstrained problems**
  - truncated Newton, (preconditioned) conjugate gradients
- **linear programs**
  - Simplex and Interior Point methods
- **equality-constrained quadratic programs**
  - projected (preconditioned) conjugate gradients
  - can incorporate trust-region constraint using Lanczos

Suggests using any of above as subproblems

SEQUENTIAL LINEAR PROGRAMMING (SLP)

$$\underset{x \in \mathbb{R}^n}{\textbf{minimize}} \; f(x) \; \textbf{subject to} \; c_{\mathcal{E}}(x) = 0 \; \textbf{and} \; c_{\mathcal{I}}(x) \geq 0$$

Find a correction Δx to solution estimate x:

$$\underset{\Delta x \in \mathbb{R}^n}{\text{minimize}} \; \Delta x^T \nabla_x f \; \text{subject to} \; \nabla_x c_{\mathcal{E}} \Delta x + c_{\mathcal{E}} = 0 \; \text{and} \; \nabla_x c_{\mathcal{I}} \Delta x + c_{\mathcal{I}} \geq 0$$

good
- ⊙ simple
- ⊙ potentially solve huge problems — $n, |\mathcal{I}| = O(10^7\text{–}10^8)$

bad
- ⊙ slow — at best linearly convergent
- ⊙ constraints may be inconsistent
- ⊙ no “natural” merit function
- ⊙ need good LP solver — most are “commercial”

NON-DIFFERENTIABLE PENALTY METHODS

$$\underset{x \in \mathbb{R}^n}{\textbf{minimize}} \;\; f(x) \;\; \textbf{subject to} \;\; c_{\mathcal{I}}(x) \geq 0$$

ℓ_1 penalty method: "solve"

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; f(x) + \rho \| \min(c_{\mathcal{I}}(x), 0) \|_1$$

for sufficiently large ρ — **exact** penalty function

Non-smooth problem $\Longrightarrow$ can't use off-the-shelf method $\Longrightarrow$

Sℓ_1-LP method

(Conn, Fletcher)

Find a correction Δx to solution estimate x:

$$\underset{\Delta x \in \mathbb{R}^n}{\text{minimize}} \;\; \Delta x^T \nabla_x f + \rho \| \min(\nabla_x c_{\mathcal{I}} \Delta x + c_{\mathcal{I}}, 0) \|_1$$

Sℓ_1-LP METHOD (cont.)

good
- ⊙ no inconsistency of constraints
- ⊙ natural merit function
- ⊙ globally convergent
- ⊙ potentially solve huge problems — $n, |\mathcal{I}| = O(10^7\text{–}10^8)$
- ⊙ ℓ_1-LP can be reformulated as an LP

bad
- ⊙ need special LP solver
- ⊙ slow — at best linearly convergent
- ⊙ needs a strategy for selection of parameter ρ
- ⊙ ρ diverges if problem is inconsistent

Sℓ_1-LP-EQP

(Fletcher, Sainz de la Maza)

$$\underset{x \in \mathbb{R}^n}{\textbf{minimize}} \; f(x) \; \textbf{subject to} \; c_{\mathcal{I}}(x) \geq 0$$

- find the set $\mathcal{A} \subseteq \mathcal{I}$ of **active** constraints from

$$\underset{\Delta x \in \mathbb{R}^n}{\text{minimize}} \; \Delta x^T \nabla_x f + \rho \| \min(\nabla_x c_{\mathcal{I}} \Delta x + c_{\mathcal{I}}, 0) \|_1$$

  for which $\nabla_x c_i \Delta x + c_i = 0$ for $i \in \mathcal{A}$

- then find a correction Δx to "approximately"

$$\underset{\Delta x \in \mathbb{R}^n}{\text{minimize}} \; \Delta x^T \nabla_x f + \tfrac{1}{2} \Delta x^T H \Delta x \; \text{ subject to } \; \nabla_x c_{\mathcal{A}} \Delta x + c_{\mathcal{A}} = 0$$

  for some symmetric H

$S\ell_1$-LP-EQP (cont.)

good
- ⊙ LP asymptotically determines "correct" active set
- ⊙ equality QP (EQP) ensures superlinear asymptotic convergence if $H \longrightarrow \nabla_{xx}[f(x) + y^T c(x)]$
- ⊙ compromise between simplicity of SLP and speed of SQP

bad
- ⊙ need good LP solver — most are "commercial"
- ⊙ may not identify "correct" active set fast
- ⊙ need efficient way to solve EQP — truncated projected conjugate gradients

SLIQUE, AN ℓ_1-LP-EQP TRUST-REGION METHOD

Overview: (Byrd, G., Nocedal, Waltz)

