Survey of NLP Algorithms

L. T. Biegler
Chemical Engineering Department
Carnegie Mellon University
Pittsburgh, PA

NLP Algorithms - Outline

- Problem and Goals
- KKT Conditions and Variable Classification
- Handling Basic Variables
- Handling Nonbasic Variables
- Handling Superbasic Variables
- Ensuring Global Convergence
- Algorithms and Software
- Algorithmic Comparison
- Summary and Future Work

Constrained Optimization (Nonlinear Programming)

Problem: $Min_x\ f(x)$

s.t. $g(x) \leq 0$

$h(x) = 0$

where:

$f(x)$ - scalar objective function

x - n vector of variables

$g(x)$ - inequality constraints, m vector

$h(x)$ - meq equality constraints.

Sufficient Condition for Unique Optimum

- $f(x)$ must be *convex*, and
- feasible region must be convex,

i.e. $g(x)$ are all *convex*

$h(x)$ are all *linear*

Except in special cases, there is no guarantee that a local optimum is global if sufficient conditions are violated.

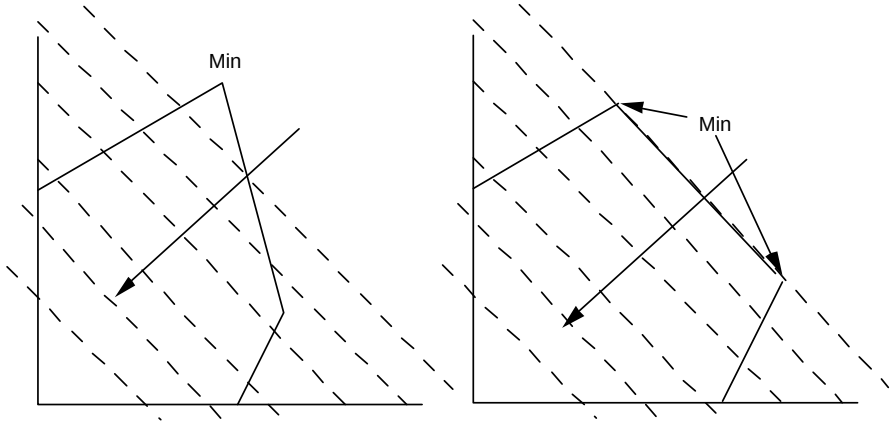
Variable Classification for NLPs

$$\begin{aligned} &\textit{Min} \quad f(x) \\ &\text{s.t.} \quad g(x) + s = 0 \text{ (add slack variable)} \\ &\qquad h(x) = 0,\ s \geq 0 \end{aligned} \qquad \Rightarrow \qquad \begin{aligned} &\textit{Min} \quad f(z) \\ &\text{s.t. } c(z) = 0 \\ &\qquad a \leq z \leq b \end{aligned}$$

- Partition variables into:
  - z_B - dependent or basic variables
  - z_N - nonbasic variables, fixed at a bound
  - z_S - independent or superbasic variables

  Analogy to linear programming. Superbasics required only if nonlinear problem

Characterization of Constrained Optima (two primal, two slacks)



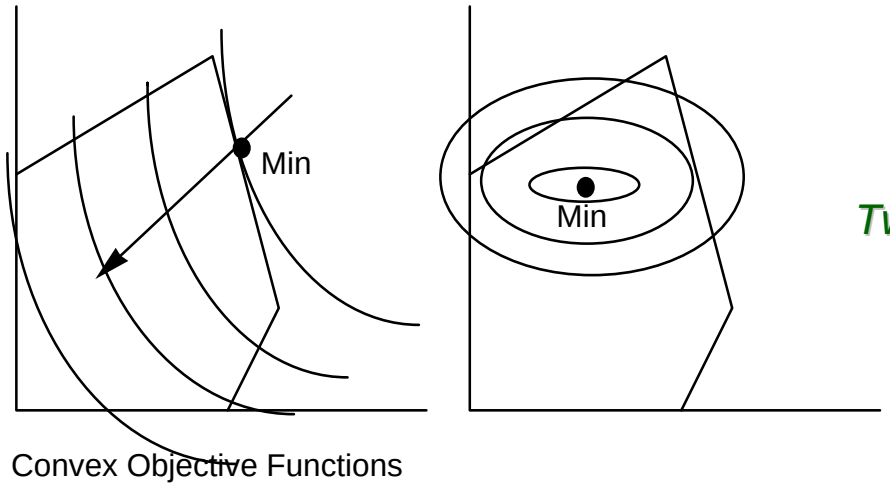
Linear Program

Linear Program
(Alternate Optima)

Two nonbasic variables, two basic, no superbasic

One nonbasic variable

Two basic, one superbasic

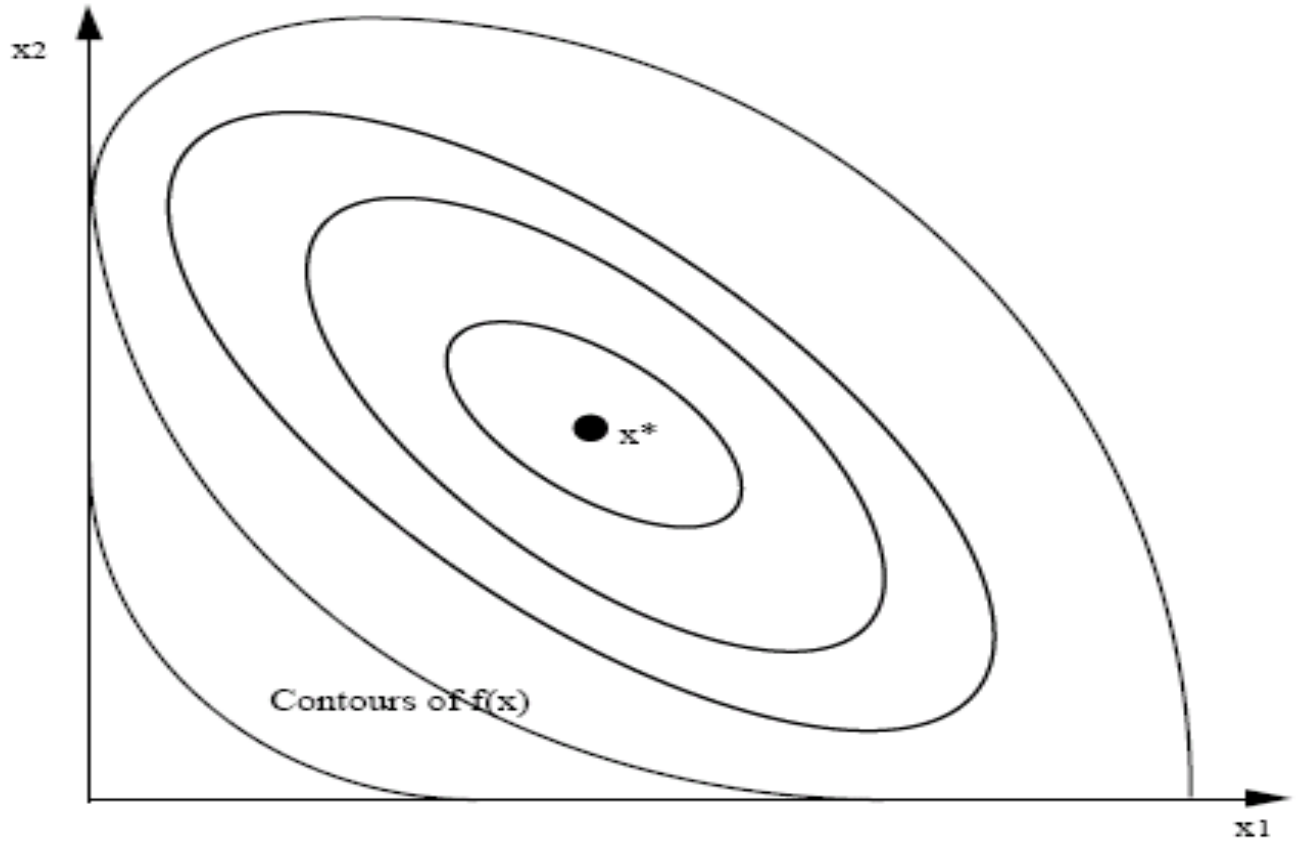


Convex Objective Functions
Linear Constraints

No nonbasic variables

Two basic, two superbasic

What conditions characterize an optimal solution?



