Title	A method for nonconvex quadratic programming
Sub Title	
Author	Tone, Kaoru
Publisher	慶応義塾大学工学部
Publication year	1974
Jtitle	Keio engineering reports Vol.27, No.8 (1974.) ,p.113- 125
JaLC DOI	
Abstract	Our purpose here is to present a method for obtaining a global solution to a linearly constrained general quadratic minimization problem. The Kuhn-Tucker necessary conditions for a global solution of the problem are a linear complementarity problem. The author of this paper has proposed a method for a generalized linear complementarity problem, which bases on an algorithm for finding all extremal rays of a polyhedral convex cone with some complementarity conditions. In this paper, we apply the method to the linear complementarity problem of the general quadratic programming. And several devices will be adopted to decrease the amount of computation.
Notes	
Genre	Departmental Bulletin Paper
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO50001004-00270008-0113

KEIO ENGINEERING REPORTS
VOL. 27 NO. 8

A METHOD FOR NONCONVEX QUADRATIC PROGRAMMING

By

KAORU TONE

FACULTY OF ENGINEERING
KEIO UNIVERSITY
YOKOHAMA 1974

KEIO ENGINEERING REPORTS
VOL. 27, NO. 8, pp. 113-125, 1974

A METHOD FOR NONCONVEX QUADRATIC PROGRAMMING

KAORU TONE

Dept. of General Education, Keio University, Yokohama 223, Japan

(Received Jul. 24, 1974)

Abstract

Our purpose here is to present a method for obtaining a global solution to a linearly constrained general quadratic minimization problem. The Kuhn-Tucker necessary conditions for a global solution of the problem are a linear complementarity problem. The author of this paper has proposed a method for a generalized linear complementarity problem, which bases on an algorithm for finding all extremal rays of a polyhedral convex cone with some complementarity conditions. In this paper, we apply the method to the linear complementarity problem of the general quadratic programming. And several devices will be adopted to decrease the amount of computation.

1. Nonconvex Quadratic Programming and Kuhn-Tucker Conditions

The problem with which we are concerned can be stated as

$$\text{(1.1)} \qquad \text{Minimize} \quad Q(x)=x'Dx/2+c'x$$
$$\text{subject to} \quad Ax\geq b, x\geq 0,$$

where D is a symmetric (n, n) matrix, A is an (m, n) matrix, c is an n-column vector, b is an m-column vector and the symbol $'$ is used to denote the transposition of a matrix or a vector. And minimization here means to obtain the global minima of this programming.

Theorem 1. (KUHN and TUCKER (1951))

For a local minimum solution x of the quadratic program (1.1), *there exist vetors y, u and v such that*

$$\text{(1.2)} \qquad u=Dx-A'y+c$$

(1.3) $$v=Ax-b$$

(1.4) $$x'u=0$$

(1.5) $$y'v=0$$

(1.6) $$x\geq 0, y\geq 0, u\geq 0, v\geq 0$$

If the matrix D is positive semidefinite, the existence of x, y, u and v satisfying (1.2)-(1.6) is sufficient for x to be a global minimum solution of (1.1). But when D is not positive semidefinite (so-called nonconvex quadratic programming), the existence of such x, y, u and v is merely a necessary condition for x to be an optimal solution of (1.1). We will say that a point x satisfies the KUHN-TUCKER conditions of the quadratic programming if there exist y, u and v such that (1.2)-(1.6) are satisfied by (x, y, u, v). Such (x, y, u, v) will be called a KUHN-TUCKER solution for x. And x will be called a KUHN-TUCKER point of the quadratic program. A feasible point x of a quadratic program is called a stationary point if the directional derivative of the objective function is nonnegative in each feasible direction at x. For a quadratic program, it is well known that an optimal point is a stationary point, and that a point is stationary if and only if it satisfies the KUHN-TUCKER conditions.

Now, let

(1.7) $$w=\begin{pmatrix} u \\ v \end{pmatrix}, z=\begin{pmatrix} x \\ y \end{pmatrix}, q=\begin{pmatrix} c \\ -b \end{pmatrix}$$

and

(1.8) $$M=\begin{pmatrix} D & -A' \\ A & 0 \end{pmatrix}.$$

Then (1.2)-(1.6) become the following system of linear equations for which we wish to find out nonnegative and complementary solutions.