Two trust regions, a "Cauchy" point & an overall trial step

- a **trust-region to control the "LP" step**

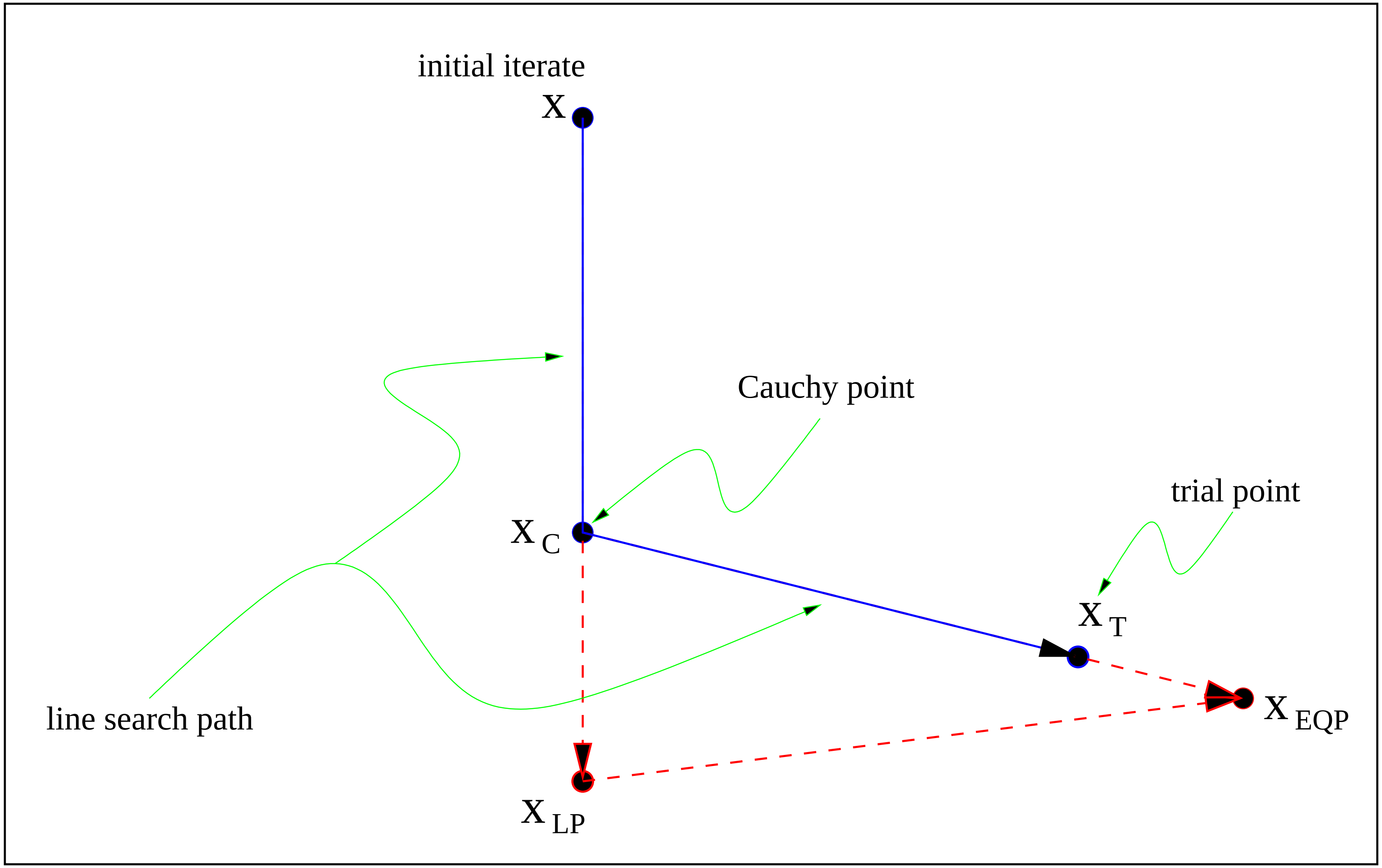
$$\Delta x_{\mathrm{LP}} = \arg\min_{\|\Delta x\|_\infty \le \Delta^{\mathrm{LP}}} \Delta x^T \nabla_x f + \rho \| \min(\nabla_x c_{\mathcal{I}} \Delta x + c_{\mathcal{I}}, 0)\|_1$$

- a **Cauchy point** $\Delta x_{\mathrm{CP}} = \alpha_{\mathrm{CP}} \Delta x_{\mathrm{LP}}$ for some $\alpha_{\mathrm{CP}} \in (0,1]$
- a **feasibility step** $\Delta x_{\mathrm{F}} = \arg\min_{\|\Delta x\|_2 \le \beta\Delta} \|\nabla_x c_{\mathcal{A}} \Delta x + c_{\mathcal{A}}\|_2, \quad \beta \in (0,1]$
- an (independent) **trust-region to control the "EQP" step**

$$\begin{aligned} \Delta x_{\mathrm{EQP}} = \ & \arg\min_{\|\Delta x\|_2 \le \Delta} \ \Delta x^T \nabla_x f + \tfrac{1}{2} \Delta x^T H \Delta x \\ & \text{subject to} \quad \nabla_x c_{\mathcal{A}} \Delta x + c_{\mathcal{A}} = \nabla_x c_{\mathcal{A}} \Delta x_{\mathrm{F}} + c_{\mathcal{A}} \end{aligned}$$

