An interior-point ℓ_1-penalty method for nonlinear optimization

Nick Gould (RAL)

joint with

Dominique Orban (Montreal)
Philippe Toint (Namur)

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

Workshop on Large Scale Nonlinear and Semidefinite Programming
University of Waterloo
13th May 2004

Summary of the talk

- Nonlinear Programming, GALAHAD and the pitfalls of SQP
- Interior-point methods for inequalities
- Non-differentiable penalty functions
- A smooth reformulation
- An appropriate barrier function and its derivatives
- Algorithmic details
- Comments and conclusions

Nonlinear Programming

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

- 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

GALAHAD 1.0 (April 2002), concentrated on

- QP methods + LANCELOT B — improved LANCELOT A

GALAHAD 2.0 (2005?) planned

- SQP successors to LANCELOT B

But . . .

SQP — drawbacks

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

- SQP step computation is too coarse/expensive a calculation, especially in early iterations $\Longrightarrow$ **inefficiency**
- would like to use exact 2nd derivatives, but ... 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**

Remedies and alternatives

One possibility is to impose “descent” as an extra constraint on the SQP subproblem, i.e., require slope of merit function along step < 0

Otherwise, use another approach

- augmented Lagrangian approach
  - too slow
  - poor at handling linear constraints
  - adversely affected by degeneracy
- our preferences
  - sequential linear/quadratic programming approach

    (Fletcher & Sainz de la Maza, Byrd, G., Nocedal & Waltz)

    $\Longrightarrow$ **Jorge Nocedal’s talk**
  - interior-point approach

Interior-point approaches for inequality constraints

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

Aim to solve problem by approximately minimizing a sequence of **barrier** functions

$$\psi(x, \mu_k) = f(x) - \mu_k \sum_{i \in \mathcal{I}} \log c_i(x)$$

as $\{\mu_k\} \longrightarrow 0_+$. (Fiacco & McCormick)

- strong global and local convergence results
- excellent practical behaviour (primal-dual variants)

Interior-point issues

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) - \mu \sum_{i \in \mathcal{I}} \log c_i(x)$$

- need an initial "interior point" $x_0 \mid c_{\mathcal{I}}(x_0) > 0$
- how do we handle **equality** constraints?
- how do we pick $\{\mu_k\}$?
- how do we cope with large problems?

The non-differentiable penalty function

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

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 **penalty parameter** 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

$$\underset{x,s}{\text{minimize}} \quad f(x) + \nu \sum_{i \in \mathcal{E}} [c_i(x) + 2s_i] + \nu \sum_{i \in \mathcal{I}} s_i$$

$$\text{subject to} \quad c_i(x) + s_i \geq 0 \quad \text{and} \quad s_i \geq 0 \quad \text{for all} \quad i \in \mathcal{E} \cup \mathcal{I}$$

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

- can use IP methods to solve this **inequality**-constrained problem
- finding an initial interior point is trivial
- if ever $c_i(x) > 0$, can simply remove s_i

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

Alternatives

May **sometimes be better** to replace $\nu|c_i(x)|$ term by

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

especially if initially $c_i(x) < 0 \implies$

$$\underset{x,s}{\text{minimize}} \quad f(x) + \nu \sum_{i \in \mathcal{E}} [2s_i - c_i(x)] + \nu \sum_{i \in \mathcal{I}} [s_i - c_i(x)]$$

$$\text{subject to} \quad s_i - c_i(x) \geq 0 \quad \text{and} \quad s_i \geq 0 \quad \text{for all} \quad i \in \mathcal{E} \cup \mathcal{I}$$

involving **elastic** "slack" variables s …or

$$\underset{x,s}{\text{minimize}} \quad f(x) + 2\nu \sum_{i \in \mathcal{E}} s_i + \nu \sum_{i \in \mathcal{I}} s_i$$

$$\text{subject to} \quad -s_i \leq c_i(x) \leq s_i \quad \text{for all} \quad i \in \mathcal{E},$$

$$\text{and} \quad c_i(x) + s_i \geq 0 \quad \text{and} \quad s_i \geq 0 \quad \text{for all} \quad i \in \mathcal{I}$$

Why is this promising?

- general constrained problem reduced to smooth unconstrained problem simply involving barrier terms $\Longrightarrow$ solve large problems(?)
- automatically satisfies (Mangasarian–Fromowitz) constraint qualification $\iff$ bounded Lagrange multipliers
- 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

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

Aim to approximately minimize the (logarithmic) **barrier function**:

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

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

using a trust-region method while maintaining $c(x) + s > 0$ and $s > 0$

- need to (approximately) minimize suitable second-order model within a trust-region
- model and trust region based on derivatives of $\Psi_{\mu,\nu}(x,s)$

First derivatives

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

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

where

$$\begin{aligned} v &= (x,s) \\ y_{\mathcal{E}}(x,s) &= \mu(C_{\mathcal{E}}(x) + S_{\mathcal{E}})^{-1} e_{\mathcal{E}} - \nu e_{\mathcal{E}} \\ y_{\mathcal{I}}(x,s) &= \mu(C_{\mathcal{I}}(x) + S_{\mathcal{I}})^{-1} e_{\mathcal{I}} \\ u(s) &= \mu S^{-1} e \\ e &= \text{vector of ones} \\ J(x) &= \nabla_x c(x) \;\; \text{and} \\ C(x) &= \text{diag } c(x) \quad \text{(etc)} \end{aligned}$$

