Advances in SLQP algorithms for large-scale nonlinear optimization

Richard A. Waltz, Northwestern University

rwaltz@ece.northwestern.edu

joint with: Richard Byrd, Nick Gould and Jorge Nocedal

40th Workshop on Large Scale Nonlinear Optimization

Erice, Sicily, Italy

23 June, 2004

Motivation

Why develop a new active-set code?

- Active-set methods have nice properties
  - warm starts
  - good active-set and sensitivity info
  - more stable
- SQP ineffective when large reduced space

GOAL: Develop large-scale active-set method

Weaknesses of SQP

- Costly when many degrees of freedom
  - Form and factorize a (dense) reduced Hessian
- Infeasible QP subproblems
- Tradeoff: Exact 2nd derivatives vs. indefinite QPs
  - TR: Exact Hessian but indefinite QPs (FilterSQP)
  - LS: BFGS Hessian but convex QPs (SNOPT)
- Can’t get a cheap Cauchy point for SQP

SLQP Methods

Investigating SLQP methods:

- Estimate active-set at each iteration via LP solve

SLQP Methods

Investigating SLQP methods:

- Estimate active-set at each iteration via LP solve
- Only solve equality constrained QPs

SLQP Methods

Investigating SLQP methods:

- Estimate active-set at each iteration via LP solve
- Only solve equality constrained QPs
- Can use iterative (projected CG) approach

SLQP Methods

Investigating SLQP methods:

- Estimate active-set at each iteration via LP solve
- Only solve equality constrained QPs
- Can use iterative (projected CG) approach
- Can solve EQP subproblems inexactly

SLQP Methods

Investigating SLQP methods:

- Estimate active-set at each iteration via LP solve
- Only solve equality constrained QPs
- Can use iterative (projected CG) approach
- Can solve EQP subproblems inexactly
- No need to factorize Hessian/reduced Hessian

SLQP Methods

Investigating SLQP methods:

- Estimate active-set at each iteration via LP solve
- Only solve equality constrained QPs
- Can use iterative (projected CG) approach
- Can solve EQP subproblems inexactly
- No need to factorize Hessian/reduced Hessian
- Can work with exact Hessians and handle nonconvexities

SLQP Methods

Investigating SLQP methods:

- Estimate active-set at each iteration via LP solve
- Only solve equality constrained QPs
- Can use iterative (projected CG) approach
- Can solve EQP subproblems inexactly
- No need to factorize Hessian/reduced Hessian
- Can work with exact Hessians and handle nonconvexities

BIG QUESTION:

How well does LP estimate the active-set???

Overview of SLQP

1. Solve LP to estimate active-set
2. Step given by LP solution determines Cauchy point
3. Solve equality constrained QP (use CG)
4. Trial step is a convex combination of LP and EQP steps (achieves at least as much decrease as Cauchy step).

- Fletcher, Sainz de la Maza (1989)
- Chin, Fletcher (1999,2003)
- Byrd, Gould, Nocedal, W. [SLIQUE] (2004)

Overview of SLIQUE

Nonlinear Problem:

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

Assume all functions twice-continuously differentiable.

LP Subproblem (SLIQUE)

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

LP Subproblem (SLIQUE)

$$\min_{d} \quad \nabla f(x)^T d$$

$$\text{s.t.} \quad h_i(x) + \nabla h_i(x)^T d = 0, \quad i \in \mathcal{E}$$

$$g_i(x) + \nabla g_i(x)^T d \geq 0, \quad i \in \mathcal{I}$$

$$\|d\|_\infty \leq \Delta^{\mathsf{LP}}$$

LP Subproblem (SLIQUE)

$$\min_{d} \quad \nabla f(x)^T d$$

$$\text{s.t.} \quad h_i(x) + \nabla h_i(x)^T d = 0, \quad i \in \mathcal{E}$$

$$g_i(x) + \nabla g_i(x)^T d \geq 0, \quad i \in \mathcal{I}$$

$$\|d\|_\infty \leq \Delta^{\text{LP}}$$

Constraints may be inconsistent!