- an **overall trial step** $\Delta x_{\mathrm{T}} = \Delta x_{\mathrm{CP}} + \alpha_{\mathrm{T}} (\Delta x_{\mathrm{EQP}} - \Delta x_{\mathrm{CP}})$ for some $\alpha_{\mathrm{T}} \in (0,1]$

THE OVERALL TRIAL STEP

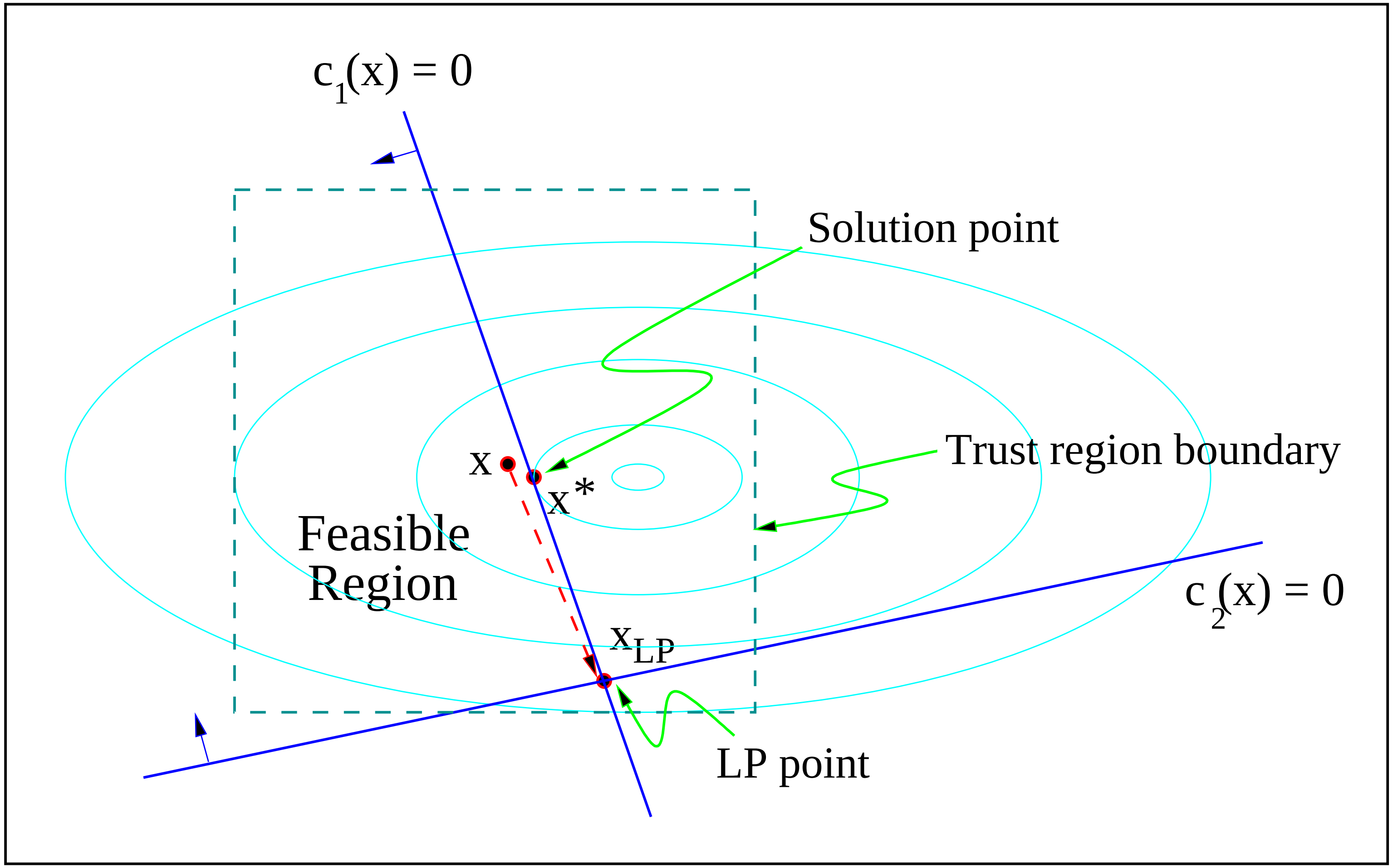


THE LP STEP

The **LP trust region**

- ⊙ stops large steps if the LP model is unbounded from below
- ⊙ must not be too large as otherwise optimally "inactive" constraints will appear in the active set
- ⊙ does not have to lie within overall trust region

THE LP TRUST REGION SHOULD NOT BE TOO BIG



THE CAUCHY STEP

The **Cauchy stepsize** α_{CP}

- ⊙ ensures that the Cauchy point lies within the overall trust region
- ⊙ ensures that the linear and quadratic model decreases are "similar"
- ⊙ should not be too small as to prevent convergence

