

4TE3/6TE3

Algorithms for

Continuous Optimization

(Algorithms for Constrained
Nonlinear Optimization Problems)

Tamás TERLAKY
Computing and Software
McMaster University

Hamilton, November 2005

terlaky@mcmaster.ca

Tel: 27780

Algorithms for

constrained optimization

Linear Equality Constraints

$$(LEC) \quad \min f(x) \\ \text{s.t.} \quad Ax = b.$$

f is continuously differentiable, $A : m \times n$ is a matrix with $\text{rank}(A) = m$ and $b \in \mathbb{R}^m$.

Given a basis B then

$$Ax = Bx_B + Nx_N = b$$

we have

$$x_B = B^{-1}b - B^{-1}Nx_N$$

$$\min f_N(x_N)$$

where $f_N(x_N) = f(x) = f(B^{-1}b - B^{-1}Nx_N, x_N)$.

This is an unconstrained problem. Further,

$$\nabla f(x)^T = ((\nabla_B f(x))^T, (\nabla_N f(x))^T),$$

The **reduced gradient** can be expressed as:

$$\begin{aligned} \nabla f_N(x_N)^T &= -(\nabla_B f(x))^T B^{-1}N + (\nabla_N f(x))^T \\ &= ((\nabla_B f(x))^T, (\nabla_N f(x))^T) \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix}. \end{aligned}$$

The Reduced Hessian:

$$\nabla^2 f_N(x_N) = (-(B^{-1}N)^T, I) \nabla^2 f(x_N) \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix}.$$

Linear Equality Constraints

Null-space method - I

$$\begin{aligned} (LEC) \quad & \min f(x) \\ & \text{s.t. } Ax = b. \end{aligned}$$

f is continuously differentiable, $A : m \times n$ is a matrix with $\text{rank}(A) = m$ and $b \in \mathbb{R}^m$.

Let \bar{x} be feasible, i.e., $A\bar{x} = b$, then $Ax = A(\bar{x} + s) = b$, thus (LEC) is equivalent to:

$$\begin{aligned} (LEC) \quad & \min f(\bar{x} + s) \\ & \text{s.t. } As = 0. \end{aligned}$$

The vector s is from the null-space of the matrix A .

If the columns of Z (an $n \times (n - m)$ matrix) give a basis of the null-space of A , then $s = Zv$ with $v \in \mathbb{R}^{n-m}$.

Then (LEC) can be given by

$$(LEC) \quad \min h(v) = f(\bar{x} + Zv).$$

This is an unconstrained problem!

Linear Equality Constraints

Null-space method - II

The null-space can easily be given:

Let B be a basis from the column space of A , then

$$A = (B, N)$$

The range(row) space of A can be given by the basis vectors

$$R = (I, B^{-1}N).$$

The null-space is

$$Z^T = \left(-(B^{-1}N)^T, I \right).$$

Clearly $RZ = AZ = 0$.

The **gradient** of $h(v)$ can be expressed as

$$\nabla h(v) = Z^T \nabla f(\bar{x} + Zv).$$

The **Hessian** of $h(v)$ can be expressed as

$$\nabla^2 h(v) = Z^T \nabla^2 f(\bar{x} + Zv) Z.$$

The reduced gradient method

Linear (in)equality constraints

$$(LC) \quad \min f(x) \\ \text{s.t.} \quad Ax = b, \\ x \geq 0.$$

f is continuously differentiable, $A : m \times n$ is a matrix with $\text{rank}(A) = m$ and $b \in \mathbb{R}^m$.

Given a basis B and a feasible $x = (x_B, x_N)$ such that $x_B > 0$. x_N do not have to be zero!

$$Bx_B + Nx_N = b$$

we have

$$x_B = B^{-1}b - B^{-1}Nx_N \\ \min f_N(x_N) \\ \text{s.t.} \quad B^{-1}b - B^{-1}Nx_N \geq 0, \\ x_N \geq 0,$$

where $f_N(x_N) = f(x) = f(B^{-1}b - B^{-1}Nx_N, x_N)$ and

$$\nabla f(x)^T = ((\nabla_B f(x))^T, (\nabla_N f(x))^T).$$

The **reduced gradient** can be expressed as

$$-r := \nabla f_N(x_N)^T = -(\nabla_B f(x))^T B^{-1}N + (\nabla_N f(x))^T.$$

The reduced gradient method

Linear constraints:

