ADDIS ABABA UNIVERSITY
FACULTY OF SCIENCE
MATHEMATICS DEPARTIMENT
GRADAUTE PROGRAM

SEMINAR II
ON

ACTIVE SET METHOD FOR SOLVING QUADRATIC PROGRAMMING

Prepared by: Diriba Kajela
Submitted to: Dr. Berihanu Guta

June 2010

Table of Contents

Title Page

1. Introduction........1
1.1. Basic Terminologies........3
2. Nature of quadratic programming problem........5
2.1 Equality- constrained quadratic programming Problem........5
2.2 Optimality conditions for quadratic programming Problem........9
3. Active set Method For quadratic programming Problem........11
3.1 Active set method for convex quadratic Programming problem........11
3.1.1 Finding a starting Feasible Solution........13
3.1.2 Direction Finding........14
3.1.3 Step Length Calculation........16
3.1.4 Adjustment to the Active Set........17
3.1.5 Complete Active Set Quadratic programming Algorithm........21
3.2 Active set method for non convex quadratic Programming problem........26
4. Active Set Method for Dual quadratic Programming problem........31
4.1 Conjugate Gradient Method........32
4.2 Optimality Conditions........33
4.3 Conjugate Gradient Algorithm for Quadratic Functions with Non Negative Constraints........34
5. References........37

ACKNOWLEDGMENT

First of all I would like to tanks my **God** which help me in all my ways from the beginning to the end.

My special tanks go to my adviser Dr. Behihanu Guta for his help, suggestion and comments. I am also glad to thanks to my friends Mis. Leta Bekere and Mis. Kefyalew Hailu for their suggestions and material support.

It is a great pleasure intended to acknowledge all my families, specially my wife W/ro Tejitu Naggese who helped me though the study of graduate program.

Last but not least my special thanks go to Prof. R. Deumlich who gives me additional idea how to organize the paper and basic knowledge on mathematical course and also help me to write programming for the algorithm of Active set method for solving quadratic programming problem.

1. Introduction:

This paper focuses on presenting Active Set Method for Quadratic Programming. The method starts by computing a feasible initial iterate x_0 and find a step from one iterate to the next by solving those equality constraints and inequality constraints which can act as equality at the point x_0.

An optimization problem in which the objective function is quadratic and the constraints are linear is called quadratic programming. "Programming" in this sense does not imply computer programming. It is an old word for "Planning".

Several important practical problems may be directly be formulated as quadratic programming problem. Quadratic programming problems are also arising as a result of approximating more general non-linear problems by linear and quadratic functions. Since quadratic programming problems have linear constraints, there solution methods are relatively simple extensions of linear programming problems.

There are several classes of algorithms for solving quadratic programming problems that contains inequality constraints and possibly equality constraints. Among these methods active set method, gradient projection method and interior point method can be mentioned.

Active set method can be applied both to convex and non convex problems and they have been the most widely used methods since 1970's.If the Hessian matrix associated to the twice differentiable quadratic objective function is positive semi definite then the quadratic programming problem is convex, and in this case the problem is sometimes not much difficult to solve than a linear problem. Non convex quadratic problems in which the Hessian matrix is an indefinite are more challenging, since they can have several stationary points and local minima.

Gradient projection method s attempt to accelerate the solution process by allowing rapid changes in the active set and are the most when the only constraints in the problem are bounded on the variables.

Interior point methods have recently been shown to be effective for solving large convex quadratic problems.

The active set method strategy was created in 1961 by Theil and van de Panne to solve the quadratic problems.

Active set methods for quadratic programming come in three varieties known as

- Primal
- Dual and
- Primal dual

Among these only the first two primal active set methods which generates iterates that remain feasible with respect to the primal problem and the dual active set method in which finding solution for which the Hessian matrix associated to the active function is invertible was discussed in this paper.

Generally in this paper basic terminologies which are core points to understand the active set set method algorithm was discussed under introduction part. In chapter two basic preliminary concepts used to understand quadratic programming problem such as nature of quadratic programming, equality-constrained quadratic programming, optimality conditions and certain basic theorem was discussed. The fourth chapter of this paper was the main body of this paper which contains active set method for solving quadratic programming which has sub division active set method for convex quadratic programming and for non convex quadratic programming. The last chapter was discuss about active set method for dual quadratic programming and use them to derive the steps to wards the feasible solution and the optimal solution.

The basic ideas of this method are:

- Finding the starting feasible solution
- Determine the direction in which the given function is improved.
- Compute an appropriate step Length along the direction so that the next points maintain feasibility.

Basic Terminologies

- The function given to be minimized or maximized is called an objective function. That is:

$$(P) \quad \min(or \max) \;\; f(x) \to x \in S.$$

Here the set S is called feasible set and each point $x \in S$ is called feasible solution or feasible point.

- Let f(x) be an objective function and S be its feasible set $x^* \in S$ is called optimal solution for minimization problem if $f(x^*) \leq f(x) \; \forall x \in S$.

- The set of conditions (equations and inequities) given to the objective functions called constraints.

- The $n \times n$ matrix Q = $\begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & - - - & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n x_1} & - - - & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}$ is called the Hessian matrix associated to twice differentiable function f.

- Let A be nxn Symmetric Matrix and o$\neq$ x$\in$ R^n, then A is called Positive semi definite if $x^T A x \geq 0$

- Let A be an (m,n) matrix and let Q be a symmetrical positive semi definite (n,n) matrix. Further more let $c \in R^n$ *and* $b \in R^m$. Then the optimization problem:

$$f(x) = c^T x + \frac{1}{2} x^T Q x \rightarrow \min,\ x \in S$$
$$S = \{x \in R^n \,/\, Ax \leq b, x \geq 0\}$$

is called a quadratic optimization problem.

- Let $M \subseteq R^n$ be a convex set and $f: M \rightarrow R$. Then the function f is said to be convex if and only if for all $x, y \in M$

$$f(\lambda x + (1-\lambda) y) \leq \lambda f(x) + (1-\lambda) f(y), \forall \lambda \in [0,1]$$

- Equality-constrained quadratic programming problem is a quadratic programming problem in which all the constraints are equations.

- LICQ-mean Linear Independent Constraint Qualification.

- LICQ occurs when active set constraint gradients is linearly independent

- Rank of a matrix is equal to the size or dimension of the largest non-singular square sub matrix that can be found by deleting any rows and columns of the given matrix.