Typically α^{CP} is required to approximate the minimizer of

$$\alpha \Delta x_{\text{LP}}^T \nabla_x f + \tfrac{1}{2}\alpha^2 \Delta x_{\text{LP}}^T H \Delta x_{\text{LP}} + \rho \| \min(\alpha \nabla_x c_{\mathcal{I}} \Delta x_{\text{LP}} + c_{\mathcal{I}}, 0) \|_1$$

within the intersection of the LP and EQP trust-regions

THE EQP STEP

The **EQP trust region**

- ⊙ stops large steps if the EQP model is unbounded from below
- ⊙ must shrink if progress is impossible otherwise
  - ⋄ measure progress by comparing decrease in EQP model with actual decrease in the penalty function — typical trust-region mechanism
- ⊙ should not shrink to zero unnecessarily as this will prevent both global and fast local convergence

THE OVERALL STEP

The **overall trial stepsize** α_{T}

- ⊙ ensures that the EQP and quadratic model decreases are "similar"
- ⊙ should ultimately be 1

STEP ACCEPTANCE

New point x^+ given as

$$x^+ = \begin{cases} x + \Delta x_{\mathrm{T}} & \text{if } \ \rho \geq 1.0^{-8} \\ x & \text{otherwise} \end{cases}$$

where

$$\rho = \frac{\phi(x) - \phi(x + \Delta x_{\mathrm{T}})}{\Delta x_{\mathrm{T}}^T \nabla_x f + \frac{1}{2}\Delta x_{\mathrm{T}}^T H \Delta x_{\mathrm{T}} + \rho \| \min(\nabla_x c_{\mathcal{I}} \Delta x_{\mathrm{T}} + c_{\mathcal{I}}, 0)\|_1}$$

and

$$\phi(x) = f(x) + \rho \| \min(c_{\mathcal{I}}(x), 0)\|_1$$

TRUST-REGION RADII UPDATES

LP radius update:

$$\Delta^{\text{LP}+} = \begin{cases} \min(\max\{1.2\|\Delta x_{\text{T}}\|_\infty, 1.2\|\Delta x_{\text{CP}}\|_\infty, 0.1\Delta^{\text{LP}}, 7\Delta^{\text{LP}}) & \text{if } \rho \geq 10^{-8} \\ \min(\max\{0.5\|\Delta x_{\text{T}}\|_\infty, 0.1\Delta^{\text{LP}}\}, \Delta^{\text{LP}}) & \text{otherwise} \end{cases}$$

Master radius update:

$$\Delta^{+} = \begin{cases} \max(\Delta, 7\|\Delta x_{\text{T}}\|_2), & \text{if } \rho \geq 0.9 \\ \max(\Delta, 2\|\Delta x_{\text{T}}\|_2), & \text{if } 0.3 \leq \rho < 0.9 \\ \Delta, & \text{if } 10^{-8} \leq \rho < 0.3 \\ \min(0.5\Delta, 0.5\|\Delta x_{\text{T}}\|_2), & \text{if } \rho < 10^{-8} \end{cases}$$

OTHER DETAILS

- currently use MINOS (simplex code) to solve LPs (Murtagh & Saunders)
- use GLTR (**GALAHAD** code) with augmented-system preconditioning to solve EQPs (G., Lucidi, Roma, Toint)
- hot-start LPs
- penalty parameter ρ update based on how good a job current ρ does in achieving "linearized feasibility" for LP
- lots of other "tricks"
- covered by more general global convergence theory (Byrd, G., Nocedal, Waltz)

HOW DOES THIS WORK IN PRACTICE?

CUTEr test set problem sizes and characteristics

Problem class	Problem size	# of problems QP	BC	GC	Total
Very Small	$1 \leq n+m < 100$	30	59	177	266
Small	$100 \leq n+m < 1000$	24	5	47	76
Medium	$1000 \leq n+m < 10000$	27	30	61	118
Large	$10000 \leq n+m$	36	13	51	100
Total	all	117	107	336	560

BC = Bound Constrained, GC = Generally Constrained

Compare with:

KNITRO – interior-point, CG-based, mature

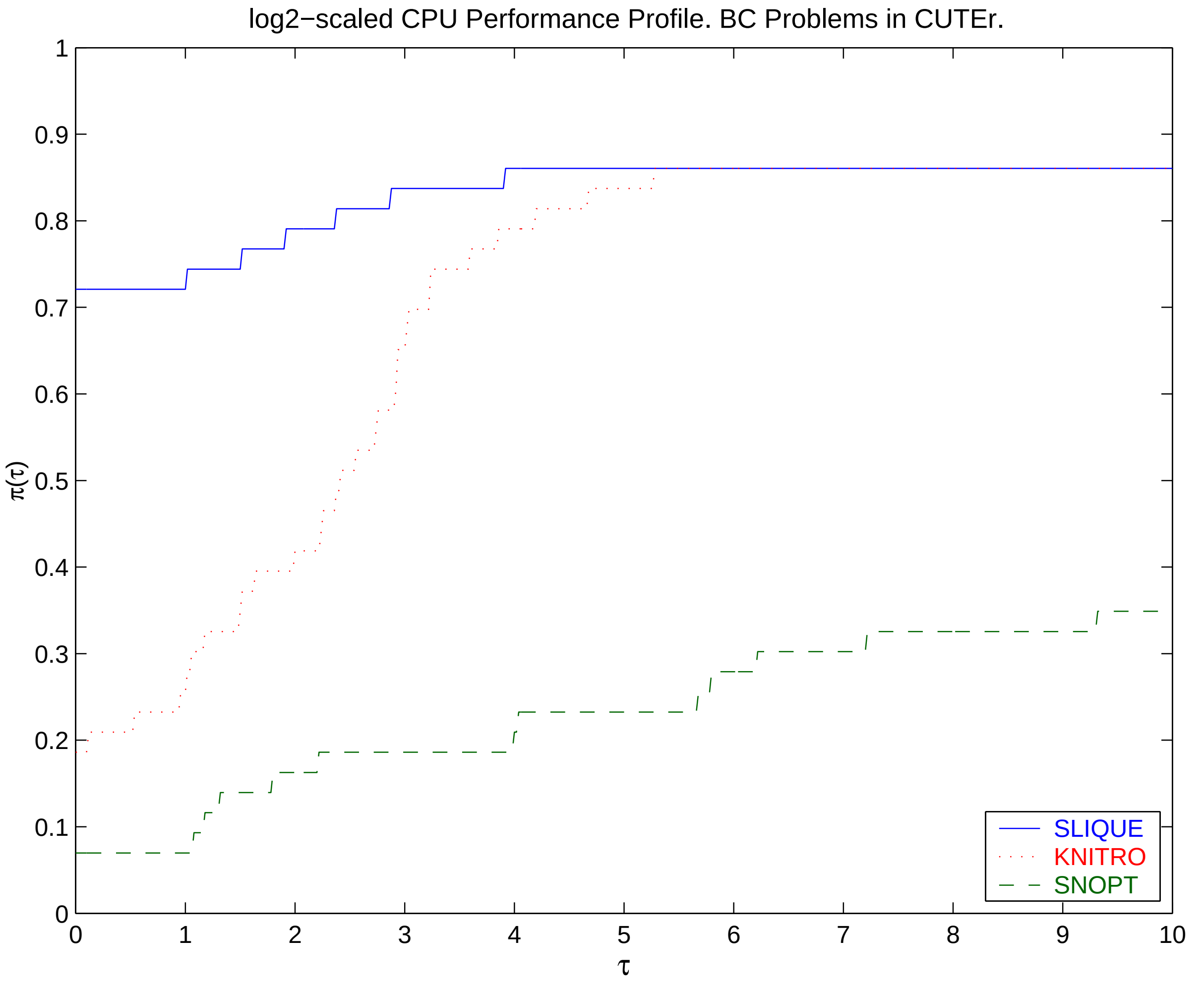
SNOPT – SQP, 1st derivative, mature

ROBUSTNESS By problem class and problem size:

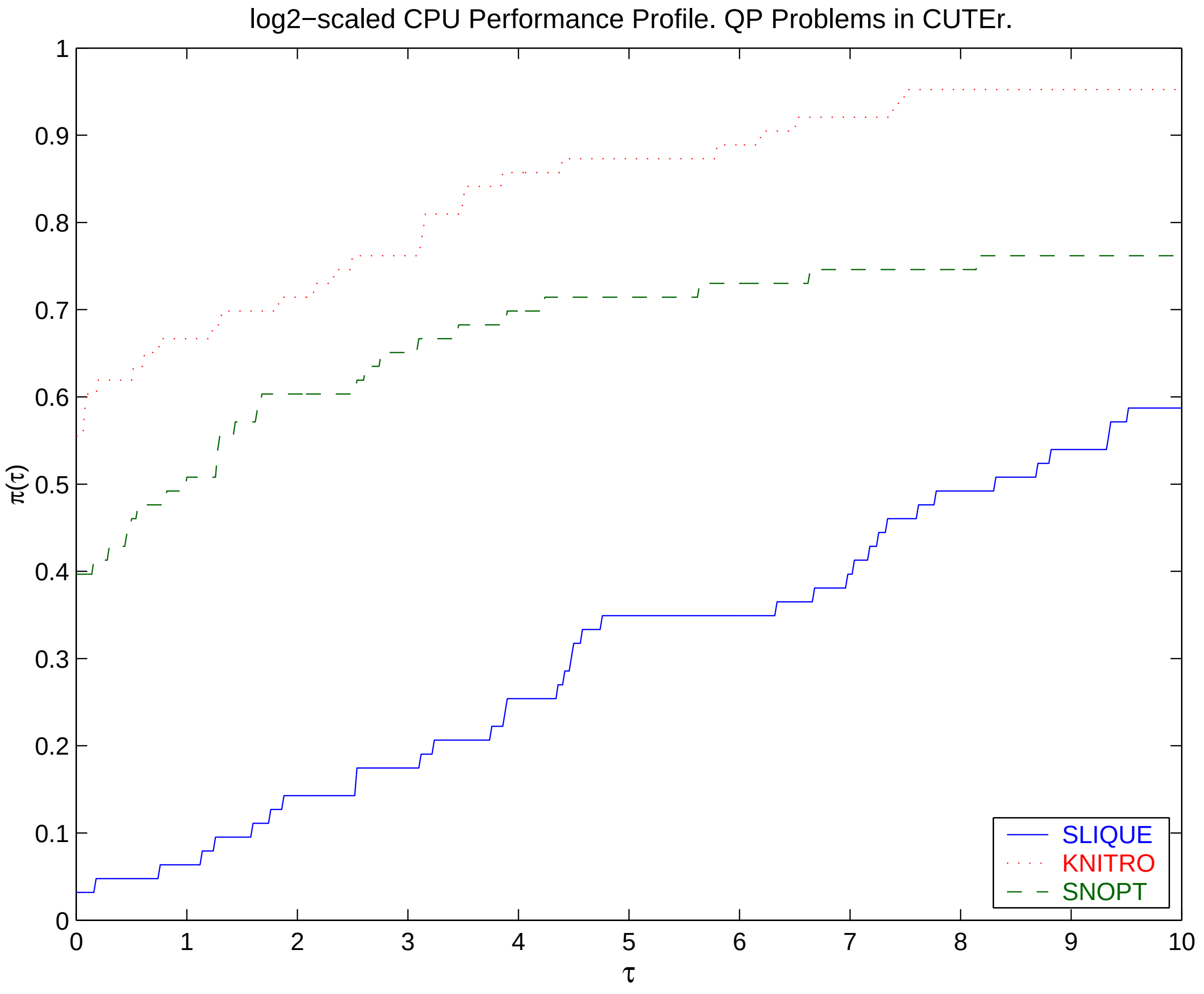
Problem class	Sample size	SLIQUE		KNITRO		SNOPT	
		# Opt	% Opt	# Opt	% Opt	# Opt	% Opt
QP	117	89	76.1	113	96.6	99	84.6
BC	107	96	89.7	98	91.6	71	66.3
GC	336	253	75.3	295	87.8	285	84.8
Total	560	438	78.2	506	90.4	455	81.3