Unconstrained Local Minimum
Necessary Conditions

$\nabla f(x^*) = 0$

$p^T \nabla^2 f(x^*)\, p \geq 0 \quad \text{for} \quad p \in \Re^n$

(positive semi-definite)

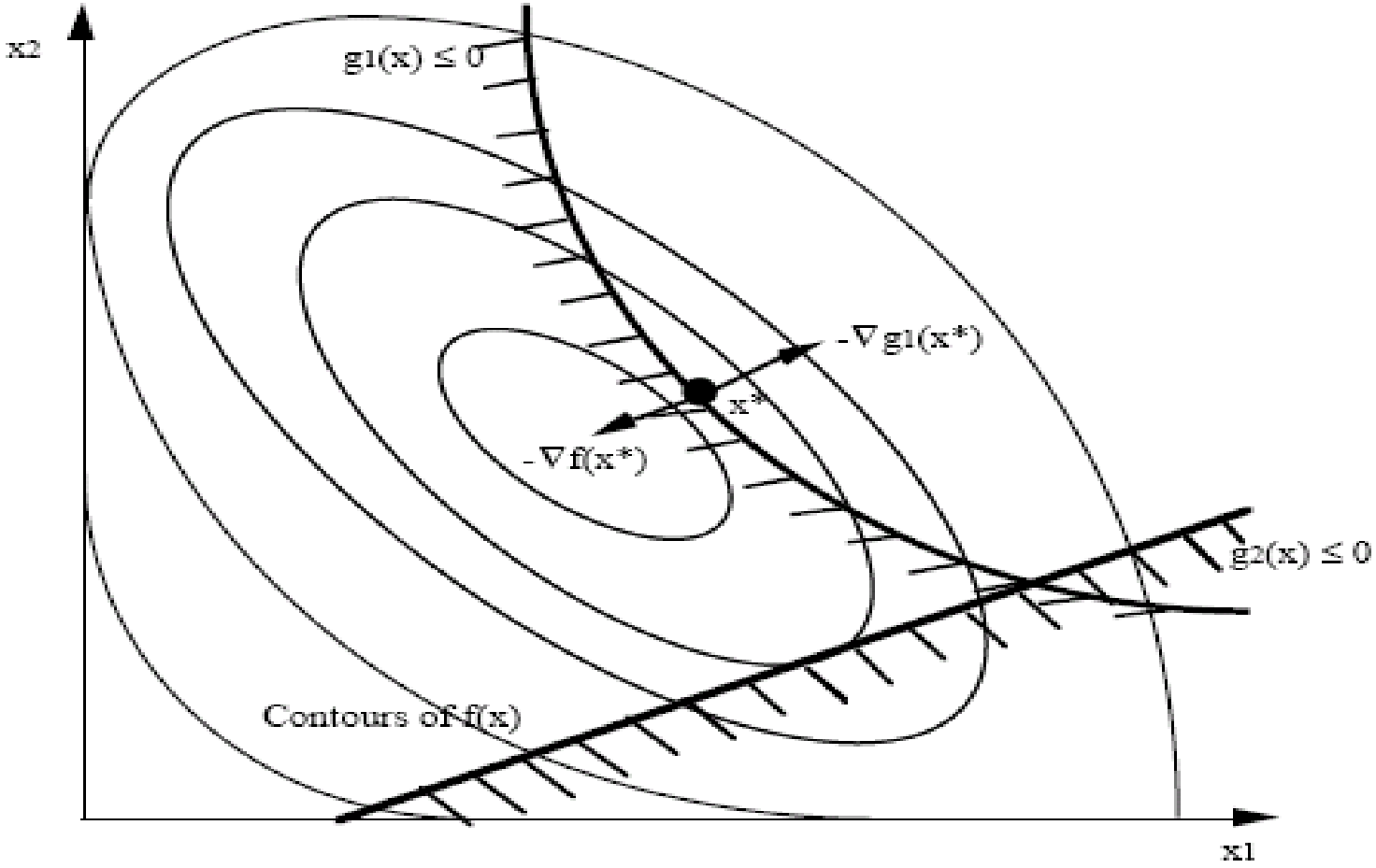
Unconstrained Local Minimum
Sufficient Conditions

$\nabla f(x^*) = 0$

$p^T \nabla^2 f(x^*)\, p > 0 \quad \text{for} \quad p \in \Re^n$

(positive definite)