(1.9) $w=q+Mz$ (system of linear equations)

(1.10) $w\geq 0, z\geq 0$ (nonnegativity)

(1.11) $z'w=0$ (complementarity).

We can regard (1.9)-(1.11) to be a sort of a generalized linear complementarity problem and we can solve the system by an extremal ray algorithm proposed by the author of this paper in TONE (forthcoming).

In the next section, we will describe the outline of the algorithm.

2. The Extremal Ray Algorithm

We define a polyhedral convex cone P_m by a set of nonnegative vector x in the intersection of m half-spaces, that is,

(2.1) $$P_m=\{x\,|\,x\geq 0, a_1'x\geq 0, \cdots, a_m'x\geq 0\},$$

where $x, a_1, \cdots, a_m$ are n-vectors. We introduce slack variables $\lambda=(\lambda_1, \cdots, \lambda_m)'$ to make the inequalities in (2.1) into equalities. Thus, we have

(2.2) $$P_m=\{x \mid x \geq 0, \lambda \geq 0, a_1'x-\lambda_1=0, \cdots, a_m'x-\lambda_m=0\}.$$

Now, we put some complementarity conditions on the elements of x and λ. For example, $x_1\lambda_1=0, x_2\lambda_4=0, \lambda_2\lambda_3=0, x_3x_4x_5=0$ etc.. We call them the *complementarity* conditions.

The problem is to find out all extremal rays of P_m that satisfy these conditions. We will call an extremal ray of a polyhedral convex cone in these conditions a *complementary extremal ray* and a vertex of a polyhedral set in these conditions a *complementary vertex.*

Our method is iterative in the sense that knowing all complementary extremal rays of the polyhedral convex cone

(2.3) $$P_{k-1}=\{x \mid x \geq 0, a_1'x \geq 0, \cdots, a_{k-1}'x \geq 0\},$$

we add a constraint $a_k'x \geq 0$ to it to determine all complementary extremal rays of the polyhedral convex cone

(2.4) $$P_k=\{x \mid x \geq 0, a_1'x \geq 0, \cdots, a_{k-1}'x \geq 0, a_k'x \geq 0\}.$$

Here, when we mention of the complementary extremal rays of P_k, we only consider the complementarity conditions among $x_1, \cdots, x_n, \lambda_1, \cdots, \lambda_k$. We take no account of the complementarity conditions related to $\lambda_{k+1}, \cdots, \lambda_m$. The latter conditions are taken into consideration step by step as we proceed our algorithm and when the index k attains m, the complementarity conditions among all variables $x_1, \cdots, x_n$, $\lambda_1, \cdots, \lambda_m$ will be taken into consideration. As, at step k of the algorithm, we only consider a subset of the whole conditions, we call it *sub-complementarity* conditions. Similarly, we mean by a sub-complementary extremal ray or a sub-complementary vertex a ray or a vertex which satisfies these sub-complementarity conditions among its elements and corresponding slack veriables.

In regard to P_k, let

(2.5) $$C_k=\{x \mid x \in P_k, 1'x=1\}, \qquad \text{where} \quad 1'=(1, \cdots, 1).$$

C_k is a convex polyhedron. And there is a one to one correspondence between the vertex set of C_k and the extremal ray set of P_k. Indeed, the correspondence between $x \in P_k (x \neq 0)$ and $y=x/(1'x) \in C_k$ is such one. Also, by this correspondence, the sub-complementary extremal rays of P_k correspond to the sub-complementary vertices of C_k and vice versa. So, we hereafter deal with the set of sub-complementary vertices of C_k which we denote V_k.

The Extremal Ray Algorithm

Step 1. Initialization.

For

$$C_0=\{x \mid x \geq 0, 1'x=1\},$$

the sub-complementary vertex set is

(2.6) $$V_0=\{e_i|e_i: \text{the } i\text{-th unit vector, } i=1, \cdots, n\}.$$

Repeat the following steps for $k=1, \cdots, m$.

Step 2. Adding a constraint.

Assume all sub-complementary vertices of C_{k-1} are known. Let it be

(2.7) $$V_{k-1}=\{v_i\},$$

and let

(2.8) $$\lambda_{ik}=a'_k v_i.$$

Step 2.1. If $\lambda_{ik}\geq 0$ for all $v_i \in V_{k-1}$, then the adding constraint $a'_k x\geq 0$ is not binding. That is,

(2.9) $$C_k=C_{k-1}.$$

And let

(2.10) $$\bar{V}_k=V_{k-1}.$$

(Go to step 2.5.)