Problem class	Sample size	SLIQUE		KNITRO		SNOPT	
		# Opt	% Opt	# Opt	% Opt	# Opt	% Opt
Very Small	266	247	92.9	251	94.4	250	94.0
Small	76	58	76.3	67	88.2	70	92.1
Medium	118	86	72.9	103	87.3	83	70.3
Large	100	47	47.0	85	85.0	52	52.0
Total	560	438	78.2	506	90.4	455	81.3

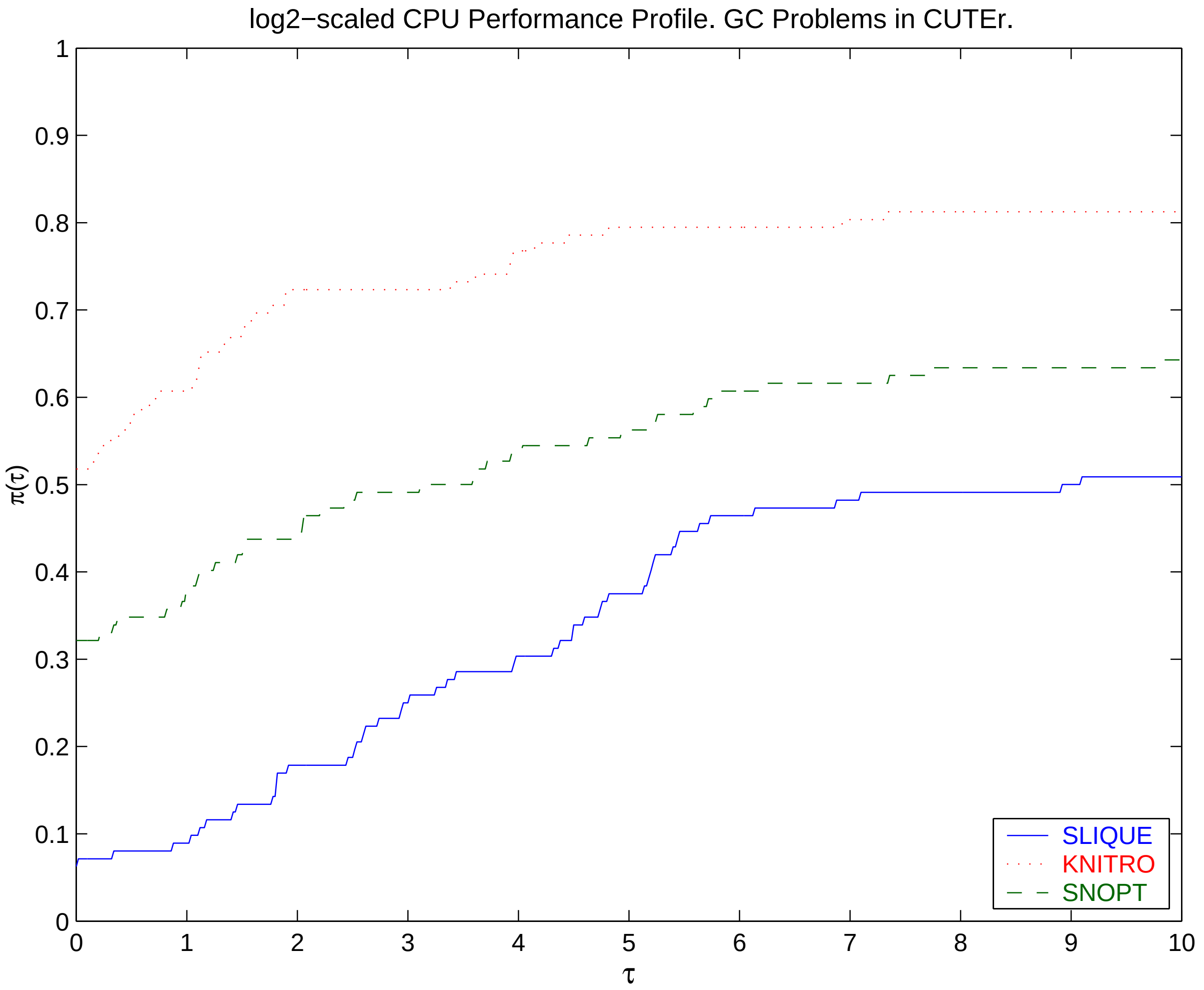
PERFORMANCE PROFILE - CPU medium/large BCs