Optimal solution for inequality constrained problem

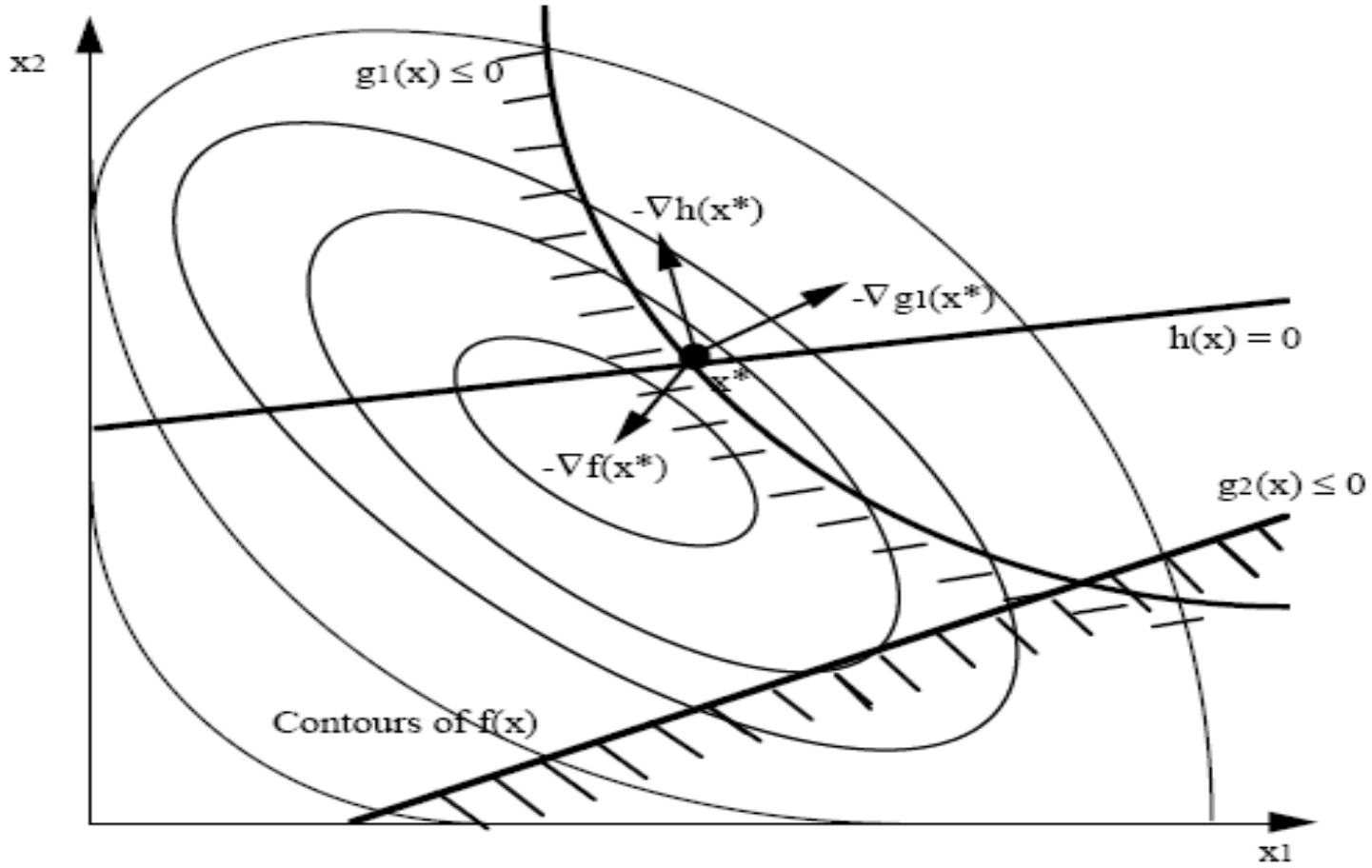


Min f(x)
s.t. $g(x) \leq 0$

Analogy: Ball rolling down valley pinned by fence

Note: Balance of forces (∇f, ∇g_1)

Optimal solution for general constrained problem



Problem: Min $f(x)$

s.t. $g(x) \leq 0$

$h(x) = 0$

Analogy: Ball rolling on rail pinned by fences

Balance of forces: ∇f, ∇g_1, ∇h

Optimality conditions for local optimum

Necessary First Order Karush Kuhn - Tucker Conditions

$\nabla L(z^*, u, v) = \nabla f(x^*) + \nabla g(x^*)\, u + \nabla h(x^*)\, v = 0$ (Balance Forces)

$u \geq 0$ (Inequalities act in only one direction)

$g(x^*) \leq 0, \; h(x^*) = 0$ (Feasibility)

$u_j\, g_j(x^*) = 0$ (Complementarity: either $g_j(x^*) = 0$ or $u_j = 0$)

u, v are "weights" for "forces," known as KKT multipliers, shadow prices, dual variables

"To guarantee that a local NLP solution satisfies KKT conditions, a constraint qualification is required. E.g., the *Linear Independence Constraint Qualification* (LICQ) requires active constraint gradients, $[\nabla g_A(x^*) \;\; \nabla h(x^*)]$, to be linearly independent. Also, under LICQ, KKT multipliers are uniquely determined."

Necessary (Sufficient) Second Order Conditions

- Positive curvature in "constraint" directions.
- $p^T \nabla^2 L(x^*)\, p \geq 0 \;\; (p^T \nabla^2 L(x^*)\, p > 0)$

  where p are the constrained directions: $\nabla g_A(x^*)^T p = 0, \; \nabla h(x^*)^T p = 0$

This is the space of the superbasic variables!

Optimality conditions for local optimum

$$Min \quad f(z),\ s.t.\ c(z) = 0,\ a \le z \le b$$

$$\nabla L(z^*, \lambda, \nu) = \nabla f(z^*) + \nabla c(z^*)\,\lambda - \nu_a + \nu_b = 0 \quad \text{(Balance Forces)}$$

$$a \le z^* \le b,\ \ c(z^*) = 0 \quad \text{(Feasibility)}$$

$$0 \le z^* - a \ \ perp \ \ \nu_a \ge 0$$

$$0 \le b - z^* \ perp \ \ \nu_b \ge 0$$

In terms of partitioned variables (when known)

<u>Basic</u> ($n_b = m$): $\nabla_B L(z^*, \lambda, \nu) = \nabla_B f(z^*) + \nabla_B c(z^*)\,\lambda = 0$

$$c(z^*) = 0$$

Square system in z_B *and* λ

<u>Nonbasic</u> (n_n): $\nabla_N L(z^*, \lambda, \nu) = \nabla_N f(z^*) + \nabla_N c(z^*)\,\lambda - \nu_{bnd} = 0$

$$z^* = bnd$$

Square system in z_N *and* ν

<u>Superbasic</u> (n_s): $\nabla_S L(z^*, \lambda, \nu) = \nabla_S f(z^*) + \nabla_S c(z^*)\,\lambda = 0$

Square system in z_S

$$n = n_n + n_b + n_s$$

Handling Basic Variables

$$\nabla_B L(z^*, \lambda, \nu) = \nabla_B f(z^*) + \nabla_B c(z^*)\,\lambda = 0$$
$$c(z^*) = 0$$

Full space: linearization and simultaneous solution of c(z) = 0 with stationarity conditions.

– embedded within Newton method for KKT conditions (SQP and Barrier)

Reduced space (two flavors):

- elimination of nonlinear equations and z_B at every point (feasible path GRG methods)
- linearization, then elimination of linear equations and z_B (MINOS and rSQP)

Handling Nonbasic Variables

$$\nabla_N L(z^*, \lambda, \nu) = \nabla_N f(z^*) + \nabla_N c(z^*)\,\lambda - \nu_{bnd} = 0$$
$$z^* = bnd$$

Barrier approach - primal and primal-dual approaches (IPOPT, KNITRO, LOQO)

Active set (two flavors)

- *QP or LP subproblems - primal-dual pivoting (SQP)*
- *gradient projection in primal space with bound constraints (GRG)*

Gradient Projection Method
(nonbasic variable treatment)

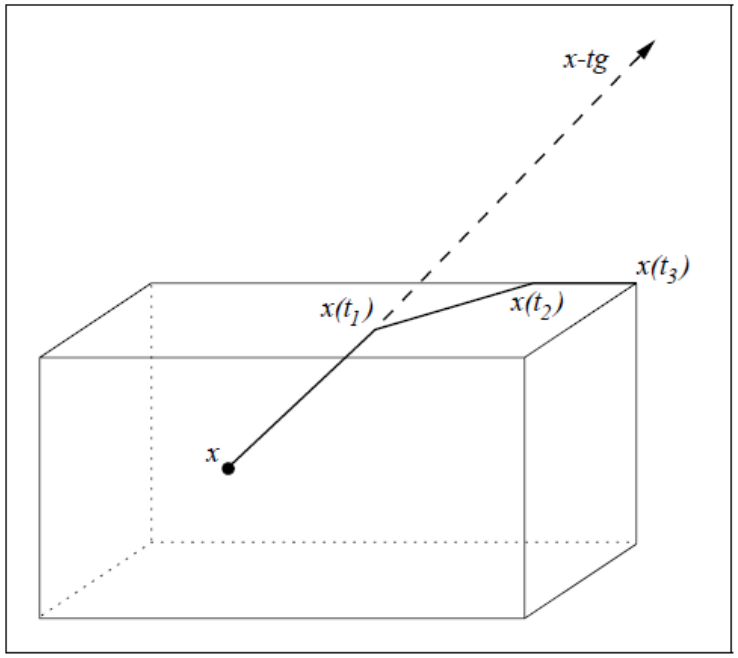


Figure 16.5 The piecewise linear path $x(t)$, for an example in $\mathbf{R}^3$.

Define the projection of an arbitrary point x onto box feasible region.