Step 2.2. If $\lambda_{ik}<0$ for all $v_i \in V_{k-1}$, then C_k is null. (The end.)

Step 2.3. If $\lambda_{ik}\leq 0$ for all i, then let

2.11) $$V_k=\{v_i|v_i \in V_{k-1}, \lambda_{ik}=0\}.$$

(Go to the beginning of step 2. Increase k by one.)

Step 2.4. If, for some i and some j, $\lambda_{ik}>0$ and $\lambda_{jk}<0$, then try the following [Common Zero Test] for v_i and v_j. If they passed the test, then compose a vector w_{ij} by

(2.12) $$w_{ij}=-\{\lambda_{jk}/(\lambda_{ik}-\lambda_{jk})\}v_i+\{\lambda_{ik}/(\lambda_{ik}-\lambda_{jk})\}v_j.$$

The w_{ij} is on the line segment joining v_i and v_j and on the hyperplane $H_k: a'_k x=0$. Try this process for all pairs of v_i (with $\lambda_{ik}>0$) and v_j (with $\lambda_{jk}<0$). Then let

(2.13) $$\bar{V}_k=\{v_i, w_{ij}|v_i \in V_{k-1}, a'_k v_i\geq 0\,;\, w_{ij} \text{ by (2.12)}\}.$$

(Go to step 2.5.)

Step 2.5. Try the following [Sub-Complementarity Test] to the elements of $\bar{V}_k$ to remove all non-sub-complementary vertices of C_k and let the remaining set be $\bar{V}_k$.
(Go to step 2.6.)

Step 2.6. Try the following [Degeneracy Test] to $\bar{V}_k$ to remove all non-vertex points of C_k from $\bar{V}_k$ and let the remaining set be V_k.
(Go to the beginning of step 2. Increase k by one.)

[Common Zero Test]

For v_i and v_j, let

(2.14) $$\lambda_{is}=a'_s v_i \qquad (s=1, \cdots, k-1),$$

(2.15)) $$\lambda_{js}=a'_s v_j \qquad (s=1, \cdots, k-1)$$

and let the extended vectors v_i^0 and v_j^0 of v_i and v_j be

(2.16) $$v_i^0=(v_{i1}, \cdots, v_{in}, \lambda_{i1}, \cdots, \lambda_{ik-1}, \lambda_{ik})',$$

and

(2.17) $$v_j^0=(v_{j1}, \cdots, v_{jn}, \lambda_{j1}, \cdots, \lambda_{jk-1}, \lambda_{jk})',$$

respectively. They are $(n+k)$-vectors. If v_i^0 and v_j^0 have no less than $(n-2)$ common zeros in their corresponding elements, then they pass the test. Otherwise, they fail.

[Sub-Complementarity Test]

For each v_i of $\bar{V}_k$, check the sub-complementarity among the elements of its extended vector v_i^0. If it does not satisfy the conditions, then remove v_i from $\bar{V}_k$.

[Degeneracy Test]

$\bar{V}_k$ consists of $v_i \in V_{k-1}$ and w_{ij} composed by (2.12). Let $\tilde{V}_k$ be the subset of $\bar{V}_k$ composed of the points on the hyperplane $H_k: a'_k x=0$. Of course $w_{ij} \in \tilde{V}_k$. If w_{ij} can be expressed by a convex combination of other points of V_k, then w_{ij} is not a vertex of C_k. To see this, test the following.

If there exist $w_{ij} \in \tilde{V}_k$ and $y_t \in \tilde{V}_k$ whose extended vectors we denote by w_{ij}^0 and y_t^0 respectively, such that for every positive elements of y_t^0, the corresponding elements of w_{ij}^0 are also positive and there is at least one positive element of w_{ij}^0 whose corresponding element of y_t^0 is zero, then w_{ij} is not a vertex of C_k. And we remove it from $\bar{V}_k$.

As to the validity of this algorithm, see TONE (forthcoming).

Remark. Degeneracy may happen rarely. And we need not try [Degeneracy Test] at every step. We may try it at the final step to final candidates.