PERFORMANCE PROFILE - CPU medium/large QPs



PERFORMANCE PROFILE - CPU medium/large GCs



AN INTERIOR-POINT ALTERNATIVE

A non-differentiable penalty-barrier method

$$\underset{x \in \mathbb{R}^n}{\textbf{minimize}} \ f(x) \ \textbf{subject to} \ c_{\mathcal{E}}(x) = 0 \ \textbf{and} \ c_{\mathcal{I}}(x) \geq 0$$

Aim to solve problem by minimizing the non-differentiable penalty f^n

$$\phi(x, \nu) = f(x) + \nu \sum_{i \in \mathcal{E}} |c_i(x)| + \nu \sum_{i \in \mathcal{I}} \max(-c_i(x), 0)$$

for some sufficiently large ν

Can reformulate this as a smooth problem:

- ⊙ replace the terms $|c_i(x)|$ and $\max(-c_i(x), 0)$ by equivalent smooth terms

NON-SMOOTH TERMS

equality constraints: write contribution $\nu|c_i(x)|$ as

$$\nu[r_i + s_i], \quad \text{where} \;\; c_i(x) = r_i - s_i \;\; \text{and} \;\; (r_i, s_i) \geq 0,$$

or alternatively as

$$\nu[c_i(x) + 2s_i], \quad \text{where} \;\; c_i(x) + s_i \geq 0 \;\; \text{and} \;\; s_i \geq 0$$

inequality constraints: write contribution $\nu \max(-c_i(x), 0)$ as

$$\nu s_i, \quad \text{where} \;\; c_i(x) = r_i - s_i \;\; \text{and} \;\; (r_i, s_i) \geq 0$$

or alternatively as

$$\nu s_i, \quad \text{where} \;\; c_i(x) + s_i \geq 0 \;\; \text{and} \;\; s_i \geq 0$$

$\Longrightarrow$

A SMOOTH REFORMULATION

Thus the minimization of ϕ may be expressed as

$$\begin{array}{ll} \underset{x,s}{\text{minimize}} & f(x) + \nu \sum_{i \in \mathcal{E}} [c_i(x) + 2s_i] + \nu \sum_{i \in \in \mathcal{I}} s_i \\ \text{subject to} & c_i(x) + s_i \geq 0 \;\text{ and }\; s_i \geq 0 \;\text{ for all }\; i \in \mathcal{E} \cup \mathcal{I} \end{array}$$

involving "surplus" variables s (G., Orban, Toint)

- can use IP methods to solve this **inequality**-constrained problem
- finding an initial interior point is trivial
- may sometimes be better to replace $\nu|c_i(x)|$ term by

  $$\nu[2r_i - c_i(x)], \quad \text{where } \; r_i - c_i(x) \geq 0 \;\text{ and }\; r_i \geq 0$$

  especially if initially $c_i(x) < 0$
- if ever $c_i(x) > 0$, can simply remove s_i

(Mayne & Polak, Tits, Wächter, Bakhtiari, Urban & Lawrence)

WHY IS THIS PROMISING?