The ith component is given by

$$P(x,l,u)_i = \begin{cases} l_i & \text{if} \quad x_i < l_i, \\ x_i & \text{if} \quad x_i \in [l_i, u_i], \\ u_i & \text{if} \quad x_i > u_i. \end{cases}$$

Piecewise linear path $x(t)$ starting at the reference point x_0 and obtained by projecting steepest descent direction at x_0 onto the box region is given by

$$x(t) = P(x^0 - tg, l, u),$$

Handling Superbasic Variables

$$\nabla_S L(z^*, \lambda, v) = \nabla_S f(z^*) + \nabla_S c(z^*)\,\lambda = 0$$

Second derivatives or quasi-Newton?

$$B^{k+1} = B^k + \frac{yy^T}{s^T y} - \frac{B^k s\, s^T B^k}{s\, B^k s}$$

BFGS Formula $s = x^{k+1} - x^k$

$$y = \nabla L(x^{k+1}, u^{k+1}, v^{k+1}) - \nabla L(x^k, u^{k+1}, v^{k+1})$$

- second derivatives approximated by change in gradients
- positive definite B^k ensures *unique* search direction and descent property

Exact Hessian – requires far fewer iterations (no buildup needed in space of superbasics)

full-space - exploit sparsity (efficient large-scale solvers needed)

reduced space - update or project (expensive dense algebra if ns large)

Algorithms for Constrained Problems

Motivation: Build on unconstrained methods for superbasics wherever possible.

Classification of Methods:

- Reduced Gradient Methods - (with Restoration) GRG2, CONOPT
- Reduced Gradient Methods - (without Restoration) MINOS
- Successive Quadratic Programming - generic implementations
- Barrier Functions - popular in 1970s, but fell into disfavor. Barrier Methods have been developed recently and are again popular.
- Successive Linear Programming - only useful for "mostly linear" problems, cannot handle superbasic variables in a systematic way

We will concentrate on algorithms for first four classes.

Reduced Gradient Methods

- Nonlinear elimination of basic variables (and multipliers, λ)
- Use gradient projection to update and remove nonbasic variables
- Solve problem in reduced space of *superbasic* variables using the concept of reduced gradients

$$\frac{df}{dz_S} = \frac{\partial f}{\partial z_S} + \frac{dz_B}{dz_S}\frac{\partial f}{\partial z_B}$$

Because $c(z) = 0$, we have :

$$dc = \left[\frac{\partial c}{\partial z_S}\right]^T dz_S + \left[\frac{\partial c}{\partial z_B}\right]^T dz_B = 0$$

$$\frac{dz_B}{dz_S} = -\left[\frac{\partial c}{\partial z_S}\right]\left[\frac{\partial c}{\partial z_B}\right]^{-1} = -\nabla_{z_S} c\left[\nabla_{z_B} c\right]^{-1}$$

This leads to reduced gradient for superbasic :

$$\frac{df}{dz_S} = \frac{\partial f}{\partial z_S} - \nabla_{z_S} c\left[\nabla_{z_B} c\right]^{-1}\frac{\partial f}{\partial z_B}$$

Example of Reduced Gradient

$$Min \quad x_1^{\,2} - 2x_2$$

$$s.t. \quad 3x_1 + 4x_2 = 24$$

$$\nabla c^T = [3 \;\; 4], \;\; \nabla f^T = [2x_1 \;\; -2]$$

Let $z_S = x_1, z_B = x_2$

$$\frac{df}{dz_S} = \frac{\partial f}{\partial z_S} - \nabla_{z_S} c \left[\nabla_{z_B} c\right]^{-1} \frac{\partial f}{\partial z_B}$$

$$\frac{df}{dx_1} = 2x_1 - 3\left[4\right]^{-1}(-2) = 2x_1 + 3/2$$

If ∇c^T is (m x n); $\nabla z_S c^T$ is m x (n-m); $\nabla z_B c^T$ is (m x m)

(df/dz_S) is the change in f along constraint direction per unit change in z_S

Sketch of GRG Algorithm

1. Initialize problem and obtain a feasible point at z^0
2. At feasible point z^k, partition variables z into z_N, z_B, z_S
3. Remove nonbasics
4. Calculate reduced gradient, (df/dz_S)
5. Evaluate search direction for z_S, $d = B^{-1}(df/dz_S)$
6. Perform a line search.
   - Find $\alpha \in (0,1]$ with $z_S := z_S^k + \alpha\, d$
   - Solve for $c(z_S^k + \alpha\, d, z_B, z_N) = 0$
   - If $f(z_S^k + \alpha\, d, z_B, z_N) < f(z_S^k, z_B, z_N)$, set $z_S^{k+1} = z_S^k + \alpha\, d$, $k := k+1$
7. If $\|(df/dz_S)\| < \varepsilon$, Stop. Else, go to 2.

GRG Algorithm Properties

1. GRG2 has been implemented on PC's as GINO and is very reliable and robust. It is also the optimization solver in MS EXCEL.
2. CONOPT is implemented in GAMS, AIMMS and AMPL
3. GRG2 uses Q-N for small problems but can switch to conjugate gradients if problem gets large.
4. CONOPT does the same but reverts to an SQP method where it then uses exact second derivatives. This is a "local" feature.
5. Convergence of $c(z_S, z_B, z_N) = 0$ can get very expensive because inner equation solutions are required.
6. Safeguards can be added so that restoration (step 5.) can be dropped and efficiency increases.
7. Line search used based on improvement of $f(z)$.

➔No global convergence proof

Reduced Gradient Method without Restoration "Linearize then Eliminate Basics" (MINOS/Augmented)

Motivation: Efficient algorithms are available that solve linearly constrained optimization problems (MINOS):

$$\begin{aligned} &Min \quad f(x) \\ &s.t.\ Ax \le b \\ &\quad\ Cx = d \end{aligned}$$

Extend to nonlinear problems, through successive linearization

Develop major iterations (linearizations) and minor iterations (GRG solutions) .

Strategy: (Robinson, Murtagh & Saunders)

1. Partition variables into basic, nonbasic variables and superbasic variables..
2. Linearize active constraints at z^k
$$D^k z = r^k$$
3. Let $\psi = f(z) + v^T c(z) + \beta (c(z)^T c(z))$ (Augmented Lagrange),
4. Solve linearly constrained problem:
$$\begin{aligned} &Min \quad \psi(z) \\ &s.t. \quad Dz = r \\ &\qquad a \le z \le b \end{aligned}$$
using reduced gradients to get z^{k+1}
5. Set $k=k+1$, go to 3.
6. Algorithm terminates when no movement between steps 3) and 4).

MINOS/Augmented Notes

1. MINOS has been implemented very efficiently to take care of linearity. It becomes LP Simplex method if problem is totally linear. Also, very efficient matrix routines.
2. No restoration takes place, nonlinear constraints are reflected in $\psi(z)$ during step 3). MINOS is more efficient than GRG.
3. Major iterations (steps 3) - 4)) converge at a quadratic rate.
4. Reduced gradient methods are complicated, monolithic codes: hard to integrate efficiently into modeling software.
5. No globalization and no global convergence proof

Successive Quadratic Programming (SQP)

Motivation:

- Take KKT conditions, expand in Taylor series about current point.
- Take Newton step (QP) to determine next point.