3. The Extremal Ray Algorithm and Nonconvex Quadratic Programming

We can apply the extremal ray algorithm of the preceding section to the KUHN-TUCKER conditions (1.2)-(1.6) of the general quadratic program (1.1). For such purpose, we introduce a scalar variable t so as to make the system homogeneous as follows:

(3.1) $$u=Dx-A'y+ct$$

(3.2) $$v=Ax-bt$$

(3.3) $$x'u=0$$

(3.4) $$y'v=0$$

(3.5) $$x\geq 0, y\geq 0, u\geq 0, v\geq 0, t\geq 0.$$

Here we impose an assumption on the coefficient matrix A of (1.1).

Assumption

A certain row vector of the coefficient matrix A is negative. And we assume, without loss of generality, $a_1'<0$.

This assumption would be met or could be made to be met in most cases of practical problems. If not, we add a constraint to the original problem as the first constraint such as

$$x_1+x_2+\cdots+x_n\leq L$$

where L is a sufficiently large positive number.

This assumption will make the feasible region of the quadratic program a compact set, that is, a bounded polyhedron. Under this assumption the matrix A is of a bounded type and has one of and hence all of the following properties:

(1) $\{x|x\geq 0, x\neq 0, Ax\geq 0\}=\phi$,,

(2) for any b, $\{x|x\geq 0, Ax\geq b\}$ is bounded,

(3) there is a $\lambda\geq 0$ such that $\lambda' A<0$.

Thus, the quadratic program (1.1) either has no feasible solution or a global minimum at some point of the feasible region.

Theorem 2.

If a nonzero solution $(\bar{x}, \bar{y}, \bar{u}, \bar{v}, \bar{t})$ *of* (3.1)-(3.5) *has* $\bar{t}>0$, *then* $(\bar{x}/\bar{t}, \bar{y}/\bar{t}, \bar{u}/\bar{t}, \bar{v}/\bar{t})$ *is a* KUHN-TUCKER *solution for* $\bar{x}/\bar{t}$. *Conversely, if* $(\bar{x}, \bar{y}, \bar{u}, \bar{v})$ *is a* KUHN-TUCKER *solution of* (1.2)-(1.6), *then* $(\bar{x}, \bar{y}, \bar{u}, \bar{v}, 1)$ *is a solution of* (3.1)-(3.5).

Otherwise if a nonzero solution $(\bar{x}, \bar{y}, \bar{u}, \bar{v}, \bar{t})$ *of* (3.1)-(3.5) *has* $\bar{t}=0$, *then* $\bar{x}$ *must be zero.*

Proof. (i) The case $\bar{t}>0$ is obvious.

(ii) If $\bar{t}=0$, then $(\bar{x}, \bar{y}, \bar{u}, \bar{v})$ satisfies

$$\bar{u}=D\bar{x}-A'\bar{y}$$

$$\bar{v}=A\bar{x}$$

$$\bar{x}'\bar{u}=0$$

$$\bar{y}'\bar{v}=0$$

$$\bar{x}\geq 0, \bar{y}\geq 0, \bar{u}\geq 0, \bar{v}\geq 0, (\bar{x}, \bar{y})\neq 0.$$

Thus, if $\bar{x}\neq 0$ then $Ax\geq 0$ has a solution $\bar{x}\geq 0, \bar{x}\neq 0$ and it contradicts the boundedness of the feasible region.

Remark. Because there exists no x such that $Ax\geq 0$, $x\geq 0$ and $x\neq 0$ by the assumption, there is a $\bar{y}$ such that $-A'\bar{y}>0, \bar{y}\geq 0, \bar{y}\neq 0$ by a theorem by TUCKER (1956). And $(0, \bar{y}, -A'\bar{y}, 0, 0)$ is a solution of (3.1)-(3.5). That is, there always exists a solution with $t=0$.

As was mentioned above, the x-part of the solution of (3.1)-(3.5) with $t=0$ is also zero and if we make a positive linear combination of such solution vector with other solution vector with $t>0$, the x-part of the latter solution suffers no influence. Thus, we can conclude that to get all KUHN-TUCKER points of (1.1), it is necessary and sufficient to have to do with the complementary extremal ray solution of (3.1)-(3.5) with $t>0$.

Theorem 3.

We can find a global optimal solution of the quadratic program (1.1) *among the x-parts of the complementary extrmal ray solutions with $t=1$ of the generalized linear complementarity problem* (3.1)-(3.5).