- ⊙ general constrained problem reduced to smooth unconstrained problem simply involving barrier terms
- ⊙ linear algebra well understood for such problems
- ⊙ Newton-like subproblem easy to truncate using (e.g.) conjugate gradients
- ⊙ to improve performance, better to use primal-dual rather than primal Newton model
- ⊙ can take direct account of (for example) linear constraints & simple bounds on variables ("phase-1" procedure)
- ⊙ global and local convergence theory established

BARRIER FUNCTION AND ITS DERIVATIVES

(logarithmic) barrier function: $\Psi_{\mu,\nu}(x,s) =$

$$f(x) + \nu e_{\mathcal{E}}^T[c_{\mathcal{E}}(x) + 2s_{\mathcal{E}}] + \nu e_{\mathcal{I}}^T s_{\mathcal{I}} - \mu e^T \log(c(x) + s) - \mu e^T \log s$$

$$\nabla_v \Psi_{\mu,\nu}(x,s) = \begin{pmatrix} g(x) - J^T(x) y(x,s) \\ \nu e - y(x,s) - u(s) \end{pmatrix}$$

$$\nabla_{vv} \Psi_{\mu,\nu}(x,s) =$$

$$\begin{pmatrix} H(x, y(x,s)) + \mu J^T(x)(C(x)+S)^{-2} J(x) & \mu J^T(x)(C(x)+S)^{-2} \\ \mu (C(x)+S)^{-2} J(x) & \mu (C(x)+S)^{-2} + \mu S^{-2} \end{pmatrix}$$

where $\qquad v = (x,s)$

$$\begin{aligned} y_{\mathcal{E}}(x,s) &= \mu (C_{\mathcal{E}}(x) + S_{\mathcal{E}})^{-1} e_{\mathcal{E}} - \nu e_{\mathcal{E}} \\ y_{\mathcal{I}}(x,s) &= \mu (C_{\mathcal{I}}(x) + S_{\mathcal{I}})^{-1} e_{\mathcal{I}} \\ u(s) &= \mu S^{-1} e \\ J(x) &= \nabla_x c(x) \ \text{ and } \ H(x,y) = \nabla_{xx} f(x) - \sum_i y_i \nabla_{xx} c_i(x) \\ C(x) &= \operatorname{diag} c(x) \quad \text{(etc)} \end{aligned}$$

BASIC SEARCH DIRECTION SUBPROBLEM

Primal-dual Hessian approximation: $\nabla_{vv}\Psi^{\text{PD}}_{\mu,\nu}(x,s) =$

$$\begin{pmatrix} H(x,y^{\text{PD}}) + J^T(x)Y^{\text{PD}}(C(x)+S)^{-1}J(x) & J^T(x)Y^{\text{PD}}(C(x)+S)^{-1} \\ Y^{\text{PD}}(C(x)+S)^{-1}J(x) & Y^{\text{PD}}(C(x)+S)^{-1} + U^{\text{PD}}S^{-1} \end{pmatrix}$$

Find search direction $\Delta v = (\Delta x, \Delta s)$ to (approximately)

$$\underset{\Delta v}{\text{minimize}} \;\; \Delta v^T \nabla_v \Psi_{\mu,\nu}(x,s) + \tfrac{1}{2}\Delta v^T \nabla_{vv}\Psi^{\text{PD}}_{\mu,\nu}(x,s)\Delta v \;\; \text{s.t.} \;\; \|\Delta v\|_B \le \Delta$$

- ⊙ B positive-definite approximation of $\nabla_{vv}\Psi^{\text{PD}}_{\mu,\nu}(x,s)$
- ⊙ B replaces $H(x,y^{\text{PD}})$ by suitable P, e.g.
  - ⋄ $P = 0$
  - ⋄ $P = I$
  - ⋄ $P = H(x,y^{\text{PD}})$ (!!)

PRECONDITIONING CONJUGATE GRADIENTS

$$\underset{\Delta v}{\textbf{minimize}} \;\; \Delta v^T \nabla_v \Psi_{\mu,\nu}(x,s) + \tfrac{1}{2}\Delta v^T \nabla_{vv} \Psi^{\mathrm{PD}}_{\mu,\nu}(x,s)\Delta v \;\; \textbf{s.t.} \;\; \|\Delta v\|_B \leq \Delta$$

Use preconditioned conjugate gradients — basic preconditioning step

$$\begin{pmatrix} P + J^T Y (C+S)^{-1} J & J^T Y (C+S)^{-1} \\ Y (C+S)^{-1} J & Y (C+S)^{-1} + U S^{-1} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \end{pmatrix} = \begin{pmatrix} r_x \\ r_s \end{pmatrix}$$

Possibly too dense $\Longrightarrow$ define $w = Y(C+S)^{-1}(J\Delta x + \Delta s) \Longrightarrow$

$$\begin{pmatrix} P & 0 & J^T \\ 0 & U S^{-1} & I \\ J & I & -Y^{-1}(C+S) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \\ w \end{pmatrix} = \begin{pmatrix} r_x \\ r_s \\ 0 \end{pmatrix}$$

P suitable $\Longleftrightarrow$ above matrix has precisely rank J -ve eigenvalues

OUTSTANDING ISSUES

- penalty parameter updates
- which side should we penalize equality constraints?
- is it better to remove surplus variables s_i as soon as possible?
- is primal-dual Hessian better “globally”?
- choice of P in preconditioner?

CONCLUSIONS

- SQP methods may be too expensive in general
- cheaper alternatives using LP & unconstrained minimization subproblems worth pursuing
- SLIQUE "promising" but needs improvements — IP LP??
- non-differentiable penalty-barrier method — SUPERB(?) — under development
- structure & decomposition likely crucial to make further progress
- there is already very good, publicly available software for solving linear & nonlinear optimization problems