- Let A be (n, n) matrix. A number λ is said to be an eigenvalue of A if and only if there is a vector x $\neq 0$ such that $Ax = \lambda x$

- A matrix of full rank is a square matrix whose determinant is non zero.

2. Nature of Quadratic programming problems

To apply an active set method for solving quadratic programming problems, first we have to understand the nature of quadratic programming problems and give necessary and sufficient conditions to be certain point is an optimal solution of the given problem.

Quadratic programming problems can be appearing in one of the following forms:

- Unconstrained quadratic programming problem
- Equality-constrained quadratic programming problem
- Inequality-constrained quadratic programming problem
- Both equality-constrained and inequality-constrained quadratic programming problems

Even though the focus of this paper was solving quadratic programming problems which contain both equalities ad inequalities constraints by using active set method, to develop techniques how to apply the algorithms of active set method we will discuss how to solve equality –constrained quadratic programming problem and lastly discuss unconstrained quadratic programming problem under dual quadratic programming problem

2.1. Equality–constrained quadratic programming problem

Consider the quadratic programming problem of the form

Minimize $$q(x) = c^T x + \frac{1}{2} x^T Q \; x$$

Subjected to $$a_i^T x = b_i \qquad , i \in E \qquad [1]$$

Where Q is $n \times n$ matrix of positive semi definite.

E is finite set of index

c, x and $(a_i), i \in E$. are vectors with $n\times 1$ elements

b is an $m \times 1$ vector of right hand side.

Let $(a_i)^T, i \in E = A$ and assume that A has full row rank and the constraints are consistent.

The first order necessary condition for x* to be a solution of equation [1] state that there is a vector λ such that the following system of equations is satisfied.

$$\begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix} \qquad [2]$$

Where λ^* is a vector of Lagrange multipliers.

The system in [2] can be written in the form that is useful for computation by expressing x* as x* = x+ d, where x is some estimate of the solution and d is the desired step. By introducing this notation and rearranging the equation we obtain:

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -d \\ \lambda^* \end{bmatrix} = \begin{bmatrix} g \\ k \end{bmatrix} \qquad [3]$$

Where $k = Ax - b$

$g = c + Qx$

$d = x^* - x$

The matrix in [3] is the Karush-Kuhn-Tucker (KKT) matrix and the following result gives condition under which it is non singular. We use Z to denote the n ×(n-m) matrix whose columns are a basis for the null space of A. That is, Z has full rank and AZ = 0.

THEOREM 1

Let A has full row rank, and assume that the reduced Hessian matrix $Z^TQ\,Z$ is positive definite. Then the KKT matrix

$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$ is non singular and there is a unique vector pair (x^* ,λ^*) satisfying the system of equation [2] above.

Proof

Suppose there are vectors d and v such that

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d \\ v \end{bmatrix} = 0$$

Since $Ad = 0$ we have that

$$o = \begin{bmatrix} d \\ v \end{bmatrix}^T \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d \\ v \end{bmatrix} = d^T Q\, d$$

Since d lies in the full space of A, it can be written as $d = Zu$ for some vector $u \in R^{n-m}$

This implies, $0 = d^T Qd = u^T Z^T QZu$

By positive definiteness of Z^TQZ implies that $u = 0$

Therefore $d = 0$ *and* $A^T v = 0$ and also the full rank of A implies that v=0

$$\Rightarrow \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d \\ v \end{bmatrix} = 0$$ is satisfied only if $d = 0$ *and* $v = 0$.

Therefore the matrix $\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$ *is non singular.*

THEOREM 2

Suppose that the conditions of **theorem 1** are satisfied .Then the vector x* satisfying equation [2] is the unique global solution of equation [1].

Proof

Let **x** be any other feasible point (satisfying $Ax=b$) and $d=x^*-x$

Since Ax* = Ax = b, we have that Ad = 0

By substituting in to the objective function of [1] we obtain:

$$q(x)=c^T(x^*-d)+\frac{1}{2}(x^*-d)^T Q(x^*-d)$$

$$=-c^T d+\frac{1}{2}d^T Qd-d^T Qx^*+q(x^*) \qquad [4]$$

From [2] we have that $Qx^*=-c+A^T\lambda^*$. So from Ad= 0 we have that $d^T Qx^*=d^T(-c+A^T\lambda^*)=-d^T c$

Substituting this equation in to [4] we obtain:

$$q(x)=\frac{1}{2}d^T Qd+q(x^*)$$

Since d lies in the null space of A, We can write $p=Zu$ for some vector $u\in R^{nx(n-m)}$, so that

$$q(x)=\frac{1}{2}u^T Z^T Q Zu+q(x^*)$$

By positive definiteness of $Z^T QZ$, we conclude that $q(x)>q(x^*)$ except when $u=0$ that is, when x = x*.

Therefore x* is the unique global solution of equation [1]

2.2 Optimality Conditions for Quadratic Programming Problems

Consider the Quadratic programming problems of the form:

Minimize $\quad q(x) = c^T x + \frac{1}{2} x^T Q\ x$

Subjected to: $\quad a_i^T x = 0 \qquad , i \in E \qquad$ [1]

$$a_i^T x \geq 0 \qquad , i \in I$$

The Lagrange for the constrained optimization of this problem is defined:

$$L(x,\lambda) = q(x) + \sum_{i \in E \cup I} \lambda_i a_i x$$

The active set A(x) at any feasible x is the union of set E with the indicates of the active inequality constraints, i.e.

$$A(x) = E \cup \{\ i \in I / a_i x = 0\ \}$$

Definition: Given the point x* and the active set A(x*), we say that the linear

independence constraint qualification (LICQ) holds if the active set constraint gradients

$\{\nabla a_i(x^*), i \in A(x^*)\ \}$ is linearly independent.

Theorem (First order necessary conditions)

Suppose that x* is a local solution of (*) and that the LICQ holds at x*. Then there is a

Lagrange multiplier vector λ^* with components $\lambda_i^*, i \in E \cup I$ such that the

following conditions are satisfied at (x^*, λ^*).

$$\nabla_x L(x^*, \lambda^*) = 0$$

$$a_i(x^*) = 0 \ , \forall i \in E$$

$$a_i(x^*) \geq 0 , \forall \ i \in I$$

$$\lambda_i^* \geq 0 \ , \forall i \in I$$

$$\lambda_i^* a_i(x^*) = 0 \ , \forall i \in E \cup I$$

These conditions are known as the Karush- Kuhn – Tucker condition.