Proof. If $(x, y, u, v, 1)$ is a solution of (3.1)-(3.5), then we have

$$Q(x)=(c'x+b'y)/2$$

That is, for the KUHN-TUCKER point of the quadratic programming, the objective function is a linear functional of the x and y. Thus it takes its global minimum on a certain extremal ray with $t=1$.

4. Algorithm for Nonconvex Quadratic Programming

Though we could directly apply the extremal ray algorithm to the linear complementarity problem of the quadratic programming, the necessary memory size and computational time would grow rapidly as the scale of the problem grows. Now, we will take several devices to improve the algorithm as outlined below.

(1) Constraints are taken into account one by one.

(2) An upper bound of the minimum of the objective function is set. And we neglect such constraints that were decided to have no KUHN-TUCKER points with $Q(x)$-values less than the upper bound.

(3) The upper bound shall be updated in the process of optimization.

The basic ideas are as follows.

Now, suppose, we have already processed the constraints up to the k-th, and we have found all KUHN-TUCKER points lying on it whose $Q(x)$-values are not greater than the upper bound. If the KUHN-TUCKER point which has the minimum $Q(x)$-value, is also a feasible point of the quadratic program, then we have already found an optimal solution. Otherwise, if the KUHN-TUCKER point with the minimum $Q(x)$-value is not feasible, we will choose a new constraint which cuts off the KUHN-TUCKER point, as the $(k+1)$-st constraint. Then, we apply the extremal ray algorithm to find out the KUHN-TUCKER points of the $Q(x)$ belonging to the region of k constraints and lying on the $(k+1)$-st constraint. If all KUHN-TUCKER points on the $(k+1)$-st constraint have $Q(x)$-values greater than the upper bound, then we can decide that the $(k+1)$-st constraint is "inactive" and we have $y_{k+1}=0$ hereafter. Otherwise, we will regard the constraint as "active" and update the KUHN-TUCKER point set, and the upper bound, if possible. Similarly, if there is no KUHN-TUCKER point not greater than the updated upper bound on the hyperplane $a_j'x=b_j$ which have been active on the foregoing stage, then we can decide that the constraint $a_j'x \geq b_j$ has become "inactive" and we have $y_j=0$ hereafter. Also, we can apply the same criterion to the constraint $x_i \geq 0$ and if there is no KUHN-TUCKER point not greater than the upper bound on the hyperplane $x_i=0$, then we have $u_i=0$ hereafter.

The above devices would contribute to decrease the necessity of getting all KUHN-TUCKER points and to decrease the necessary memory size.

Summary of Notations and Abbreviations Used

AS={Index of active constraints}. By active constraints are meant such constraints that might contain an optimal solution on the corresponding hyperplane. But this set is temporary and its elements may vary from time to time.

And, in order to keep the feasible region of each problem QPk (see below) compact, we always hold the first constraint (the bounding constraint) to this set.

IAS={Index of inactive constraints}. We call a constraint inactive if it was decided to contain no optimal point on the corresponding hyperplane, or if there exists, on the corresponding hyperplane, a KUHN-TUCKER point whose $Q(x)$-value updates the upper bound of $\min Q(x)$.

RS={Index of remainder constraints}. At the beginning of the algorithm, all constraints belong to this set.

UZS={Index i of elements of x for which the hyperplane $x_i=0$ was decided to contain no optimal point}. The corresponding $u_i=0$.

KTS={KUHN-TUCKER points which have $Q(x)$-values not greater than the upper bound}.

$X=\{x \mid Ax \geq b, x \geq 0\}$. The feasible region.

UPB=An upper bound of the minimum of $Q(x)(x \in X)$.

$KTX_{\min}$={x in KTS which has the minimum $Q(x)$-value}.

Algorithm

Step 1. Initialization.

Step 1.1. $AS=IAS=UZS=KTS=\phi$.

Step 1.2. $RS=\{i \mid i=1, \cdots, m\}$.

Step 1.3. UPB=An upper bound of $\min Q(x)$, if available. Otherwise, ∞. (Go to step 2.)

Step 2. Solving *QP1*.

According to the assumption of the preceding section, the constraint $a_1'x \geq b_1$ is bounding, that is, $\{x \mid a_1'x \geq b_1, x \geq 0\}$ is bounded and non-empty. So we have non-empty KUHN-TUCER point in this region.