ℓ_1 LP Formulation (SLIQUE)

$$\begin{aligned}\ell(d) &= \nabla f(x)^T d + \nu \sum_{i \in \mathcal{E}} |h_i(x) + \nabla h_i(x)^T d| \\ &\quad + \nu \sum_{i \in \mathcal{I}} \max(0, -g_i(x) - \nabla g_i(x)^T d)\end{aligned}$$

ℓ_1 LP Formulation (SLIQUE)

$$\ell(d) = \nabla f(x)^T d + \nu \sum_{i \in \mathcal{E}} |h_i(x) + \nabla h_i(x)^T d| + \nu \sum_{i \in \mathcal{I}} \max(0, -g_i(x) - \nabla g_i(x)^T d)$$

$$\min_d \quad \ell(d)$$
$$\text{s.t.} \quad \|d\|_\infty \le \Delta^{\text{LP}}$$

ℓ_1 LP Formulation (SLIQUE)

$$\ell(d) = \nabla f(x)^T d + \nu \sum_{i \in \mathcal{E}} |h_i(x) + \nabla h_i(x)^T d| + \nu \sum_{i \in \mathcal{I}} \max(0, -g_i(x) - \nabla g_i(x)^T d)$$

$$\begin{array}{ll} \min\limits_{d} & \ell(d) \\ \text{s.t.} & \|d\|_\infty \le \Delta^{\text{LP}} \end{array}$$

$$\mathcal{W}(x) = \{i \in \mathcal{E} \mid h_i(x) + \nabla h_i(x)^T d^{\text{LP}} = 0\} \cup \{i \in \mathcal{I} \mid g_i(x) + \nabla g_i(x)^T d^{\text{LP}} = 0\}$$

Model Function and Cauchy Point

Minimize quadratic model along LP direction

$$q(d) = \ell(d) + \frac{1}{2} d^T H(x, \lambda) d,$$

Model Function and Cauchy Point

Minimize quadratic model along LP direction

$$q(d) = \ell(d) + \frac{1}{2} d^T H(x, \lambda) d,$$

$$\min_{\alpha} q(\alpha d^{\mathsf{LP}}), \quad \alpha \in (0, 1]$$

Model Function and Cauchy Point

Minimize quadratic model along LP direction

$$q(d) = \ell(d) + \tfrac{1}{2} d^T H(x,\lambda) d,$$

$$\min_{\alpha} q(\alpha d^{\mathsf{LP}}), \qquad \alpha \in (0,1]$$

Global convergence theory based on Cauchy step.

- Byrd, Gould, Nocedal, W. (2004)

EQP Subproblem (SLIQUE)

$$\begin{aligned} \min_{d} \quad & \tfrac{1}{2} d^T H(x,\lambda) d + \nabla f(x)^T d \\ \text{s.t.} \quad & h_i(x) + \nabla h_i(x)^T d = 0, \quad i \in \mathcal{E} \cap \mathcal{W} \\ & g_i(x) + \nabla g_i(x)^T d = 0, \quad i \in \mathcal{I} \cap \mathcal{W} \\ & \|d\|_2 \le \Delta \end{aligned}$$

- Solve using projected CG

EQP Subproblem (SLIQUE)

$$\begin{aligned} \min_{d} \quad & \tfrac{1}{2} d^T H(x,\lambda) d + \nabla f(x)^T d \\ \text{s.t.} \quad & h_i(x) + \nabla h_i(x)^T d = 0, \quad i \in \mathcal{E} \cap \mathcal{W} \\ & g_i(x) + \nabla g_i(x)^T d = 0, \quad i \in \mathcal{I} \cap \mathcal{W} \\ & \|d\|_2 \le \Delta \end{aligned}$$

- Solve using projected CG
- NOTE: Inactive constraints completely ignored!

EQP Subproblem (SLIQUE)

$$\begin{aligned} \min_{d} \quad & \tfrac{1}{2} d^T H(x,\lambda) d + \nabla f(x)^T d \\ \text{s.t.} \quad & h_i(x) + \nabla h_i(x)^T d = 0, \quad i \in \mathcal{E} \cap \mathcal{W} \\ & g_i(x) + \nabla g_i(x)^T d = 0, \quad i \in \mathcal{I} \cap \mathcal{W} \\ & \|d\|_2 \le \Delta \end{aligned}$$

- Solve using projected CG
- NOTE: Inactive constraints completely ignored!
- Redefine the EQP Hessian and gradient to include info on inactive, violated constraints

Redefined EQP Hessian and gradient

Redefine EQP Hessian H^{EQP} and gradient g^{EQP} to take into account violated constraints