$x^k \in \mathbb{R}^n$ is the current iterate;

the basis is nondegenerate;

Search direction: $s^T = (s_B^T, s_N^T)$ in $\mathcal{N}(A)$

$s_B = -B^{-1}Ns_N$ and s_N properly given,

the feasibility of $x^k + \lambda s$ is guaranteed as long as

$$x^k + \lambda s \geq 0, \quad \text{i.e.} \quad \lambda \leq \bar{\lambda} = \min_{1 \leq i \leq n, s_i < 0} \left\{ \frac{x_i^k}{-s_i} \right\}.$$

Further, s_N should be a descent direction of f .

$$s_j = \begin{cases} 0 & \text{if } x_j^k = 0 \text{ and } \frac{\partial f_N(x_N^k)}{\partial x_j} \geq 0, \\ -\frac{\partial f_N(x_N^k)}{\partial x_j} & \text{otherwise} \end{cases} \quad j \in N.$$

Make a line search:

$$x^{k+1} = \arg \min_{0 \leq \lambda \leq \bar{\lambda}} f(x^k + \lambda s).$$

If all the coordinates x_B^{k+1} stay strictly positive we keep the basis, else a pivot is made to eliminate the zero variable from the basis and replace it by a positive but currently non-basic coordinate.

The reduced gradient method

Convergent variant:

$x^k \in \mathbb{R}^n$ is the current iterate;

the basis is nondegenerate;

Search direction: $s^T = (s_B^T, s_N^T)$ in $\mathcal{N}(A)$

$s_B = -B^{-1}N s_N$ and s_N is given by

$$s_j = \begin{cases} -x_i r_j & \text{if } r_j \geq 0, \\ -r_j & \text{otherwise} \end{cases} \quad j \in N.$$

the feasibility of $x^k + \lambda s$ is guaranteed as long as

$$x^k + \lambda s \geq 0, \quad \text{i.e.} \quad \lambda \leq \bar{\lambda} = \min_{1 \leq i \leq n, s_i < 0} \left\{ \frac{x_i^k}{-s_i} \right\}.$$

Theorem 1 *The search direction s at x^k is always a descent direction unless $s = 0$. If $s = 0$, then x^k is a KKT point of problem (LC).*

Theorem 2 *Any accumulation point of the sequence $\{x^k\}$ is a KKT point.*

Remarks on

the reduced gradient method

- x^k is not necessarily a basic solution – *super-basic* variables.
- Degeneracy imply possible zero steps, thus anticycling techniques are needed.
- Convex simplex method:
Only one coordinate j of s_N can be nonzero.
 $s_j = -\frac{\partial f_N(x_N^k)}{\partial x_j} > 0$, the rest of s_N is zero;
 $s_B = -B^{-1}N s_N = -B^{-1}a_j s_j$.
- The simplex method for linear optimization (programming) is a further specialization for the case when $f(x) = c^T x$ for some $c \in R^n$ and for the initial basis B we have $x_N = 0$. The property $x_N = 0$ is preserved throughout the process.

As in the convex simplex method we let only one component j of s_N to be nonzero. Due to linearity, the line search will always get to the boundary and makes a basis coordinate (a coordinate of x_B) zero.

The Generalized

reduced gradient (GRG) method

$$(NC) \quad \min f(x) \\ \text{s.t.} \quad h_j(x) = 0, \quad j = 1, \dots, m \\ x \geq 0.$$

f, h_1, \dots, h_m are continuously differentiable.

Assumption: the gradients of the constraints h_j are linearly independent at every point $x \geq 0$.

A feasible $x^k \geq 0$ with $h_j(x^k) = 0 \quad \forall j$ is given.

The Jacobian $H(x) = (h_1(x), \dots, h_m(x))^T$ at each $x \geq 0$ has full rank. Denote it at x^k by $A = JH(x^k)$.

A basis B , with $x_B^k > 0$ is given.

For the linearized constraints we have

$$H(x^k) + JH(x^k)(x - x^k) = 0 + A(x - x^k) = 0.$$

From this one has

$$Bx_B + Nx_N = Ax^k$$

$$[\text{or} \quad Bx_B + Nx_N = Ax^k - H(x^k)]$$

and by introducing $b = Ax^k$ we have

$$x_B = B^{-1}b - B^{-1}Nx_N$$

$$\min f_N(x_N) \\ \text{s.t.} \quad B^{-1}b - B^{-1}Nx_N \geq 0, \\ x_N \geq 0,$$

GRG cntd.

where $f_N(x_N) = f(x) = f(B^{-1}b - B^{-1}Nx_N, x_N)$.