Step 2.1. Solve the KUHN-TUCKER point set of the following $QP1$, by the extremal ray algorithm.

[$QP1$] Minimize $Q(x)$,
subject to $a_1'x \geq b_1, x \geq 0$.

Step 2.2. $AS = AS + \{1\}$.

Step 2.3. $RS = RS - \{1\}$.

Step 2.4. $KTS = \{$KUHN-TUCKER points of $QP1$ whose $Q(x)$-value is not greater than $UPB\}$.

Step 2.5. Let $Q(\bar{x}) = \min Q(x)(\bar{x}, x \in X \cap KTS)$.
If $Q(\bar{x}) < UPB$, then update UPB by $Q(\bar{x})$.

Step 2.6. If, for every $x \in KTS$ such that $a_1'x = b_1$, $Q(x)$-value is greater than UPB, then we have no optimal point on $a_1'x = b_1$. Thus, put $y_1 = 0$ hereafter. Similarly, if for every $x \in KTS$ such that $x_i = 0$, $Q(x)$-value is greater than UPB, then we have no optimal point on $x_i = 0$. Put $u_i = 0$ hereafter. $UZS = UZS + \{i\}$.
(Go to step 3.)

Step 3. Finding $KTX_{\min}$ and Checking the Feasibility.

Step 3.1. Let $KTX_{\min} = \{\bar{x} | Q(\bar{x}) = \min Q(x)\,;\, \bar{x}, x \in KTS\}$.

Step 3.2. If there exists an $\bar{x} \in KTX_{\min}$ which is also QP-feasible, then $\bar{x}$ is an optimal solution of the quadratic program (1.1). (The end.)
Otherwise, go to step 4.

Step. 4. Finding a Cutting Constraint.

Step 4.1. Take up, from RS, a constraint which cuts off a point of $KTX_{\min}$. Let it be $a_k'x \geq b_k$.

Step 4.2. $RS = RS - \{k\}$.
(Go to step 5.)

Step 5. Solving *QPk*.

Step 5.1. Solve the following QPk by the extremal ray algorithm and get the KUHN-TUCKER points on the hyperplane $a_k'x = b_k$.

[PQk] Minimize $Q(x)$,
subject to $a_i'x \geq b_i (i \in AS), a_k'x = b_k, x \geq 0$, where $y_j = 0 (j \in IAS)$ and $u_i = 0 (i \in UZS)$ and $y_1 = 0$ if step 2.6. is effective.

Step 5.2. If PQk has no feasible point, then the quadratic program (1.1) has no feasible point. (The end.)
Otherwise, go to step 6.

[Remark 1: Here we solve the QPk independently of the preceding subproblem QPk'. But we could take advantage of the information about the vertex set of the feasible region of QPk', in order to solve the QPk. For such purpose, we must keep the information about the vertex set of the polyhedron formed by the active constraints and also must update it at certain steps of the algorithm.]

[Remark 2: It is certain that the necessary memory size for solving the QPk is fairly less than that for solving the QP by applying directly the extremal ray algorithm.]

Step 6. Is the *k*-th constraint active?

Step 6.1. For each KUHN-TUCKER point of QPk which is also feasible with respect to the constraints in IAS, is the $Q(x)$-value greater than

UPB? If not, go to 6.2. Otherwise, go to 6.4.

Step 6.2. Let $Q(\bar{x})$ be the minimum of $Q(x)$ where x is a KUHN-TUCKER point of *QPk*, satisfying *IAS*-feasibility. If $\bar{x} \in X$, then go to 6.3. otherwise go to step 7.

Step 6.3. If $Q(\bar{x}) \leq KTX_{\min}$, then $\bar{x}$ is an optimal solution of *QP*. (The end.) Otherwise, update *UPB* by $Q(\bar{x})$. $KTS = KTS + \{\bar{x}\}$.

Step 6.4. $IAS = IAS + \{k\}$.
(Go to step 8.)

[Remark 3: When we follow step 6.3.-6.4., the point $\bar{x}$ is the only virtually necessary point on the k-th hyperplane, as $\bar{x}$ is the minimum *QP*-feasible point on the hyperplane. In this case, we regard the k-th constraint inactive. This will contribute to keep the active constraint set small and hence to decrease the amount of computation.]