Because the complementary condition implies that the Lagrange multipliers corresponding to in active inequality constraints are zero, we can omit the terms for indices $i \notin A(x^*)$ from above and rewrite the conditions as:

$$0 = \nabla_x L(x^*, \lambda^*) = \nabla q(x^*) + \sum_{i \in A(x^*)} \lambda_i^* \nabla a_i(x^*)$$

Theorem (Second- order necessary Condition)

If x* is a local minimize of f and $\nabla^2 f$ is continuous in an open neighborhood of x*, Then $\nabla f(x^*) = 0$ *and* $\nabla^2 f(x^*)$ is positive definite.

Theorem (Second- order Sufficient Condition)

Suppose that $\nabla^2 f$ is continuous in an open neighborhood of x* and that $\nabla f(x^*) = 0$ *and* $\nabla^2 f(x^*)$ is positive definite. Then x* is a local minimizer of f.

3 Active Set Method for Solving Quadratic Programming Problem

3.1. Active Set method for solving convex quadratic programming problem

If the Hessian matrix Q is positive semi definite, we say the given quadratic programming is convex quadratic programming and also if the Hessian matrix Q is indefinite, the quadratic programming is non convex quadratic programming problem which was discussed in the last section. Quadratic problems can be solved in a finite number of methods, but the effort required to find a solution depend strongly on the characteristics of the objective function and the number of constraints. One of these methods is primal active set method used to find feasible solution and optimal solution of the primal Quadratic programming problem.

To discuss the method consider the following quadratic programming (QP) of the form:

Minimize $$q(x) = c^T x + \frac{1}{2} x^T Q\ x$$

Subjected to: $$a_i^T x = b_i \quad , i \in E \qquad [1]$$

$$a_i^T x \geq b_i \quad , i \in I$$

Where Q is symmetric $n \times n$ matrix of positive semi definite.

E and I are finite set of induces

c and x are vectors with nx1 elements

$(a_i), i \in E \cup I$ is an $m \times n$ matrix and

b is an $m \times 1$ vector of right hand side.

Definition3.1 The active set A(x) at any feasible point x is the union of equality constraints E with the indices of the active inequality constraints. Which means

$$A(x) = E \cup \{ i \in I \,/\, a_i^T x = b_i \}$$

The active set $A(x^*)$ at the optimal point x^* is called an optimal active set, $A(x^*)$.

Definition3.2 Working set is a set contains induces of linearly independent constraints which are active at the given point x.

Under this condition the general quadratic programming in equation [1] above can be restated as:

$$\text{Minimize} \quad q(x) = c^T x + \frac{1}{2} x^T Q\, x$$

$$\text{Subject to} \quad a_i^T x = b_i\,, \quad i \in A(x^*) \qquad [2]$$

How ever since the above form is similar to the general non linear Optimization problem it is more convenient to use it directly for direction finding.

Active set method for quadratic programming is differ from simplex method in that the iteration may not move from one vertex of the feasible region to other. Some iterates (the solution of the problem) may lie at other points on the boundary or interior of the feasible region.

Active set method usually starts by computing a feasible initial iterate $\mathbf{x_0}$, and then ensure that all subsequent iterates remain feasible. They find a step from one iterate to the next by solving a quadratic sub problem in which a subset of the constrains, $a_i^T x = b_i,\ i \in E$ *and* $a_i^T x > b_i,\ i \in I$ is imposed as equalities. This subset referred to as the working set denoted by W_k.

Working set consists of all the equality constraints $i \in E$ together with some (but not necessarily all) of the active inequality constraints.

An important requirement we impose on a working set W_k is that the gradients a_i of the constraints in the working set be linearly independent even when the full set of active constraints at the point has linearly dependant gradients.

When the initial active constraints have independent gradients we can include them all in W_0. Alternatively, we can select a subset so we have flexibility in the choice of the initial working set, and that each initial choice leads to a different iteration sequence. Even if the initial working set W_0 coincides with the initial active set, the sets W_k and $A(x_k)$ may differ at later iterations. For instance, when a particular step encounters more than one blocking constraints, just one of them is added to the working set, so the identification between W_k and $A(x_k)$ is broken.

3.1.1 Finding a starting feasible solution

To start the active set method for solving the quadratic programming problem we have to have a starting feasible point x_0. This starting feasible solution can found by standard phase I of simplex procedure as stated below.

By introducing an artificial variable for each constraint $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$. The **phase I** problem is started as follows:

Minimize $$\phi = x_{n+1} + x_{n+2} + \ldots\ldots\ldots\ldots$$

Subject to
$$a_i^T x + x_{n+i} = b_i \ , i \in E$$
$$a_i^T x + x_{n+i} \le b_i \ , i \in I$$

The solution of this problem is when $\phi = 0$, giving us a point that is feasible for the original problem,

After getting a starting feasible solution x_0, a very important issue is that if x_0 is not an optimal solution in which direction we have to move in the feasible region to get the better feasible solution that improves the optimal value. This leads us to the process of direction finding

3.1.2 Direction Finding

At the current point x_0 some of the inequality constraints may be satisfied as equalities. These constraints together with the equality constraints constitute the active set or working set(when the gradient vectors are linearly independent). The set of active constraint indices can therefore be defined as follows:

$$A_k = E \cup \{ i / a_i^T x_k = b_i \quad , i \in I\}$$

Given an iterate x_k and the working set W_k, we first check whether x_k minimizes the quadratic function $q(x)$ in the sub space defined in the working set. If not, we compute a step d_k by solving an equality constrained quadratic programming sub problem in which the constraints corresponding to the working set W_k are regarded as equalities and all other constraints are temporary disregarded. To express this sub problem interims of the step d, we define:

$$x = x_k + d_k$$

Then by using equation [2] above, the direction finding problem can therefore be stated as follows:

$$\text{Minimize} \quad c^T(x_k + d_k) + \frac{1}{2}(x_k + d_k)^T Q(x_k + d_k)$$

$$\text{Subject to} \quad a_i^T (x_k + d_k) = b_i, \qquad i \in w_k$$

Then by expanding and ignoring the constant terms in the objective function, we have

$$\text{Minimize} \quad [\ Q x_k + c\ ]^T d_k + \frac{1}{2} d_k^T Q d_k + constant$$

$$\text{Subject to} \quad a_i^T d_k = b_i - a_i^T x_k = 0, \qquad i \in w_k$$

Now by introducing the notion for these expressions as:

$$g_k = Qx_k + c \quad and \quad A_k = \begin{pmatrix} a_1^T \\ a_2^T \\ \cdot \\ \cdot \end{pmatrix}$$

Here A_k is those rows of A which are in the working set.