Using the notation

$$\nabla f(x)^T = ((\nabla_B f(x))^T, (\nabla_N f(x))^T),$$

the *reduced gradient* can be expressed as

$$\nabla f_N(x)^T = -(\nabla_B f(x))^T B^{-1}N + (\nabla_N f(x))^T.$$

Get search direction s goes exactly the same way as in the linearly constrained case.

Due to the nonlinearity of the constraints $H(x^{k+1}) = H(x^k + \lambda s) = 0$ will not hold!

Something to be done to recover feasibility.

SQP-I

Sequential quadratic programming

Equality constraints

$$(NC) \quad \min f(x) \\ \text{s.t.} \quad h_j(x) = 0, \quad j = 1, \dots, m$$

The Lagrange function is

$$L(x, y) = f(x) + \sum_{j=1}^m y_j h_j(x),$$

where $y_j \in R$, $j = 1, \dots, m$. Let us denote

$$H(x) = (h_1(x), \dots, h_m(x))^T.$$

Then the KKT conditions are:

$$\begin{aligned} \nabla_x L(x, y) &= 0 \\ H(x) &= 0. \end{aligned}$$

Let a candidate solution (x^k, y^k) be given and apply Newton's method to solve this nonlinear equation system:

$$\begin{aligned} \nabla_{xx}^2 L(x^k, y^k) \Delta x + (\nabla H(x^k))^T \Delta y &= -\nabla_x L(x^k, y^k) \\ \nabla H(x^k) \Delta x &= -H(x^k). \end{aligned}$$

SQP-II

Sequential quadratic programming

Equality constraints

This equation system

$$\begin{aligned}\nabla_{xx}^2 L(x^k, y^k) \Delta x + (\nabla H(x^k))^T \Delta y &= -\nabla_x L(x^k, y^k) \\ \nabla H(x^k) \Delta x &= -H(x^k)\end{aligned}$$

is the KKT condition of the following linearly constrained quadratic optimization problem:

$$\begin{aligned}\min \quad & \frac{1}{2} \Delta x^T \nabla_{xx}^2 L(x^k, y^k) \Delta x + \nabla_x L(x^k, y^k)^T \Delta x \\ \text{s.t.} \quad & \nabla H(x^k) \Delta x = -H(x^k).\end{aligned}$$

One needs to:

- solve this quadratic problem at each iteration,
- to make a line-search where feasibility and optimality need to be considered
- and repeating the process from the new point until the optimality condition is satisfied.

Barrier functions

How can one replace the constraint $t \geq 0$ (i.e., $-g_j(x) \geq 0$) by a good barrier function?

Desired properties of a barrier functions $B(t)$ of $t \geq 0$.

1. $B(t)$ is a smooth (infinitely many times) differentiable, strictly convex.
2. As combined with an $f(t)$, the function $f(t) + B(t)$ should not assume its minimum at $t = 0$. The derivative of $B(t)$ goes to $-\infty$ as $t \rightarrow 0$.
3. $B(t)$ goes to infinity as $t \rightarrow 0$.

Functions satisfying all three properties are

barrier functions,

functions satisfying the first two properties are

quasi-barriers.

Note:

For barrier functions you need inequality constrains!

Quasi-Barrier functions

Quasi barrier functions

- The *entropy* function $t \log t$.
- Let $0 < r < 1$. The function $-t^r$.

Barrier functions

1. The *logarithmic barrier* function $-\log t$.
 2. Let $r > 1$. The *inverse barrier* function t^{-r} .
-

Then the barrier function for the (CO) problem

$$\begin{aligned} (CO) \quad & \min f(x) \\ & \text{s.t. } g_j(x) \leq 0, \quad j = 1, \dots, m \end{aligned}$$

is given by

$$f_\mu(x) = \frac{f(x)}{\mu} + \sum_{j=1}^m B(-g_j(x)) \quad (1)$$

where $\mu > 0$. The original problem (CO) is solved by subsequentially minimizing the function $f_\mu(x)$ for a series of μ values as $\mu \rightarrow 0$.