Step 7. The *k*-th constraint is active.

Step 7.1. $AS = AS + \{k\}$.

Step 7.2. Let $Q(\bar{\bar{x}})$ be the minimum of $Q(x)$ where x is a KUHN-TUCKER point of *QPk*, satisfying *QP*-feasibility. If $Q(\bar{\bar{x}}) < UPB$, then update *UPB* by $Q(\bar{\bar{x}})$.

Step 7.3. $KTS = KTS + \{$*IAS*-feasible KUHN-TUCKER point of *QPk* whose $Q(x)$-value is not greater than *UPB*.$\}$
(Go to step 8.)

Step 8. Updating *KTS, AS, IAS* and *UZS*.

Step 8.1. Remove from *KTS* such points that were cutted off by the k-th constraint.

Step 8.2. Remove from *KTS* such points that have $Q(x)$-value greater than *UPB*, if possible.

Step 8.3. Remove from *AS* such indices ($\neq 1$) that have no *KTS* points on the corresponding hyperplanes. Let such indices belong to *IAS*.

Step 8.4. If there is at least one *QP*-feasible point in *KTS* then $UZS = UZS + \{i \mid$The corresponding hyperplane $x_i = 0$ has no *KTS* point.$\}$, if any.
(Go to step 3.)

5. A Numerical Example

Minimize $\quad Q(x) = x_1^2/2 + x_2^2 + x_3^2/2 + 2x_1x_2 + 2x_1x_3 - 3x_1 + 2x_2 - 4x_3,$

subject to

$$-x_1 - x_2 - x_3 \geq -10$$

$$x_1 + x_2 - x_3 \geq -2$$

$$-x_1 - 2x_2 \geq -6$$

$$4x_1 - 4x_2 - x_3 \geq -4$$

$$x_1, x_2, x_3 \geq 0.$$

First, taking up the bounding constraint $-x_1-x_2-x_3\geq-10$, we solve the following $QP1$.

[$QP1$] Minimize $Q(x)$,

subject to $-x_1-x_2-x_3\geq-10, x_1, x_2, x_3\geq 0.$

The KUHN-TUCKER conditions are as follows;

$$v_1=-x_1-x_2-x_3+10$$

$$u_1=x_1+2x_2+2x_3+y_1-3$$

$$u_2=2x_1+2x_2+y_1+2$$

$$u_3=2x_1+x_3+y_1-4$$

$$x_1u_1=x_2u_2=x_3u_3=y_1v_1=0$$

$$x_1, x_2, x_3, y_1, u_1, u_2, u_3, v_1\geq 0.$$

As shown in table 1, we found three KUHN-TUCKER points by the extremal ray algorithm, that is, $(x_1=3, x_2=0, x_3=0)$, $(x_1=0, x_2=0, x_3=4)$ and $(x_1=5/3, x_2=0, x_3=2/3)$ and their $Q(x)$-values are -4.5, -8, -3.83, respectively. The first is QP-feasible but the others are not. Thus, we have $UPB=-4.5$. Because the three points do not lie on $-x_1-x_2-x_3=-10$ and the first is QP-feasible, we have $y_1=0$ hereafter by step 2.6. Next, we take up the constraint $x_1+x_2-x_3\geq-2$, for it cuts off $KTX_{\min}$=the second. Thus, we have:

[$QP2$] Minimize $Q(x)$,

subject to $-x_1-x_2-x_3\geq-10, x_1+x_2-x_3=-2, x_1, x_2, x_3\geq 0.$

The KUHN-TUCKER conditions are as follows:

Table 1. The solution of $QP1$

	x_1	x_2	x_3	y_1	t	v_1	u_1	u_2	u_3	Remark
$V1$					1	10	-3	*	*	
$V2$	10				1	0	7	*	*	
$V3$		10			1	0	17	22	*	
$V4$			10		1	0	17	2	6	
$V5$				1	0	0	1	1	1	
$V6=V(1,2)$	3				1	7	0	8	2	A KUHN-TUCKER point, $Q=-4.5$
$V7=V(1,3)$		3/2			1	17/2	0	5	*	
$V8=V(1,4)$			3/2		1	17/2	0	2	$-5/2$	
$V9=V(4,8)$			4		1	6	5	2	0	A KUHN-TUCKER point, $Q=-8$
$V10=V(6,8)$	5/3		2/3		1	23/3	0	16/3	0	A KUHN-TUCKER point, $Q=-3.83$