Since we can drop the constants from the objective function with out changing the solution of the problem we have the following equality constrained quadratic programming sub problem to be solved at the k^{th} iteration as follows:

$$\text{Minimize} \qquad g_k^T d_k + \frac{1}{2} d_k^T Q d_k \qquad [3]$$

$$\text{Subject to} \qquad A_k d_k = 0$$

The Kuhn Tucker conditions for the minimization of this problem give the following system of equations:

$$Qd_k + g_k + A_k^T v = 0$$

$$A_k d_k = 0$$

Where v is a vector of Lagrange multipliers associated with active constraints.

Both equations can be written in a matrix form, as follows

$$\begin{pmatrix} Q & A_k^T \\ A_k & 0 \end{pmatrix} \begin{pmatrix} d_k \\ v \end{pmatrix} = \begin{pmatrix} -g_k \\ 0 \end{pmatrix}$$

From this the direction d_k together with the lag range multipliers v can be calculated by solving the above linear system of equations.

3.1.3 Step length calculation

After computing the step direction d_k, the step length α is selected to be as large as possible to maintain feasibility with respect to the inactive inequality constraints.

$$a_i^T(x_k + \alpha d_k) \geq b_i, \quad i \notin w_k$$

Note that for each $i \in w_k$; the term $a_i^T x_k$ does not change as we move along d_k, sine we have $a_i^T(x_k + d_k) = a_i^T x_k = b_i$, It follows that since the constraints in W_k were satisfied at x_k, they are also satisfied at $x_k + \alpha d_k$ for any value of α.

Suppose the optimal d_k for equation [3] above is non zero, we need to decide how far to move along this direction to get the improved value for the given objective function with in the feasible region. If $x_k + d_k$ is feasible with respect to all the constraints, we set

$$x_{k+1} = x_k + d_k$$

Other wise, we set $x_{k+1} = x_k + \alpha_k d_k$

Where the step length parameter α_k is chosen to be the largest value in the range (0,1] for which all constraints are satisfied. We can derive an explicit definition of α_k by considering what happens to the constraints $i \notin W_k$, since the constraints $i \in w_k$ will certainly be satisfied regardless of the choice of α_k.

If $a_i^T d_k \geq 0$ for some $i \notin w_k$ then for all $\alpha_k \geq 0$ we have $a_i^T(x_k + \alpha_k d_k) \geq a_i^T d_k \geq b_i$.hence, this constraint will be satisfied for all non negative choice of the step length parameter.

When ever $a_i^T d_k < 0$ for some $i \notin W_k$, however, we have that $a_i^T(x_k + \alpha_k d_k) \geq b_i$.

Only if, $\alpha_k \leq \dfrac{b_i - a_i^T x_k}{a_i^T d_k}$ $for\ a_i^T d_k < o, i \notin W_k$

Since we want α_k to be as large as possible in (0, 1] subject to retaining feasibility. We have to select the step length parameter α_k as follows:

$$\alpha_k = \min\left(1, \min \frac{b_i - a_i^T x_k}{a_i^T d_k}\ for\ a_i^T d_k < 0\ and\ i \notin w_k\right) \quad [4]$$

3.1.4 Adjustments to the active set

We call the constraint i for which $\alpha < 1$ in equation [4] is achieved is called the blocking constraints.

If $\alpha_k = 1$ and no new constraints are active at $x_{k+1} = x_k + d_k$, then there are no blocking constraints on the iteration. We update $x_{k+1} = x_k + \alpha_k d_k$ but do not change the working set $W_{k+1} = W_k$.

If $\alpha_k < 1$, that is, the step along d_k was blocked by some constraint(s) not in w_k, a new working set w_{k+1} is constructed by adding one of the blocking constraints to w_k.

We continue to iterate in this manner, adding constraints to the working set until we reach a point $\hat{x}$ that minimizes the quadratic objective function over it's current working set $\hat{w}$.

Such a point can be recognized because the sub problem [3] above has a solution $d_k = o$

Since $d_k = o$ satisfies Optimality condition we have

$$\sum a_i \lambda_i = g_k = Q\hat{x} + c \qquad [5]$$

for some Lagrange multipliers λ_i, $i \in w$.

Since the Lagrange multipliers for all inequality constraints must be positive, if an element in the v vector corresponding to an inequality constraint in the active set is negative, this means that the constraint can not be active and must be dropped from the active set. If there are several active inequalities with negative multiplier, usually the constraint with the most negative multiplier is dropped from the active set.

THEOREM 1

Suppose that the point $\hat{x}$ Satisfies first order conditions for the equality constrained sub problem with working set $\hat{w}$; that is equation [5] is satisfied along with $a_i^T x = b_i \; \forall i \in \hat{w}$ such that $\lambda_j < 0$.Finally let d be the solution obtained by dropping the constraint j and solving the following sub problem.

Minimize $\quad \frac{1}{2} d^T Q d + (Qx + c)^T d$

Subject to $\quad a_i^T d = 0 \; \forall \; i \in W \; with \; i \neq j \qquad [1]$

Then d is a feasible direction for constraint j, that is $a_j^T d \geq 0$. More over, if d satisfies second order sufficient condition for this sub problem, then we have that $a_j^T d > 0$ and d is descent direction for $q(.)$.

Proof

Since d solves equation [1] we have that there are multipliers $\hat{\lambda}_i$, for all i $\in \hat{W}$ with i$\neq$ j, such that

$$\sum_{i\in W, i\neq j} \lambda_j a_j - = Q\,d + (Qd + c) \qquad [2]$$

In addition by second –order necessary condition if Z is the null space basis vector for the matrix $(a_i)T$ i $\in \hat{W}$ with i≠ j.