$\mathcal{V}$: set of violated constraints at EQP origin
$\mathcal{V}^c$: complement of $\mathcal{V}$

$$
\begin{aligned}
H^{\text{EQP}}(x,\lambda) = \quad & \nabla^2 f(x) \\
& - \sum_{i \in \mathcal{V}^c \cap \mathcal{E}} \lambda_i \nabla^2 h_i(x) - \sum_{i \in \mathcal{V}^c \cap \mathcal{I}} \lambda_i \nabla^2 g_i(x) \\
& + \nu \sum_{i \in \mathcal{V} \cap \mathcal{E}} \text{sign}() \nabla^2 h_i(x) - \nu \sum_{i \in \mathcal{V} \cap \mathcal{I}} \nabla^2 g_i(x)
\end{aligned}
$$

EQP gradient redefined in a similar way

Total Step

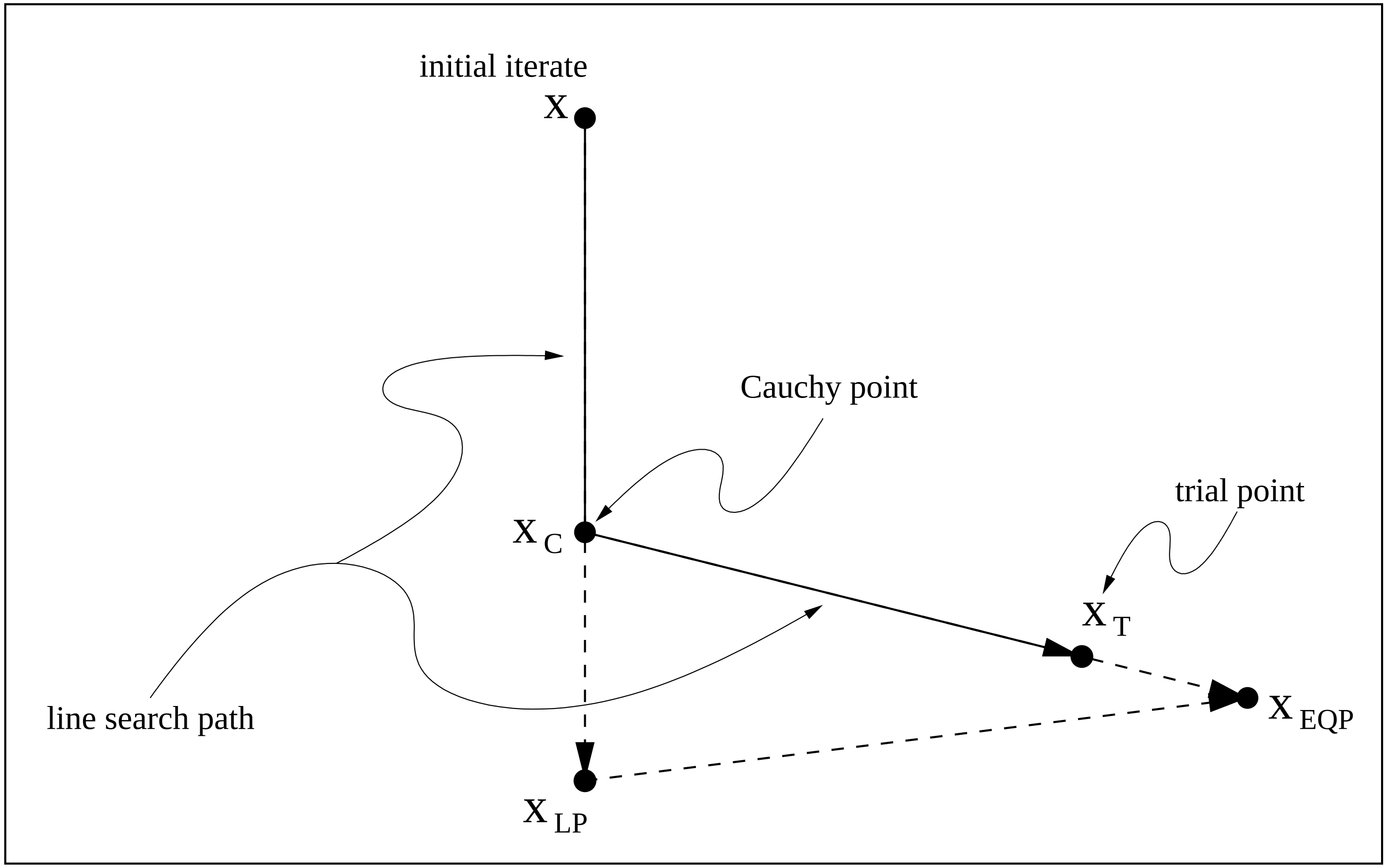


Figure 0: Dogleg path for step computation.