Table 2. The solution of $QP2$

	x_1	x_2	x_3	y_2	t	v_1	v_2	u_1	u_2	u_3	Remark
V1	10				1	0	12	*	*	*	
$V2$		10			1	0	12	*	*	*	
$V3$			10		1	0	-8	*	*	*	
$V4$				1	1	0	0	-1	*	*	
$V5$					1	10	2	*	*	*	
$V6=V(1,3)$	4		6		1	0	0	13	*	*	
$V7=V(2,3)$		4	6		1	0	0	17	10	*	
$V8=V(3,5)$			2		1	8	0	1	2	-2	
$V9=V(4,6)$	4		6	13	1	0	0	0	-3	*	
$V10=V(4,7)$		4	6	17	1	0	0	0	-7	*	
$V11=V(4,8)$			2	1	1	8	0	0	1	-1	
$V12=V(9,11)$	1		3	4	1	6	0	0	0	5	
$V13=V(7,10)$		4	6	10	1	0	0	7	0	12	
$V14=V(10,11)$		1/2	5/2	3	1	7	0	0	0	3/2	
$V15=V(11,12)$	1/6		13/6	3/2	1	23/3	0	0	5/6	0	A KUHN-TUCKER point, $Q=-6.08$

$$v_1=-x_1-x_2-x_3+10$$

$$0=v_2=x_1+x_2-x_3+2$$

$$u_1=x_1+2x_2+2x_3-y_2-3$$

$$u_2=2x_1+2x_2-y_2+2$$

$$u_3=2x_1+x_3+y_2-4$$

$$x_1u_1=x_2u_2=x_3u_3=0. \quad \text{(Notice } y_1=0.)$$

As shown in table 2, we found a KUHN-TUCKER point $(x_1=1/6, x_2=0, x_3=13/6)$ with the $Q(x)$-value -6.08. Now, KTS consists of $(x_1=3, x_2=0, x_3=0)$, $(x_1=5/3, x_2=0, x_3=2/3)$ and $(x_1=1/6, x_2=0, x_3=13/6)$ and $KTX_{\min}$ is the last which is also QP-feasible. Thus, we have found an optimal solution of the quadratic program, $(x_1=1/6, x_2=0, x_3=13/6)$.

6. Concluding Remark

The nonconvex quadratic programming is, together with the integer linear programming, a representative of general nonconvex programmings and several methods have recently been proposed on this subject among which RITTER's cutting plane algorithm (RITTER (1966)) and TUI's algorithm (TUI (1964)) for the global minimization of a concave function subject to linear inequality constraints are remarkable. But, to our regret, the convergence to the global optimum solution of these methods does not seem to be guaranteed, as pointed out by ZWART (1973). Our method will be said to be of an opposite feature as far as this point of convergence is concerned. Although our method is enumerative in nature, it would be possible to reduce the amount of computation, by taking advantage of the complementarity conditions and by using several devices developed in the preceding section.

Acknowledgments

The author would like to express his grateful appreciation to Professor TATSUO KAWATA of Keio University for his support and encouragement. And also he wishes to thank a referee for helpful comments on an original draft of this paper.

References

KUHN, H. W. and A. W. TUCKER (1951): Nonlinear programming, Proecedings of the Berkeley Symposium of Mathematical Statistics and Probability, University of California Press, Berkeley, California.

RITTER, K. (1966): A method for solving maximum-problem with a nonconcave quadratic objective function, Z. Wahrscheinlichkeitstheorie verw. Geb., **4**, pp. 340–351.

TONE, K. (forthcoming): An algorithm for finding all extremal rays of polyhedral convex cones with some complementarity conditions, to appear in the Journal of the Operations Research Society of Japan.

TUCKER, A. W. (1956): Dual systems of homogeneous linear relations, in "Linear Inequalities and Related Systems," Annals of Mathematics Studies Number 38, Princeton University Press, Princeton N.J.

TUI, H. (1964): Concave programming under linear constraints, Dokl. Akad. Naul. SSR., **159**, pp. 1437–1440.

ZWART, P. B. (1973): Nonlinear programming: counterexamples to global optimization algorithms proposed by Ritter and TUI, Opns. Res. **21**, 1260–1266.