Then $Z^T Q \; Z$ is positive semi definite

$\Rightarrow$ d has the $\lambda_j a_j^T d = d^T Q d$ [3]

Form d = Z d_z for some vector d_z

$\Rightarrow \quad d^T Q \; d \;\geq\; 0$

We have that the assumption that $\hat{x}$ and $\hat{W}$ satisfies the relation [5] .By subtracting equation [5] from the equation [1] we obtain:

$$\sum_{i\in W, i\neq j} (\lambda_i - \lambda_j) a_i - \lambda_j a_j = Q\,d \qquad [4]$$

By taking inner products of both sides with p and using the fact that $a_i^T d = 0$ for all $i \in W$ with $i \neq j$ we have that

$$-\lambda_j a_j^T d = d^T Q d \qquad [5]$$

Since $d^T Qd \geq and \, \lambda_j < 0$ by assumption, it follows immediately that $a_j^T d \geq 0$

If the second –order sufficient conditions are satisfied we have that $Z^T Q Z$ defined above is positive definite.

From equation [5] we can have $a_j^T d = 0 \;\; only\, if \;\; d\,T\,Q\,d = dz^T Z^T Q\,Z\,dz$,

which happens only if $dz = 0 \;\; and \; d = 0$

But if d=0 by substituting in to equation [4] and using linear independence of a_i for $i \in W$,we must have that $\lambda_j = 0$ which contradicts our choice of j.

We conclude that $d^T Q d > 0$ in [5] and therefore $a_j^T d > 0$, when d satisfies the second order sufficient condition for [1].

Theorem 2

Suppose that the solution d_k of equation [3] is non zero and satisfies the second order sufficient conditions for optimality for the problem. then the function q(.) is strictly decreasing along the direction d_k.

Proof

Since d satisfies the second –order conditions, that is $Z^T Q Z$ is positive definite for the matrix Z whose columns are a basis foe the null space of the constraints $a_i^T d_k = 0, \forall i \in W_k$

We have by applying theorem 2 above to equation [3] that d_k is the unique global solution of equation [3]

Since d = 0 is also a feasible point for equation [3] it's objective value must be larger than that of d_k, so we have :

$$\frac{1}{2} d_k^T Q d_k + g_k^T d_k < 0.$$

Since $d_k^T Q d_k \ge 0$ by convexity

$q(x_k + \alpha_k d_k) = q(x_k) + \alpha_k g_k d_k + \frac{1}{2}\alpha^2 d_k^T Q d_k < q(x_k)$ by this inequality implies that $g_k^T d_k < 0$

Therefore we have $q(x_k + \alpha_k d_k) < q(x_k)$

For all $\alpha > 0$ sufficiently small.

3.1.5 Complete Active set Quadratic programming Algorithm

Initialization:

set k=0. If there are equality constraints start by setting x_o to the solution of these constraints; otherwise, set $x_{o=0}$. If this x_o satisfies inequality constraints as well, we have a starting feasible solution x_o. If not determine the starting feasible solution by using phase I procedure. Set w_o to the incidences of the constraints active at x_o.

a. From $g_k = Q x_k + c$

b. Solve the following System of equations

$$\begin{pmatrix} Q & A_k^T \\ A_k & 0 \end{pmatrix} \begin{pmatrix} d_k \\ v \end{pmatrix} = \begin{pmatrix} -g_k \\ 0 \end{pmatrix}$$

c. If $d_k = 0$, go to step f to check for optimality. Other wise, continue

d. Compute step Length

$$\alpha_k = \min\left(1, \min \frac{b_i - a_i^T x_k}{a_i^T d_k} \ for\ a_i^T d_k < 0\ and\ i \notin w_k \right)$$

If α<1, add the constraint controlling the step length to active set.

e. Update

$x_{k+1} = x_k + \alpha d_k$, and go to seep a

f. Check the sign of Lagrange multipliers corresponding to inequality constraints. If they are all positive, sop. We have the optimum. Otherwise, remove the constraint that corresponds to the most negative multiplier and go to step b.

Example

Solve the following Quadratic Programming Problem by using Active Set Method

Minimize $f = x_1^2 + x_1x_2 + x_3^2 + 2x_2x_3 + 4x_1 + 6x_2 + 12x_3$

Subject to
$$-x_1 - x_2 - x3 \le -6$$
$$x_1 + x_2 - 2x_3 \le -2$$
$$-x_1 \le 0$$
$$-x_3 \le 0$$

Solution

$$Q = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & - - - & \frac{\partial^2 f(x)}{\partial x_1 \partial x_3} \\ , & & , \\ , & & , \\ , & & , \\ \frac{\partial^2 f(x)}{\partial x_3 x_1} & - - - & \frac{\partial^2 f(x)}{\partial x_3^2} \end{pmatrix} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & 2 & 4 \end{pmatrix}$$

$$c = \begin{pmatrix} 4 \\ 6 \\ 12 \end{pmatrix}, \quad A = \begin{pmatrix} -1 & -1 & -1 \\ 1 & 1 & -1 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} -6 \\ -2 \\ 0 \\ 0 \end{pmatrix}$$

To find the initial starting point x_0 by using phase I of simplex procedure we get

$$x_o = \left(\frac{10}{3}, o, \frac{8}{3}\right)$$

*** Iteration 1 ***

$x_o = \left(\frac{10}{3}, o, \frac{8}{3}\right)$ Active constraints at x_o are 1 and 2. This implies $W_o = \{1, 2\}$

$A_k = \begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & -2 \end{pmatrix}$ Where A_k is composed of those rows A which are in the working set.

Now $g_k = Qx_o + c = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & 2 & 4 \end{pmatrix} \begin{pmatrix} \frac{10}{3} \\ 0 \\ \frac{8}{3} \end{pmatrix} + \begin{pmatrix} 4 \\ 6 \\ 12 \end{pmatrix} = \begin{pmatrix} \frac{20}{3} \\ \frac{26}{3} \\ \frac{32}{3} \end{pmatrix} + \begin{pmatrix} 4 \\ 6 \\ 12 \end{pmatrix} = \begin{pmatrix} \frac{32}{3} \\ \frac{44}{3} \\ \frac{68}{3} \end{pmatrix}$

$$\begin{pmatrix} Q & A_k^T \\ A_k & 0 \end{pmatrix} \begin{pmatrix} d_k \\ v \end{pmatrix} = \begin{pmatrix} -g_k \\ 0 \end{pmatrix} \quad , d_k = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}, v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} 2 & 1 & 0 & -1 & 1 \\ 1 & 4 & 2 & -1 & 1 \\ 0 & 2 & 4 & -1 & -2 \\ -1 & -1 & -1 & 0 & 0 \\ 1 & 1 & -2 & 0 & 0 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \frac{32}{3} \\ \frac{44}{3} \\ \frac{68}{3} \\ 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} 2d_1 + d_2 - v_1 + v_2 = \frac{32}{3} \\ d_1 + 4d_2 + 2d_3 - v_1 + v_2 = \frac{44}{3} \\ 2d_2 + 4d_3 - v_1 - 2v_2 = \frac{68}{3} \\ -d_1 - d_2 - d_3 = 0 \\ d_1 + d_2 - 2d_3 = 0 \end{cases}$$