Difficulties

- LP inefficiencies
- Managing 2 trust-regions
- Handling degeneracy
- How to handle the penalty parameter?
- ...

LP Inefficencies

LP Ineffciencies

LP subproblem (in standard LP form):

$$\begin{aligned}
\min_{d,q,r,s,t} \quad & \nabla f(x)^T d + \nu \sum_{i \in \mathcal{E}} (q_i + r_i) + \nu \sum_{i \in \mathcal{I}} t_i \\
\text{s.t} \quad & h_i(x) + \nabla h_i(x)^T d = q_i - r_i, \quad i \in \mathcal{E} \\
& g_i(x) + \nabla g_i(x)^T d = s_i - t_i, \quad i \in \mathcal{I} \\
& \|d\|_\infty \leq \Delta^{\text{LP}} \\
& q, r, s, t, \geq 0
\end{aligned}$$

LP Ineffcicencies

LP subproblem (in standard LP form):

$$\begin{aligned}
\min_{d,q,r,s,t} \quad & \nabla f(x)^T d + \nu \sum_{i\in\mathcal{E}} (q_i + r_i) + \nu \sum_{i\in\mathcal{I}} t_i \\
\text{s.t} \quad & h_i(x) + \nabla h_i(x)^T d = q_i - r_i, \quad i \in \mathcal{E} \\
& g_i(x) + \nabla g_i(x)^T d = s_i - t_i, \quad i \in \mathcal{I} \\
& \|d\|_\infty \le \Delta^{\text{LP}} \\
& q, r, s, t, \ge 0
\end{aligned}$$

1. Many LP TR constraints active at solution
2. Problem constraints stabilize, but TR constraints don't!

LP Inefficencies

Attempts to reduce the LP cost:

- Give better initial basis to LP - improve warm starts

LP Inefficencies

Attempts to reduce the LP cost:

- Give better initial basis to LP - improve warm starts
  1. Apply grad. proj. to estimate active TR constraints

LP Inefficencies

Attempts to reduce the LP cost:

- Give better initial basis to LP - improve warm starts
  1. Apply grad. proj. to estimate active TR constraints
  2. Solve LP with initial basis given by:
     - active problem constraints from previous LP
     - active TR constraints from gradient projection

LP Ineffcencies

Attempts to reduce the LP cost:

- Give better initial basis to LP - improve warm starts
  1. Apply grad. proj. to estimate active TR constraints
  2. Solve LP with initial basis given by:
     - active problem constraints from previous LP
     - active TR constraints from gradient projection
- Truncate LP (stop with inexact solution)

LP Inefficencies

Attempts to reduce the LP cost:

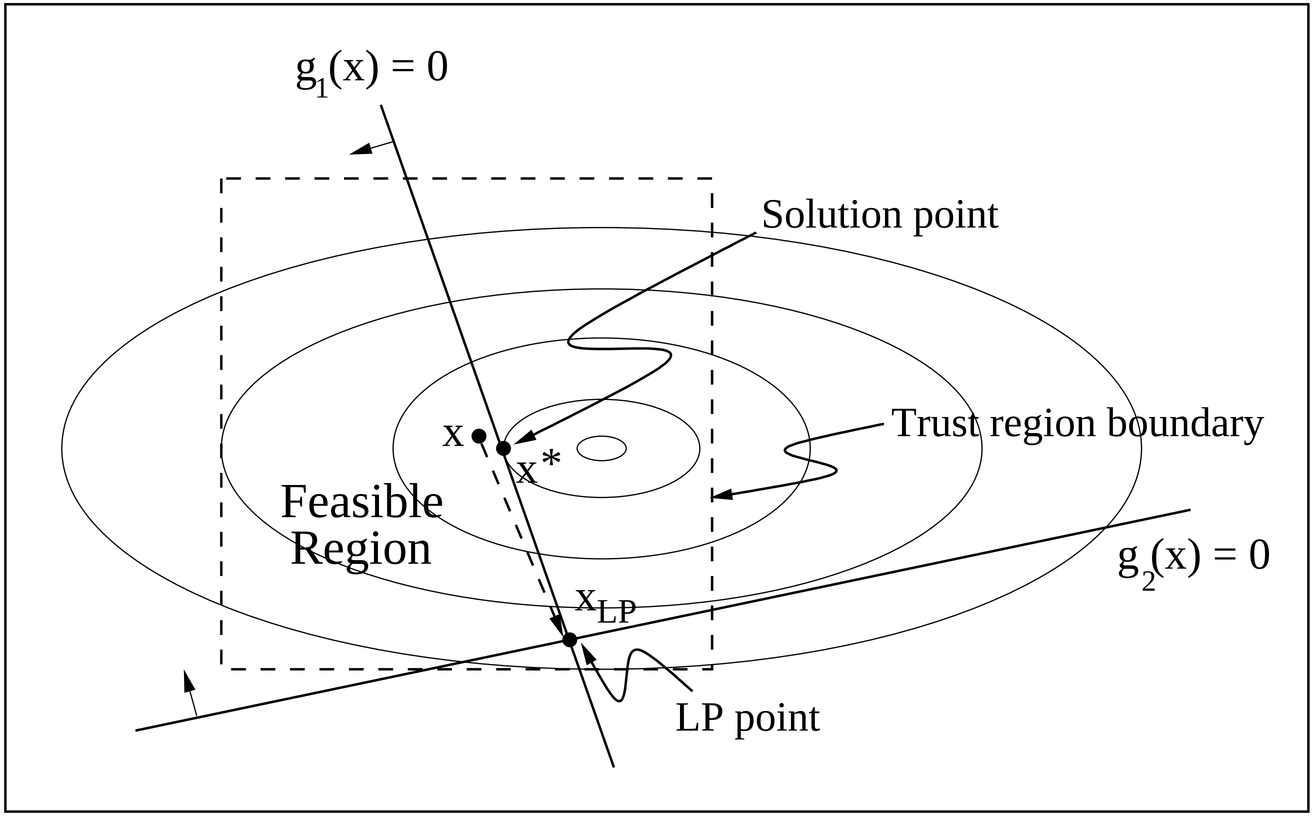
- Give better initial basis to LP - improve warm starts
  1. Apply grad. proj. to estimate active TR constraints
  2. Solve LP with initial basis given by:
     - active problem constraints from previous LP
     - active TR constraints from gradient projection
- Truncate LP (stop with inexact solution)
- Skip LP (when problem constraints stabilize)
  - When to skip?
  - How to recover if active-set is wrong?

Managing 2 Trust-Regions

Managing 2 Trust-Regions

How to manage 2 trust-regions?

- LP: box trust-region must not get “too big” near solution
- EQP: spherical trust-region (standard)



Managing 2 Trust-Regions

LP and EQP trust-regions operate quasi-independently

LP: focuses on active-set identification

Managing 2 Trust-Regions

LP and EQP trust-regions operate quasi-independently

LP: focuses on active-set identification

- accepted step: $\Delta_{k+1}^{\mathsf{LP}} = \max(1.2\|d_k\|_\infty, 1.2\|d_k^{\mathsf{C}}\|_\infty)$

Managing 2 Trust-Regions

LP and EQP trust-regions operate quasi-independently

LP: focuses on active-set identification

- accepted step: $\Delta_{k+1}^{\mathsf{LP}} = \max(1.2\|d_k\|_\infty, 1.2\|d_k^{\mathsf{C}}\|_\infty)$
- rejected step: $\Delta_{k+1}^{\mathsf{LP}} = \min(0.5\|d_k\|_\infty, \Delta_k^{\mathsf{LP}})$

Managing 2 Trust-Regions

LP and EQP trust-regions operate quasi-independently

LP: focuses on active-set identification

- accepted step: $\Delta_{k+1}^{\mathsf{LP}} = \max(1.2\|d_k\|_\infty, 1.2\|d_k^{\mathsf{C}}\|_\infty)$
- rejected step: $\Delta_{k+1}^{\mathsf{LP}} = \min(0.5\|d_k\|_\infty, \Delta_k^{\mathsf{LP}})$

EQP: measures quality of our model (standard TR)