Implications from first derivatives

Aim that

$$\begin{pmatrix} g(x) - J^T(x)y(x,s) \\ \nu e - y(x,s) - u(s) \end{pmatrix} = 0$$

where

$$\begin{aligned} y_{\mathcal{E}}(x,s) &= \mu(C_{\mathcal{E}}(x) + S_{\mathcal{E}})^{-1} e_{\mathcal{E}} - \nu e_{\mathcal{E}} > -\nu e_{\mathcal{E}} \\ y_{\mathcal{I}}(x,s) &= \mu(C_{\mathcal{I}}(x) + S_{\mathcal{I}})^{-1} e_{\mathcal{I}} > 0 \\ u(s) &= \mu S^{-1} e > 0 \end{aligned}$$

$g(x) - J^T(x)y(x,s) = 0 \Longrightarrow$ **gradient of Lagrangian wrt x vanishes**

$\nu e - y(x,s) - u(s) = 0 \Longrightarrow$

$$0 < y_{\mathcal{E}}(x,s) + \nu e_{\mathcal{E}}, u_{\mathcal{E}}(s) < 2\nu e_{\mathcal{E}} \;\text{ and }\; 0 < y_{\mathcal{I}}(x,s), u_{\mathcal{I}}(s) < \nu e_{\mathcal{E}}$$

$\Longrightarrow$ **Lagrange multiplier (estimates) bounded**, cf. MFCQ

Second derivatives

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

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

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

Basic search direction subproblem

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

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

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

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

- y^{PD} and u^{PD} are primal-dual multiplier estimates
- 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

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

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

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

Eliminate $\Delta s \Longrightarrow$

Preconditioning (continued)

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

and then recover Δs from

$$\Delta s = -U^{-1}Sw + U^{-1}Sr_s$$

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

Modifications to usual Trust-region algorithm

- may need to backtrack along $(x + \alpha\Delta x, s + \alpha\Delta s)$ to ensure feasibility
- can use linesearch technique within trust-region method

(Yuan & Nocedal)

Convergence results

Under standard assumptions:

- Provided care is taken over allowable primal-dual Lagrange multiplier estimates, TR iterates converge to stationary point of $\Psi_{\mu_k,\nu_k}(x,s)$ for fixed μ_k and $\nu_k \implies$ allows for early termination
- Provided care is taken with inner-iteration termination rules, μ_k is reduced gradually to zero, and ν_k is only increased when infeasibility appears to stagnate, approximate stationary points of $\Psi_{\mu_k,\nu_k}(x,s)$ converge to either KKT points for original problem (ν_k bounded) or stationary points of infeasibility ($\nu_k \to \infty$)
- Provided μ_k decreased superlinearly to zero, iterates converge (componentwise) superlinearly for bounded ν_k

(G., Orban, Sartenaer, Toint)

Improvements (I) — magical steps

Having computed Δx, instead choose correction $s(x+\Delta x)$ to

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

over $s \geq \max(0, -c(x))$

$\Longrightarrow$ $|\mathcal{E} \cup \mathcal{I}|$ independent one-dimensional minimizations

- often known as a **magical step**
- easy or even trivial to find each elastic $s_i(x+\Delta x)$
- prefer instead the Newton path $(x+\alpha\Delta x, s(x+\alpha\Delta x))$
- very helpful in practice

Improvements (II) — implicit elastics

As before, define $s(x)$ to

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

over $s \geq \max(0, -c(x))$

$\Longrightarrow$ $|\mathcal{E} \cup \mathcal{I}|$ easy or even independent one-dimensional minimizations

Then aim to approximately minimize the (logarithmic) **implicit-elastic barrier function**:

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

using a trust-region method

Implicit elastics (continued)

- differentiability inherited from original problem
- gives the **same** Newton correction Δx as with magical steps
- trades a reduction in problem size against a slight increase in nonlinearity
- linear algebra actually very similar
- tends to be better at keeping away from the constraint boundary
- highly valuable in practice

Improvements (III) — Upper bound on s

Might want to include upper bounds $s \leq s_u$

- bounds should permit at least initial $s_0 \leq s_u$
- helps prevent unbounded barrier problems
- need to introduce additional barrier term to cope
- may need to reject trial points if $c(x + \alpha\Delta x) + s_u \leq 0$
- very helpful in practice

Outstanding issues

- penalty parameter updates
- which side should we penalize equality constraints?
  or should we penalize on both sides ?
- is it better to remove elastic variables s_i as soon as possible?
- is primal-dual Hessian better “globally”?
- choice of P in preconditioner?
- is degeneracy harmful? — we believe not
- shift and scale variables and functions at initial point
  (and possibly thereafter)?

Numerical results

Developing GALAHAD package: the

Sequential

Unconstrained minimization of an ℓ_p

Penalty function treating

Equality and inequality

Restrictions by

Barrier terms

- can now solve “all” Hock & Schittkowski problems with default solver values
- behaves broadly similarly to GALAHAD IP-QP solver on large QPs
- so far, not robust as we would like for general NLPs … currently working on this

Conclusions

- reformulation of equality constraints to allow interior-point solution
- reformulation “regularized”
- “simple” algorithm using well-studied linear algebra
- potentially well suited for large-scale case
- global and local convergence theory established
- GALAHAD code SUPERB(?) under development
- “promising” but premature for detailed results

RAL TR-2003-022