Derivation – KKT Conditions

$$\nabla_x L(x^*, u^*, v^*) = \nabla f(x^*) + \nabla g_A(x^*)\, u^* + \nabla h(x^*)\, v^* = 0$$

$$h(x^*) = 0$$

$$g_A(x^*) = 0, \qquad \text{where } g_A \text{ are the active constraints.}$$

Newton - Step

$$\begin{bmatrix} \nabla_{xx} L & \nabla g_A & \nabla h \\ \nabla g_A^T & 0 & 0 \\ \nabla h^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta v \end{bmatrix} = - \begin{bmatrix} \nabla_x L\left(x^k, u^k, v^k\right) \\ g_A\left(x^k\right) \\ h\left(x^k\right) \end{bmatrix}$$

Requirements:

- $\nabla_{xx} L$ must be calculated and should be 'regular'
- correct active set g_A, *need to know nonbasic variables!*
- good estimates of u^k, v^k

SQP Chronology

1. *Wilson (1963)*
   - active set (nonbasics) can be determined by solving QP:

$$\begin{aligned} \underset{d}{Min} \quad & \nabla f(x_k)^T d + 1/2\, d^T \nabla_{xx} L(x_k, u_k, v_k)\, d \\ s.t. \quad & g(x_k) + \nabla g(x_k)^T d \le 0 \\ & h(x_k) + \nabla h(x_k)^T d = 0 \end{aligned}$$

2. *Han (1976), (1977), Powell (1977), (1978)*
   - approximate $\nabla_{xx} L$ using a positive definite quasi-Newton update (BFGS)
   - use a line search to converge from poor starting points.

Notes:
- Similar methods were derived using penalty (not Lagrange) functions.
- Method converges quickly; very few function evaluations.
- Not well suited to large problems (full space update used).
  For n > 100, say, use reduced space methods (e.g. MINOS).

Elements of SQP – Search Directions

How do we obtain search directions?

- Form QP and let QP determine constraint activity
- At each iteration, k, solve:

$$\begin{aligned} \underset{d}{Min} \quad & \nabla f(x^k)^T d + 1/2\, d^T B^k d \\ s.t. \quad & g(x^k) + \nabla g(x^k)^T d \le 0 \\ & h(x^k) + \nabla h(x^k)^T d = 0 \end{aligned}$$

Convergence from poor starting points

- As with Newton's method, choose α (stepsize) to ensure progress toward optimum: $x^{k+1} = x^k + \alpha\, d$.
- α is chosen by making sure a *merit function* is decreased at each iteration.

Exact Penalty Function

$$\psi(x) = f(x) + \mu\,[\Sigma \max(0, g_j(x)) + \Sigma |h_j(x)|]$$

$$\mu > \max_j \{|u_j|, |v_j|\}$$

Augmented Lagrange Function

$$\psi(x) = f(x) + u^T g(x) + v^T h(x) + \eta/2\,\{\Sigma (h_j(x))^2 + \Sigma \max(0, g_j(x))^2\}$$

Newton-Like Properties for SQP

Fast Local Convergence

$B = \nabla_{xx}L$	Quadratic
$\nabla_{xx}L$ is p.d and B is Q-N	1 step Superlinear
B is Q-N update, $\nabla_{xx}L$ not p.d	2 step Superlinear

Enforce Global Convergence

Ensure decrease of merit function by taking $\alpha \leq 1$

Trust region adaptations provide a stronger guarantee of global convergence - but harder to implement.

Basic SQP Algorithm

0. Guess x^0, Set $B^0 = I$ (Identity). Evaluate $f(x^0)$, $g(x^0)$ and $h(x^0)$.
1. At x^k, evaluate $\nabla f(x^k)$, $\nabla g(x^k)$, $\nabla h(x^k)$.
2. If k > 0, update B^k using the BFGS Formula.
3. Solve:
$$\underset{d}{Min} \quad \nabla f(x^k)^T d + 1/2\, d^T B^k d$$
$$s.t. \quad g(x^k) + \nabla g(x^k)^T d \le 0$$
$$h(x^k) + \nabla h(x^k)^T d = 0$$

   If KKT error less than tolerance: $\|\nabla L(x^*)\| \le \varepsilon$, $\|h(x^*)\| \le \varepsilon$, $\|g(x^*)_+\| \le \varepsilon$. STOP, else go to 4.
4. Find α so that $0 < \alpha \le 1$ and $\psi(x^k + \alpha d) < \psi(x^k)$ sufficiently

   (Each trial requires evaluation of $f(x)$, $g(x)$ and $h(x)$).
5. $x^{k+1} = x^k + \alpha\, d$. Set k = k + 1 Go to 2.

Large-Scale SQP

$$\begin{aligned} \textit{Min} \quad & f(z) \\ \textit{s.t.} \quad & c(z)=0 \\ & z_L \le z \le z_U \end{aligned} \qquad \begin{aligned} \textit{Min} \quad & \nabla f(z^k)^T d + 1/2\, d^T W^k d \\ \textit{s.t.} \quad & c(z^k) + (A^k)^T d = 0 \\ & z_L \le z^k + d \le z_U \end{aligned}$$

Active set strategy (usually) applied to QP problems (could use interior point) to handle nonbasic variables

Few superbasics (10 - 100)
Apply reduced space (linearized) elimination of basic variables

Many superbasics (≥ 1000)
Apply full-space method

Few degrees of freedom => reduced space SQP (rSQP)

- Take advantage of sparsity of $A = \nabla c(x)$
- project W into space of active (or equality constraints)
- curvature (second derivative) information only needed in space of degrees of freedom
- second derivatives can be applied or approximated with positive curvature (e.g., BFGS)
- use dual space QP solvers

+ easy to implement with existing sparse solvers, QP methods and line search techniques
+ exploits 'natural assignment' of dependent and decision variables (some decomposition steps are 'free')
+ does not require second derivatives

- reduced space matrices are dense
- may be dependent on variable partitioning
- can be very expensive for many degrees of freedom
- can be expensive if many QP bounds

Reduced space SQP (rSQP)
Range and Null Space Decomposition

Assume nonbasics removed, QP problem with *n* variables and *m* constraints becomes:

$$\begin{bmatrix} W^k & A^k \\ A^{kT} & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda_+ \end{bmatrix} = -\begin{bmatrix} \nabla f(x^k) \\ c(x^k) \end{bmatrix}$$

- Define reduced space basis, $Z^k \in \Re^{n \times (n-m)}$ with $(A^k)^T Z^k = 0$
- Define basis for remaining space $Y^k \in \Re^{n \times m}$, $[Y^k\ Z^k] \in \Re^{n \times n}$ is nonsingular.
- Let $d = Y^k d_Y + Z^k d_Z$ to rewrite:

$$\begin{bmatrix} \begin{bmatrix} Y^k & Z^k \end{bmatrix}^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} W^k & A^k \\ A^{kT} & 0 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} Y^k & Z^k \end{bmatrix} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} d_Y \\ d_Z \\ \lambda_+ \end{bmatrix} = -\begin{bmatrix} \begin{bmatrix} Y^k & Z^k \end{bmatrix}^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \nabla f(x^k) \\ c(x^k) \end{bmatrix}$$

Reduced space SQP (rSQP)
Range and Null Space Decomposition

$$\begin{bmatrix} Y^{k^T} W^k Y^k & Y^{k^T} W^k Y^k & Y^{k^T} A^k \\ Z^{k^T} W^k Y^k & Z^{k^T} W^k Z^k & 0 \\ A^{k^T} Y^k & 0 & 0 \end{bmatrix} \begin{bmatrix} d_Y \\ d_Z \\ \lambda_+ \end{bmatrix} = -\begin{bmatrix} Y^{k^T} \nabla f(x^k) \\ Z^{k^T} \nabla f(x^k) \\ c(x^k) \end{bmatrix}$$