- use standard TR update rules

Degeneracy

Degeneracy

LP subproblem (in standard LP form):

$$\begin{aligned}
\min_{d,q,r,s,t} \quad & \nabla f(x)^T d + \nu \sum_{i \in \mathcal{E}} (q_i + r_i) + \nu \sum_{i \in \mathcal{I}} t_i \\
\text{s.t} \quad & h_i(x) + \nabla h_i(x)^T d = q_i - r_i, \quad i \in \mathcal{E} \\
& g_i(x) + \nabla g_i(x)^T d = s_i - t_i, \quad i \in \mathcal{I} \\
& \|d\|_\infty \le \Delta^{\text{LP}} \\
& q, r, s, t, \ge 0
\end{aligned}$$

Want maximal, linearly independent working set of problem constraints in degenerate case.

Penalty Parameter Update

Penalty Parameter Update

SLIQUE is a penalty method

Penalty parameter plays a key role in:

1. LP subproblem (active-set identification)
2. Step computation
3. Model function
4. Merit function and step acceptance

Key idea:

Use LP subproblem as a mechanism for choosing ν.

Penalty Parameter Update

LP subproblem:

$$\min_{d} \quad \nabla f(x)^T d + \nu \sum_{i \in \mathcal{E}} |h_i(x) + \nabla h_i(x)^T d|$$

$$+\nu \sum_{i \in \mathcal{I}} \max(0, -g_i(x) - \nabla g_i(x)^T d)$$

$$\text{s.t} \quad \|d\|_\infty \le \Delta^{\text{LP}}$$

Choose ν such that:

1. Enforce sufficient reduction in linearized infeasibility (satisfy linearized constraints if possible)
2. Enforce reduction in LP objective a fraction of reduction in linearized infeasibility

Global Theory

Global Theory

For fixed penalty parameter ν:

Theorem 1

If the sequence x_k is bounded, there exists a limit point x^* which is a critical point for the ℓ_1 penalty function

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

Global Theory

Using an idealized SLIQUE-like penalty update rule:

Theorem 2

If the sequence x_k is bounded, and ν_k is bounded, there exists a limit point x^* which is either

1. a KKT point of the NLP, or
2. an infeasible critical point of $w(x) = \|h(x)\| + \|\text{max}(0, -g_i(x))\|$

Global Theory

Using an idealized SLIQUE-like penalty update rule:

Theorem 2

If the sequence x_k is bounded, and ν_k is bounded, there exists a limit point x^* which is either

1. a KKT point of the NLP, or
2. an infeasible critical point of $w(x) = \|h(x)\| + \|\max(0, -g_i(x))\|$

If $\nu_k \to \infty$,

1. if $w(x) \nrightarrow 0$, then $\exists$ a limit pt. which is an infeas. stationary pt.
2. if $w(x) \to 0$, then either
   (a) $\exists$ a limit point which satisfi es the KKT conditions; or
   (b) $\exists$ a feasible limit point where MFCQ is violated

Results Summary (Robustness)

CUTEr problems with inequality constraints/bounds

Problem class	Sample size	KNITRO (IP) % optimal	SLIQUE (SLQP) % optimal	SNOPT (SQP) % optimal
$1 \leq n+m < 1000$	344	89.0	88.1	92.1
$1000 \leq n+m < 10000$	125	81.6	73.6	70.4
$10000 \leq n+m$	147	76.9	68.7	53.7
Total	616	84.6	80.5	79.2
DOF > 2000	171	82.5	78.9	53.8

Table 0: Robustness results by problem size

References

- Byrd, Gould, Nocedal, Waltz, *An Algorithm for Nonlinear Programming Using Linear Programming and Equality Constrained Subproblems*, Report OTC 4/2002, Optimization Technology Center, Northwestern University, Evanston, IL 60208, USA. To appear in Mathematical Programming B, Volume 100, Number 1, May 2004.
- Byrd, Gould, Nocedal, Waltz, *On the Convergence of Successive Linear Programming Algorithms* Report OTC 5/2002, Optimization Technology Center, Northwestern University, Evanston, IL 60208, USA.
- Waltz, *Algorithms for Large-Scale Nonlinear Optimization*, PhD Dissertation, Northwestern University, Evanston, IL 60208, USA, June 2002.

`http://www.ece.northwestern.edu/~rwaltz/articles.html`