From these systems of equations we get:

$$d_1 = 1,\ d_2 = -1,\ d_3 = 0$$

$$Thefore\ \ d_k = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$$

Inactive constraints are 3 and 4

$$A_i = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix},\ i \notin W_k \qquad b_i = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$b_i - A_i x_o = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \frac{10}{3} \\ \frac{8}{3} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} \frac{-10}{3} \\ \frac{-8}{3} \end{pmatrix} = \begin{pmatrix} \frac{10}{3} \\ \frac{8}{3} \end{pmatrix}$$

$$b_i - A_i\ x_0\ (i \notin W_k) \rightarrow (\frac{10}{3}\ \ \frac{8}{3})$$

$$A_i d_k = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

$$A_i\ d_k (\ i \notin W_k\ ) \rightarrow (\ -1\ \ 0\ )$$

By component wise division we get:

$$\Rightarrow \quad \frac{b_i - A_i^T x_o}{A_i^T d_k} = (\infty \quad \infty)$$

$$\Rightarrow \quad The\ step\ length\ \alpha = 1\ and\ \ W_1 = \{1, 2\}$$

$$\Rightarrow x_1 = x_o + \alpha\ d_k = \begin{pmatrix} \frac{10}{3} \\ 0 \\ \frac{8}{3} \end{pmatrix} + 1 \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{13}{3} \\ -1 \\ \frac{8}{3} \end{pmatrix}$$

**** Iteration 2 ****

$$x_1 = \begin{pmatrix} \frac{13}{3} \\ -1 \\ \frac{8}{3} \end{pmatrix} \quad , \quad W_1 = \{1,2\} \quad , \quad A_k = \begin{pmatrix} -1 & -1 & -1 \\ 1 & 1 & -2 \end{pmatrix}$$

$$g_k = Q\,x_1 + c = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & 2 & 4 \end{pmatrix} \begin{pmatrix} \frac{13}{3} \\ -1 \\ \frac{8}{3} \end{pmatrix} + \begin{pmatrix} 4 \\ 6 \\ 12 \end{pmatrix} = \begin{pmatrix} \frac{23}{3} \\ \frac{17}{3} \\ \frac{26}{3} \end{pmatrix} + \begin{pmatrix} 4 \\ 6 \\ 12 \end{pmatrix} = \begin{pmatrix} \frac{35}{3} \\ \frac{35}{3} \\ \frac{62}{3} \end{pmatrix}$$

Inequality constraint values at the current point becomes $\rightarrow \begin{pmatrix} 0 & 0 & \frac{-13}{3} & \frac{-8}{3} \end{pmatrix}$

$$\begin{pmatrix} Q & A_k^{\,T} \\ A_k & 0 \end{pmatrix} \begin{pmatrix} d_k \\ v \end{pmatrix} = \begin{pmatrix} -g_k \\ 0 \end{pmatrix} \quad , d_k = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}, v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} 2 & 1 & 0 & -1 & 1 \\ 1 & 4 & 2 & -1 & 1 \\ 0 & 2 & 4 & -1 & -2 \\ -1 & -1 & -1 & 0 & 0 \\ 1 & 1 & -2 & 0 & 0 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \frac{-35}{3} \\ \frac{-35}{3} \\ \frac{-62}{3} \\ 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} 2d_1 + d_2 - v_1 + v_2 = \dfrac{-35}{3} \\ d_1 + 4d_2 + 2d_3 - v_1 + v_2 = \dfrac{-35}{3} \\ 2d_2 + 4d_3 - v_1 - 2v_2 = \dfrac{-62}{3} \\ -d_1 - d_2 - d_3 = 0 \\ d_1 + d_2 - 2\,d_3 = 0 \end{cases}$$

From this system of equations we get

$$d_1 = 0\,, d_2 = 0\,, d_3 = 0\,, v_1 = \frac{44}{3} \;\; and \;\; v_2 = 3$$

$$Therefore \;\; d_k = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \;, v = \begin{pmatrix} \frac{44}{3} \\ 3 \end{pmatrix}$$

Therefore $d_k = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ and all the Lagrange Multipliers $v_1 = \frac{44}{3}$ *and* $v_2 = 3$ are positive we get an optimal solution $x_1 = \begin{pmatrix} \frac{13}{3} \\ -1 \\ \frac{8}{3} \end{pmatrix}$ with the minimal value of the objective function $f(x_1) = \frac{206}{3}$

3.2 Active Set Method for Indefinite Quadratic Programming

we now consider the case in which the Hessian matrix Q has some negative eigen values. Algorithm set under (3.1.5) the active set method for convex quadratic programming can be adapted to this indefinite case by modifying the computation of the search direction in certain situations.

To explain the need for the modification we consider the computation of a step by a null space method that is $d = Z d_z$ when d_z is given by

$$(z^T Q Z)dz = -z^T g \quad for \quad g = c + Qx$$

If the reduced Hessian $Z^T QZ$ is positive definite them the step d points to the minimize of the sub problem (3) under chapter [3.1.2] and the logic of the iteration need not be changed. If $z^T Q z$ has negative values, however, d points only to a saddle point of (3) and therefore not a suitable step. In stead, we seek an alternative direction S_z that is a direction of negative curvature of $Z^T QZ$ we then have that

$$q\left(x+\alpha z s_z\right) \rightarrow -\infty \ as \ x \rightarrow \infty$$

Additionally we can choose the sign of Sz so that $Z s_z$ is a non ascent direction for q at the current point x, that is, $\nabla q(x)^T Z s_z \leq 0$. By moving along the direction $Z s_z$ we will encounter a constraint that can then be added to the working set for the next iteration.

If we do not find such a constraint, the problem is unbounded.

If the reduced Hessian matrix for the new working set is not positive definite we can repeat this process until enough constraints have been to make the reduced Hessian positive definite.

A difficult with this general approach, however, is that if the allow the reduced Hessian to have several negative eigen values, we need to compute it's spectral factorization or symmetric indefinite factorization in order to obtain appropriate negative curvature directions, but it is difficult to make these methods efficient when the reduced Hessian changes from one working set to the next.