0 0

- $(A^T Y)\, d_Y = -c(x^k)$ is square, d_Y determined from bottom row.
- Cancel $Y^T W Y$ and $Y^T W Z$; (unimportant as d_Z, d_Y --> 0)
- $(Y^T A)\, \lambda = -Y^T \nabla f(x^k)$, λ can be determined by first order estimate
- Calculate or approximate $w = Z^T W Y\, d_Y$, solve $Z^T W Z\, d_Z = -Z^T \nabla f(x^k) - w$
- Compute total step: $d = Y\, dY + Z\, dZ$

Barrier Methods for Large-Scale Nonlinear Programming

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

Original Formulation

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

$$x \geq 0$$

Can generalize for

$$a \leq x \leq b$$

Barrier Approach

$$\min_{x \in \Re^n} \varphi_\mu(x) = f(x) - \mu \sum_{i=1}^{n} \ln x_i$$

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

$\Rightarrow$ As μ ➔ 0, $x^*(\mu)$ ➔ x^* Fiacco and McCormick (1968)

Solution of the Barrier Problem

⇒**Newton Directions (KKT System)**

$$\begin{aligned} \nabla f(x) + A(x)\lambda - v &= 0 \\ Xv - \mu e &= 0 \\ c(x) &= 0 \end{aligned}$$

$e^T = [1, 1, 1 \ldots]$

$X = diag(x)$

⇒ **Reducing the System**

$$d_v = \mu X^{-1} e - v - X^{-1} V d_x$$

$$\begin{bmatrix} W+\Sigma & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} d_x \\ \lambda^+ \end{bmatrix} = -\begin{bmatrix} \nabla\varphi_\mu \\ c \end{bmatrix}$$

$$\Sigma = X^{-1} V$$

IPOPT Code – www.coin-or.org

Global Convergence of Newton-based Barrier Solvers

Merit Function

Exact Penalty: $P(x, \eta) = f(x) + \eta \, ||c(x)||$

Aug'd Lagrangian: $L^*(x, \lambda, \eta) = f(x) + \lambda^T c(x) + \eta \, ||c(x)||^2$

Assess Search Direction (e.g., from IPOPT)

Line Search – choose *stepsize* α to give sufficient decrease of merit function using a 'step to the boundary' rule with $\tau \sim 0.99$.

$$\text{for } \alpha \in (0, \bar{\alpha}], \; x_{k+1} = x_k + \alpha \, d_x$$

$$x_k + \bar{\alpha} \, d_x \geq (1-\tau) x_k > 0$$

$$v_{k+1} = v_k + \bar{\alpha} \, d_v \geq (1-\tau) v_k > 0$$

$$\lambda_{k+1} = \lambda_k + \alpha \, (\lambda_+ - \lambda_k)$$

- How do we balance $\phi(x)$ and $c(x)$ with η?
- Is this approach globally convergent? Will it still be fast?

Line Search Filter Method

Store (ϕ_k, θ_k) at allowed iterates

Allow progress if trial point is acceptable to filter with θ margin

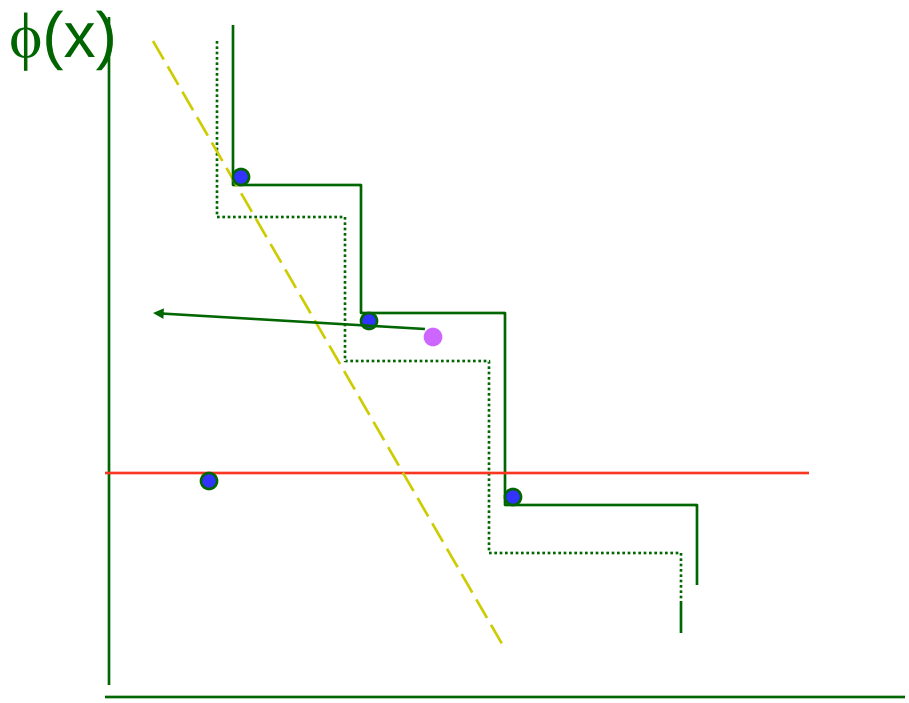
If switching condition

$$\alpha[-\nabla\phi_k^T d]^a \geq \delta[\theta_k]^b, a > 2b > 2$$

is satisfied, only an Armijo line search is required on ϕ_k

If insufficient progress on stepsize, evoke restoration phase to reduce θ.

Global convergence and superlinear local convergence proved (with second order correction)



IPOPT Algorithm – Features

Line Search Strategies for Globalization

- ℓ_2 exact penalty merit function

- Filter method (adapted and extended from Fletcher and Leyffer)

Hessian Calculation

- BFGS (full/LM)

- SR1 (full/LM)

- Exact full Hessian (direct)

- Exact reduced Hessian (direct)

- Preconditioned CG

Algorithmic Properties

Globally, superlinearly convergent (Wächter and B., 2005)

Easily tailored to different problem structures

Freely Available

CPL License and COIN-OR distribution:
http://www.coin-or.org

Beta version recently rewritten in C++

Solved on thousands of test problems and applications

Trust Region Barrier Method

KNITRO

Consider Jacobian $A_k = [N_k \; C_k]$

$N_k \in \mathbb{R}^{m\times(n-m)}$, $C_k \in \mathbb{R}^{m\times m}$ nonsingular

Null-space of A_k : $Z_k \;\Rightarrow\; A_k Z_k = 0$

Consider Y_k s.t. $[Z_k \; Y_k]$ nonsingular

Now, $d := Y_k p_Y + Z_k p_Z$

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix}\begin{bmatrix} d \\ \lambda \end{bmatrix} = -\begin{bmatrix} g_k \\ c_k \end{bmatrix}$$

First-order necessary conditions of QP:

$$\begin{bmatrix} Z_k^\top W_k Z_k & Z_k^\top W_k Y_k & 0 \\ Y_k^\top W_k Z_k & Y_k^\top W_k Y_k & Y_k^\top A_k^\top \\ 0 & A_k Y_k & 0 \end{bmatrix}\begin{bmatrix} p_Z \\ p_Y \\ \lambda_k \end{bmatrix} = -\begin{bmatrix} Z_k^\top g_k \\ Y_k^\top g_k \\ c_k \end{bmatrix},$$

QP $\equiv$ 2 subproblems:

Quasi-normal sub-problem: $c_k + A_k Y_k p_Y = 0$

Tangential sub-problem: $\min_{p_Z} \quad (Z_k^\top g_k + w_k)^\top p_Z + \frac{1}{2} p_Z^\top W_k p_Z$