Generic IPM

Input:

$\mu = \mu_0$ the barrier parameter value;

θ the reduction parameter, $0 < \theta < 1$;

$\epsilon > 0$ the accuracy parameter;

x^0 a given interior feasible point;

Step 0: $x := x^0, \mu := \mu_0$;

Step 1: If $\mu < \epsilon$ STOP, $x(\mu)$ is returned as solution.

Step 2: Calculate (approximately) $x(\mu)$;

Step 3: $\mu := (1 - \theta)\mu$;

Step 4: GO TO Step 1.

Remarks on IPMs

- IPM for LO will be discussed.
- Newton's method is used to calculate an approximation of the new target point $x(\mu)$.
- A smoothness condition **self-concordancy**, is needed to control Newton efficiently.
- A proximity measure: the length of the Newton step in the Hessian norm.
- If the measure is "large", a fix reduction is realized in the barrier value.
If the measure is "small", Newton is quadratically convergent.
- Depending on θ we have small or large step methods. In small step methods μ is reduced by a small value (e.g. multiplied by $(1 - \frac{\gamma}{\sqrt{n}})$ with some $\gamma > 0$) at each iteration, while in large step methods it is significantly reduced (let us say divided by ten).

Log-barrier methods

and Lagrange multiplier estimates

The log-barrier function for the (CO) problem

$$(CO) \quad \min f(x) \\ \text{s.t.} \quad g_j(x) \leq 0, \quad j = 1, \dots, m$$

for $\mu > 0$ is given by

$$f_\mu(x) = f(x) + \mu \sum_{j=1}^m -\log(-g_j(x))$$

The optimality condition when minimizing $f_\mu(x)$ is:

$$\nabla f(x) + \sum_{j=1}^m \frac{\mu}{-g_j(x)} \nabla g_j(x) = 0. \quad (*)$$

On the other hand, the Lagrange function for (CO)

$$L(x, y) = f(x) + \sum_{j=1}^m y_j g_j(x).$$

In the Wolfe dual we get the constraint:

$$\nabla f(x) + \sum_{j=1}^m y_j \nabla g_j(x) = 0, \quad (**)$$

the optimality condition to minimize $L(x, y)$ in x . Comparing $(*)$ and $(**)$ we have that

$$\frac{\mu}{-g_j(x)} \quad \text{is an estimate of} \quad y_j,$$

the Lagrange multiplier.

Penalty functions

How can one force the equality constraints $t = 0$ (i.e., $h_i(x) = 0$) and the inequality constraints $t \leq 0$ (i.e., $g_j(x) \leq 0$) by penalizing the non-satisfaction of these constraints?

What are the desirable properties of a penalty function?

Desired properties of a penalty functions $P(t)$.

1. $P(t)$ is nonnegative and strictly convex;
2. $P(t) = 0$ for feasible points;
3. $P(t)$ goes to infinity as infeasibility increases;
4. $P(t)$ increases sharply as infeasibility occurs;
5. $P(t)$ is a smooth (infinitely many times) differentiable.

Functions satisfying at least the first three properties are called

penalty functions.

Penalty functions

For the equality constraints $t = 0$ (i.e., $h_i(x) = 0$)

- Quadratic penalty function: $P(t) = t^2$.
- Exact penalty function: $P(t) = |t|$.

For the inequality constraints $t \leq 0$ (i.e., $g_j(x) \leq 0$)

1. Quadratic penalty function:

$$P(t) = \begin{cases} 0 & \text{if } t \leq 0; \\ t^2 & \text{if } t > 0; \end{cases} = (\max\{0, t\})^2.$$

2. Exact penalty function: $P(t) = \max\{0, t\}$.

$$\begin{aligned} (CO) \quad & \min f(x) \\ & \text{s.t. } g_j(x) \leq 0, \quad j = 1, \dots, m, \\ & \quad h_i(x) = 0, \quad i = 1, \dots, k. \end{aligned}$$

The quadratic penalty function for (CO) is:

$$P(x) = f(x) + \vartheta \left(\sum_{j=1}^m (\max\{0, g_j(x)\})^2 + \sum_{i=1}^k (h_i(x))^2 \right).$$

The exact penalty function for (CO) is:

$$P(x) = f(x) + \vartheta \left(\sum_{j=1}^m \max\{0, g_j(x)\} + \sum_{i=1}^k |h_i(x)| \right).$$