Inertia controlling methods are practical class of algorithms for indefinite quadratic programming that never allow the reduced Hessian to have more than one negative Eigen values. As in the convex case there is a preliminary phase in which a feasible starting point x_0 is found. We place additional demand on x_0 that it be either a vertex in the case the reduced Hessian is the null matrix or a constrained stationary point at which the reduced Hessian is positive definite. At each iterations, the algorithm will either add or remove a constraint from the working set. If a constraint is added the reduced Hessian is of smaller dimension and must remain positive definite to be null matrix. Therefore, an indefinite reduced Hessian can arise only when one of the constraints is removed from the working set, which happens only when the current point is minimizer with respect to the current working set. In this case, we will choose the new search direction to be a direction of negative curvature for the reduced Hessian.

The are various algorithms for indefinite quadratic programming that differ in the may that indefiniteness is detected, in the computation of the negative curvature direction, and in the handling of the working set.

We now discuss an algorithm that makes use of pseudo constraints and that computes directions of negative curvature by means of LDL^T factorization. Suppose that the current reduced Hessian Z^TQZ is positive definite and that is factored as $Z^TQZ = LDL^T$, where L is unit lower triangular and D is diagonal with positive diagonal entries.

We denote the number of elements in the current working set W by t. After removing a constraint from the working set, the new null space basis can be chosen in the form $Z_+ = [Z/z]$ (that is one additional column), so that the new factors have the form.

$$L_+ = \begin{bmatrix} L & 0 \\ l^T & 1 \end{bmatrix} \quad , \quad D_+ = \begin{bmatrix} D & 0 \\ 0 & d_{n-t+1} \end{bmatrix}$$

For some vector l and element d_{n-t+1}. If we discover that d_{n-t+1} in D is negative, we know the reduced Hessian is indefinite on the manifold defined by the new working set. We can then compute the direction S_z of negative curvature for $Z^T QZ$ and corresponding direction s of negative curvature for Q is as follows:

$$L^T{}_+ Sz = e_{n-t+1},\; S = Z_+ S_z$$

We can verify that these directions have the desired properties

$$\begin{aligned} S^T QS &= S_z{}^T Z_+^T QZ_+ S_z \\ &= S_z{}^T L_+ D_+ L^T{}_+ S_2 \\ &= e^T{}_{n-t+1} D_+ e_{n-t+1} \\ &= d_{n-t+1} < 0 \end{aligned}$$

Before moving along this direction s, we will define the working set in a special way. Suppose that i is the index of the constraint that is scheduled to be removed from the working set the index whose removal causes the reduced Hessian to become indefinite and leads to the negative curvature direction s derived above. Rather than removing i explicitly from the working set as we do in algorithm under chapter (3.1.5), we leave it in the working set as a pseudo constraint

By doing so, we ensure that the reduced Hessian for this working set remains positive definite. We now move along negative curvature direction s until we encounter a constraint, and them continue to take steps in the usual manner, adding constraints until the solution of an equality constrained sub problem is found. If at this point we can safely delete the pseudo constraint i from the working set while retaining positive definiteness of the reduced Hessian, then we do so. If not, we retail it in the working set until a similar opportunity arises on a later iteration.

4. Active Set method for the Dual quadratic programming Problem

In this section, we consider a special form of quadratic programming problem in which Q is invertible. The problem involves a quadratic objective function and simple non – negative constraints on problem variables. We start this chapter with a brief discussion of duality for convex quadratic programming. Duality can be useful tool in quadratic programming because in some classes of problems we can take advantage of the special structure of the dual to solve the problem more efficiently.
If Q is positive definite, the dual of

$$\text{Minimize} \quad q(x) = c^T x + \frac{1}{2} x^T Q\ x \qquad [1]$$

$$\text{Subject to} \quad Ax \geq b$$

is given by

$$\text{Maximize} \quad q(x) = \frac{1}{2} x^T Q x + x^T c - \lambda^T (Ax - b)$$

$$\text{Subject to} \quad \begin{aligned} &Qx + c - A^T \lambda = 0 \\ &\lambda \geq o \end{aligned} \qquad [2]$$

By eliminating x from the second equation of [2] we obtain bounded unconstrained problem.

$$\text{Maximize} \quad q(x) = -\frac{1}{2} \lambda^T \left(AQ^{-1} A^T \right) \lambda + \lambda^T \left(b + AQ^{-1} c \right) - \frac{1}{2} c^T Q^{-1} c$$

$$\text{Subject to} \quad \lambda \geq 0$$

This type of quadratic problem can be solved by using varies types of algorithms. However, because of simple nature of constraints, a more efficient method can be developed by a simple extension of any of the methods for unconstrained minimization.

In this section the conjugate gradient method is extended to solve this special quadratic programming problem . So in order to use the extension of this algorithm to solve dual quadratic programming problem we have to discuss conjugate gradient method as follows.

4.1 Conjugate gradient methods

Conjugate gradient methods were proposed by **Hestenes** and **Stiefel** in 1952 for solving systems of equations. The use of this method for unconstrained optimization was prompted by the fact that the minimization of a positive definite quadratic functions is equivalent to solving the linear equation system that results when it's gradient is set at zero. Actually, the extension of conjugate gradient methods solving non linear equation system and it's use in solving general unconstrained minimization problem was first done in 1964 by Fletcher and Reeves.

The basic scheme of conjugate gradient methods for minimizing a differentiable function q (x) is to generate a sequence of iterates x_j according to

$$x_{j+1} = x_j + \lambda_j d_j$$

Where d_j steps direction

λ_j is the step length which minimizes q along d_j from the point x_j.

For j=1, the step direction $d_1 = -\nabla q(x_1)$ can be used and for subsequent iterations given x_{j+1} with $\nabla q(x_{j+1}) \neq o$ for $j \geq 1$ We use $d_{j+1} = -\nabla q(x_{j+1}) + \alpha_j d_j$

Where α_j is a suitable deflection parameter that characterizes a particular conjugate gradient method and

$$\alpha_j = \frac{\|\nabla q(x_{j+1})\|^2}{\|\nabla q(x_j)\|^2}$$

So, summery of conjugate gradient method for minimizing a general differentiable function is given below.