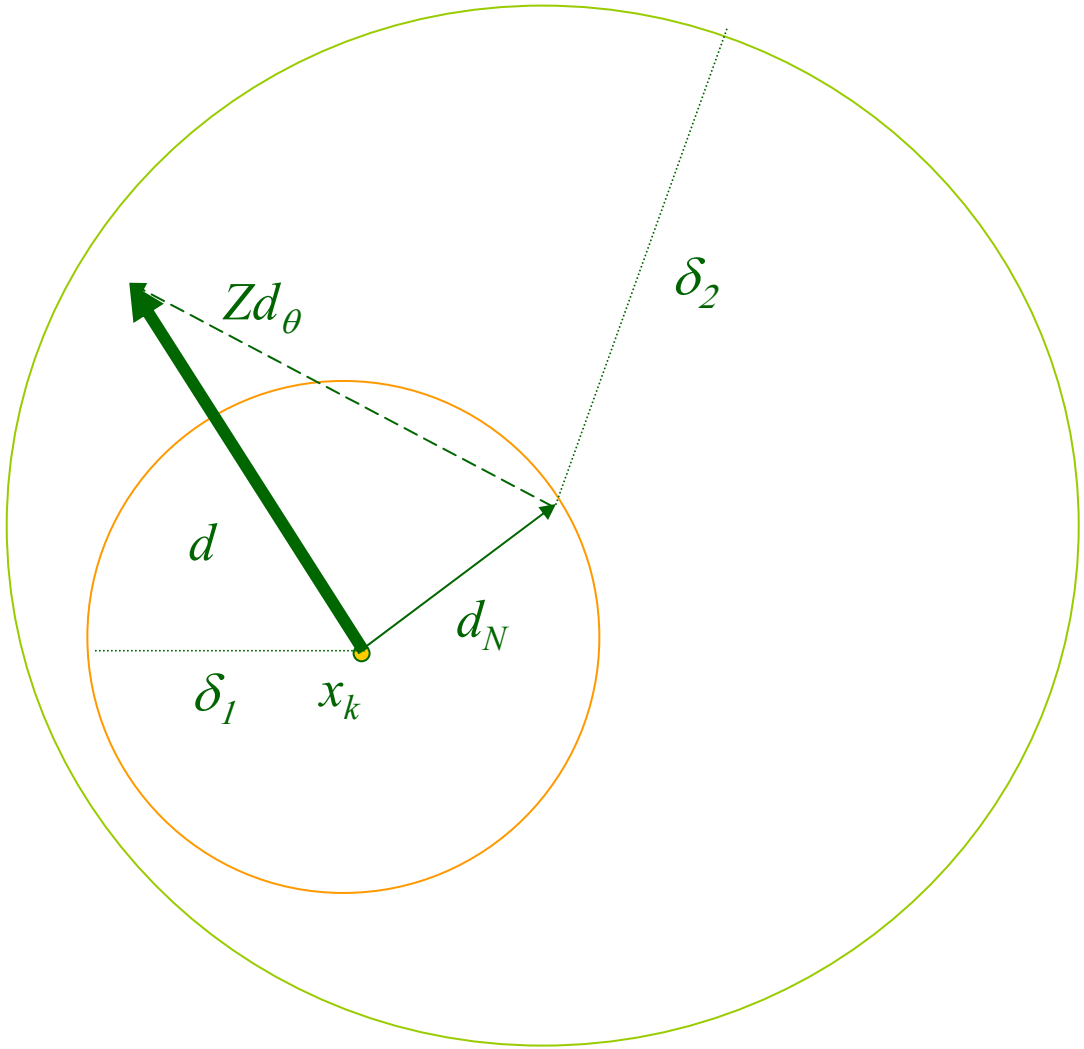
$$p_Z^L \le p_Z \le p_Z^U,$$

Composite Step Trust-region SQP

Split barrier problem step into 2 parts: *normal* (basic) and *tangential*

(nonbasic and superbasic) search directions

Trust-region on tangential component: larger combined radius



KNITRO Algorithm – Features

Overall method

- Similar to IPOPT

- But does not use filter method

- applies full-space Newton step and line search (LOQO is similar)

Trust Region Strategies for Globalization

- based on ℓ_2 exact penalty merit function

- expensive: used only when line search approach gets "into trouble"

Tangential Problem

- Exact full Hessian (direct)

- Preconditioned CG to solve tangential problem

Normal Problem

- use dogleg method based on Cauchy step

Algorithmic Properties

Globally, superlinearly convergent (Byrd and Nocedal, 2000)

Stronger convergence properties using trust region over line search

Available through Zenia

Available free to academics
http://www.neos.mcs.anl.gov

Solved on thousands of test problems and applications

Typical NLP algorithms and software

SQP - NPSOL, VF02AD, fmincon

reduced SQP - SNOPT, rSQP, MUSCOD, DMO, LSSOL…

GP + Elimination - GRG2, GINO, SOLVER, CONOPT

GP/Lin. Elimin. - MINOS

Second derivatives and barrier -

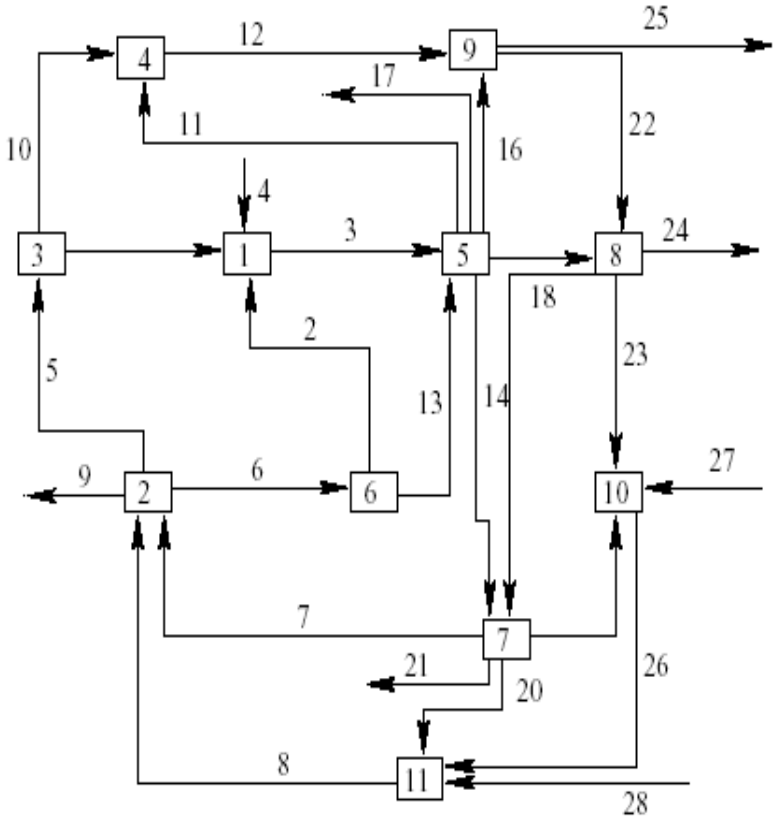
- IPOPT, KNITRO, LOQO

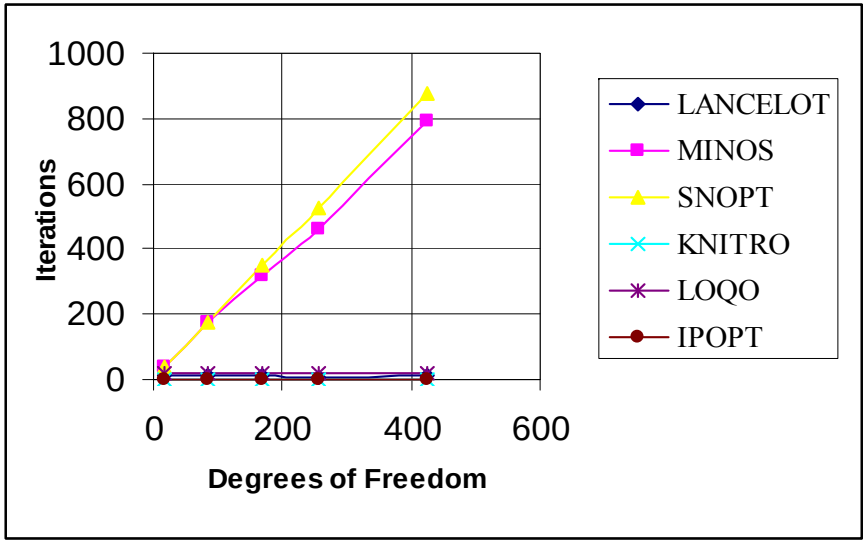
Interesting hybrids -

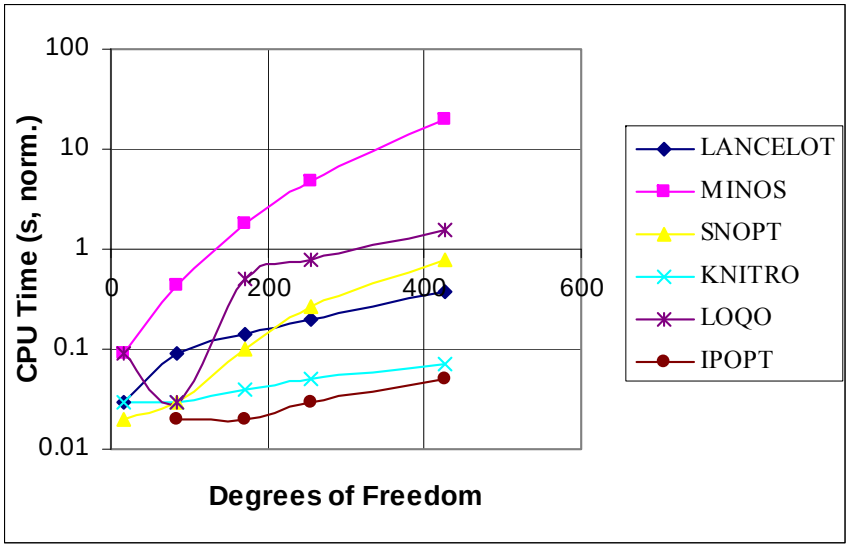
•FSQP/cFSQP - SQP and elimination

•LANCELOT (actually AL for equalities along with GP)

Comparison of NLP Solvers: Data Reconciliation (2004)

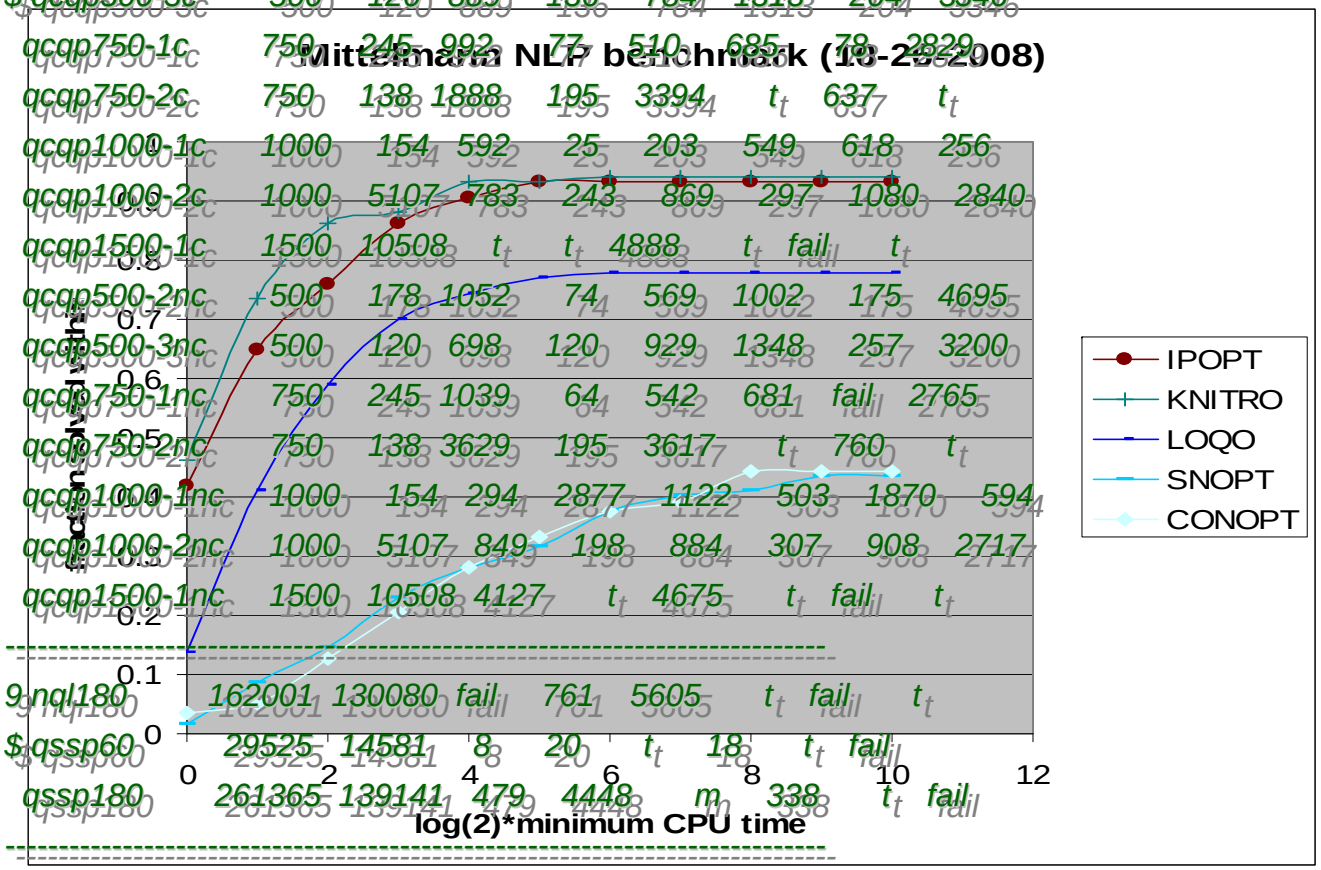






Comparison of NLP solvers

(latest Mittelmann study)



	Limits	Fail
IPOPT	7	2
KNITRO	7	0
LOQO	23	4
SNOPT	56	11
CONOPT	55	11

117 Large-scale Test Problems

500 - 250 000 variables, 0 – 250 000 constraints

NLP Summary

Widely used - solving nonlinear problems with thousands of variables and superbasics routinely

Exploiting sparsity and structure leads to solving problems with millions of variables (and superbasics)

Availability of structure and second derivatives is the key.

Global and fast local convergence properties are well-known for many (but not all algorithms)

Exceptions: LOQO, MINOS, CONOPT, GRG

Current Challenges

- Exploiting larger problems in parallel (shared and distributed memory),

- Irregular problems (e.g., singular problems, MPECs)

- Extracting accurate first and second derivatives from codes...

- Embedding robust NLPs within MINLP and global solvers in order to quickly detect infeasible solutions in subregions