Initialization step

Choose a termination scalar $\varepsilon > 0$ and an initial point x_1

Let $d_1 = -\nabla q(x_1), k = j = 1$ and go to the main step

Main step

1. If $\|\nabla q(x_j)\| \leq \varepsilon$, stop. Other wise let λ_j be an optimal solution to the problem to minimize $q(x_j + \lambda d_j)$ subject to $\lambda \geq 0$ and let $x_{j+1} = x_j + \lambda_j d_j$. If $j < n$ go to step 2; other wise go to step 3

2. Let $d_{j+1} = -\nabla q(x_{j+1}) + \alpha_j d_j$

Where $\alpha_j = \dfrac{\|\nabla q(x_{j+1})\|^2}{\|\nabla q(x_j)\|^2}$

Replace j by j+1 and go to step 1

3. Let $x_1 = x_{n+1}$ and let $d_1 = -\nabla q(x_1)$. Let j = 1

Replace j by j+1 and go to step 1

4.2 Optimality conditions

By introducing a vector of slack variables s and the Lagrange multipliers, $\lambda_i \geq 0, i = 1,2......, m \ and \ \lambda_i = 0, i = m+1,....., n$

The Lagrange for the problem is as follows:

$$L(x, \lambda, s) = f(x) + \lambda^T(-x + s^2)$$

The necessary optimality conditions are :

$$\nabla f(x) - \lambda = 0$$

$$\lambda_i \geq 0, i = 1,2,....m \ and \ \lambda_i = 0, i = m+1,....., n$$

$$\lambda_i s_i = 0, i = 1,2,....m \qquad (Switchingcondtions)$$

From the switching conditions, either $\lambda_i = 0 \; or \; s_i = 0$

If $\lambda_i = 0$, then the gradient condition says that the corresponding partial derivative $\frac{\partial f}{\partial x_i}$ must be zero .

If $s_i = 0$ then there is no slack and therefore $x_i = 0$. Furthermore since now $\lambda_i \neq 0$, the gradient condition say that the partial derivative $\frac{\partial f}{\partial x_i}$ must be greater than or equal to zero. Thus the optimality conditions for the problem can be written entirely in terms of optimization variables x_i, without explicitly involving the Lagrange multipliers , as follows :

$$\begin{pmatrix} \frac{\partial f}{\partial x_i} = 0 \; if \; x_i > 0 \\ \frac{\partial f}{\partial x_i} \geq 0 \; if \; x_i = 0 \end{pmatrix} \quad for \; i=1,2,....,m$$

$$\frac{\partial f}{\partial x_i} = 0 \quad for \; i=m+1,.........,n$$

4.3 Conjugate Gradient Algorithm for Quadratic Function with Non negativity Constraints

A conjugate gradient algorithm for minimizing an unconstrained function was discussed above. Two simple extensions are introduced into this algorithm to make It suitable for the present situation.

First, an active set idea is introduced. Active variables are defined as those variables that are either greater than zero or that violate the optimality condition. The remaining variables are called passive variables. By setting passive variables to zero, the objective function is expressed in terms of active variables only. The standard conjugate gradient algorithm for unconstrained problems is used to find direction with respect to active variables.

The **second** modification is in the step-length calculations. The step length given by the standard conjugate gradient algorithm for a quadratic function is first computed. If this step length does not make any of the restricted variables take on a negative value, then it is accepted and we proceed to the next iteration. If the standard step length is too large, a smaller value is computed that makes one of the active variables take a zero value. This variable is removed from the active set for the subsequent iteration, and the process is repeated until the optimality conditions are satisfied. The complete algorithm is as follows:

Initialization:

Set k =0. Choose an arbitrary starting point x_0.

1. Test for optimality:

$$If \quad \frac{\partial f}{\partial x_i} = 0 \ \ for\ x_i^k \succ 0 \ \ and$$

$$\frac{\partial f}{\partial x_i} \geq 0 \ for \ \ x_i^k = 0 \ \ for\ i=1,2,...m\ and,$$

$$\frac{\partial f}{\partial x_i} = 0\ for\ i = m+1,....n$$

Then stop. We have an optimum. Otherwise, continue.

Define the set of active variables consisting of those indices for which either $x_j > 0$ or $x_j = 0$ but corresponding $\frac{\partial f}{\partial x_i} < 0$.

The remaining variables are passive variables. All unrestricted variables are considered active. Thus,

$$I_a^k \ \left\{ i / \left(x_i^k > 0\right) or \left(x_i^k = 0\, and\, \frac{\partial f}{\partial x_i} < 0 \right) or \left(i > m\right) \right\},$$

$$I_p^k = \left\{ i / i \notin I_a^k \right\}$$

C. set passive variables to zero and use the conjugate gradient algorithm to minimize f(x) with respect to active variables.

1. Set the iteration counter $i = 0$.
2. Compute $\nabla f(x^i)$ with respect to active variables.
3 Compute

$$\beta = \begin{pmatrix} 0 & i=0 \\ \dfrac{[\nabla f(x^i)]^T \nabla f(x^i)}{[\nabla f(x^{i-1})]^T \nabla f(x^{i-1})} & i \succ 0 \end{pmatrix}$$

4. Compute direction $d^{i+1} = \begin{pmatrix} \nabla f(x^i) & i=0 \\ \nabla f(x^i) + \beta d^i & i \succ o \end{pmatrix}$

5. If $\|d^{i+1}\| \leq$tol, we have found the minimum of $f(x)$ with respect to active variables. Go to step (A) to check for optimality. Otherwise, continue.
6. Compute step length, keeping in mind that none of the variables should become negative.

$$\alpha = \min \left[- d^{i+1T} \nabla f(x^i) / d^{i+1T} Q\, d^{i+1}, \min \{ - x_j^i / d^i, d^i \succ 0, j = 1,2 \ldots, m \} \right]$$

7. Update $x^{i+1} = x^i + \alpha d^{i+1}$
8. If a new variable with an index less than or equal to m has reached a zero value, move that to the passive variable. Set $i = i + 1$ and go to **step (2).**

References:

1) Deumlich, R.**(1996)**, "*Optimization Theory II*", Text for the lecture, Addis Ababa .

2) Donalld M.Simons, " *Non Linear Programming for Operation research*

3) M. Asghar Bhatti,**(2000),"** *Practical Optimization methods with Mathematical application"*, Springer Verlag New York

4) Mokhtar S.Bazaraa;Hanif D.Shell, "*Non linear programming Theory and Algorithms"*

5) Nocedal,J. and Wright,S.**(1999),** "*Numerical Optimization*" Springer series in operation research, New York