Journal of Global Optimization **16:** 77–93, 2000.

An Algorithm for Global Minimization of Linearly Constrained Quadratic Functions*

OSCAR BARRIENTOS[1] and RAFAEL CORREA[2]
[1]*Subgerencia Planificación Nacional, ENDESA, Santa Rosa 76, Santiago, Chile (E-mail: obarrien@dim.uchile.cl);* [2]*Depto. de Ingeniería Matemática, Universidad de Chile, Casilla 170/3 Correo 3, Santiago, Chile*

(Received 6 August 1996; accepted 23 September 1998)

Abstract. A branch and bound algorithm is proposed for finding an approximate global optimum of quadratic functions over a bounded polyhedral set. The algorithm uses Lagrangian duality to obtain lower bounds. Preliminary computational results are reported.

Key words: Constrained global optimization, quadratic programming, branch and bound methods, duality techniques

0. Introduction

In this paper we consider several aspects of quadratic programming problems. The general problem is of the form

$$\min_{y\in\Omega} \phi(y) =< Qy, y > + < c, y >$$

where $\Omega = \{y \in \mathbb{R}^n : \ Ay \leqslant b\}$ is a bounded polyhedral set, $Q \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $< \cdot, \cdot >$ denotes the usual inner product on $\mathbb{R}^n$. Quadratic programming is a very old and important problem of mathematical programming. It has numerous applications in many diverse fields of science and technology, and plays a key role in many nonlinear programming methods.

On the other hand, a broad class of difficult combinatorial problems can be formulated as nonconvex quadratic global minimization problems, for example: integer programming, quadratic 0-1 programming, quadratic assignment problems, bilinear programming, linear complementary problems, max-min problems (see Pardalos and Rosen [13]).

In the nonconvex case, it has been shown that the linearly constrained problem is NP-complete (see Murty and Kabadi [11]). Moreover, to check only local optimality in constrained nonconvex programming is NP-hard (see Pardalos and Schnitger [14]). From a computational viewpoint, this means that, in the worst

*This research was supported in part by CONICYT-CEE and FONDECYT # 1961107

case, the computing time required to obtain a solution will grow exponentially with the number of variables.

Traditional nonlinear programming methods usually obtain local solutions when applied to indefinite quadratic problems. In many applications the global optimum or a good approximation to global optimum is required. Recently some new approaches have been developed for finding the global optimum for this problem. One such approach is given in [7]. Other earlier works include [10], which is based on a search procedure using gradient projection, and a generalized Benders' cut procedure developed by Geoffrion [5] and used in [8]. Similarly Tuy [19] uses a method based on Benders' decomposition technique for the global minimization of difference of two convex functions. More recent approaches are presented in [3, 6, 12, 15, 16, 17].

In this paper we use branch and bound methods to obtain approximate solutions, and Lagrangian duality is used to obtain lower bounds. This idea was introduced by Falk in [4]. Ben-Tal et al. in [2], using a technique for reducing the duality gap, incorporate this idea for solving bilinear constrained linear programs. The description of this technique and a result showing that if the diameter of cover is sufficiently small then the duality gap is arbitrarily small are presented in the first section.

Next, in Section 2, we describe the general branch and bound method which is implicit in this technique. Moreover, we prove some results concerning the behavior of this method, such as its convergence and after a finite number of refinements of cover the only sets that remain are those containing a global solution of the problem.

In Section 3, we apply these results to linearly constrained quadratic problem.

Finally, in the last section, we describe the algorithm for quadratic problems and we report preliminary computational results.

1. Reducing the Duality Gap

Our starting problem in this section is the constrained minimization program

$$\text{(P)} \qquad \alpha = \min\{f_0(y) : f_i(y) \leqslant 0, i \in \{1, \ldots, m\}, y \in B\}, \tag{1}$$

where f_i are continuous real-valued functions on $\mathbb{R}^n$ for each $i \in \{0, 1, \ldots, m\}$, and B is non-empty closed set in $\mathbb{R}^n$. We suppose that the minimum in (P) is attained.

It is well known that we may write (P) as

$$\alpha = \min_{y\in B} \sup_{u\in\mathbb{R}^m_+} \left\{ f_0(y) + \sum_{i=1}^{m} u_i f_i(y) \right\}. \tag{2}$$

Moreover, the dual of program (P) is usually written as

$$\text{(D)} \qquad \beta = \sup_{u \in \mathbb{R}^m_+} \inf_{y \in B} \left\{ f_0(y) + \sum_{i=1}^{m} u_i f_i(y) \right\}, \tag{3}$$

and we always have the inequality

$$\beta \leqslant \alpha, \tag{4}$$

while the equality in (4) does not necessarily hold. The general objective of this section is to develop an idea for reducing the duality gap $\alpha - \beta$.

Let us consider a closed cover

$$\Lambda = \{R_j : j \in J\} \tag{5}$$

of the closed set B, that is, for each $j \in J$, R_j is a closed subset of B and $\cup_{j \in J} R_j = B$. Now, let us write (2) in the form

$$\alpha = \min_{j \in J} \min_{y \in R_j} \sup_{u \in \mathbb{R}^m_+} L(y, u), \tag{6}$$

where L is the Lagrange function defined on $\mathbb{R}^n \times \mathbb{R}^m_+$, that is, $L(y, u) = f_0(y) + \sum_{i=1}^{m} u_i f_i(y)$.

From (6), we define the *intermediate dual program* of program (P) as

$$\gamma(\Lambda) = \min_{j \in J} \sup_{u \in \mathbb{R}^m_+} \inf_{y \in R_j} L(y, u) \tag{7}$$

and we assume that the infima in (7) are attained.

It is easy to see that

$$\beta \leqslant \gamma(\Lambda) \leqslant \alpha, \tag{8}$$

that the left inequality in (8) is an equality when $\Lambda = \{B\}$ and that the right one is an equality when $\Lambda = \{\{y\} : y \in B\}$. *The specific goal of this section is to prove that there always exists a finite closed cover Λ of B such that the gap $\alpha - \gamma(\Lambda)$ is arbitrarily small.*

We define the following closed subsets of B:

$$C = \{y \in B : f_i(y) \leqslant 0, i \in \{1, \ldots, m\}\} \tag{9}$$

and for any $\bar{y} \in B$

$$B_{\bar{y}}(t) = \{y \in B : \|\bar{y} - y\| \leqslant t\}. \tag{10}$$

We moreover define the function $\ell_{\bar{y}}$ from $\mathbb{R}^m_+ \times \mathbb{R}_+$ to $\mathbb{R}$ by

$$\ell_{\bar{y}}(u, t) = \min_{y \in B_{\bar{y}}(t)} L(y, u) \tag{11}$$

and by associating the sets R_j in (7) with $B_{\bar{y}}(t)$, we define the function $F_{\bar{y}}$ from $\mathbb{R}_+$ to $\mathbb{R} \cup \{+\infty\}$ by

$$F_{\bar{y}}(t) = \sup_{u \in \mathbb{R}^m_+} \ell_{\bar{y}}(u, t). \tag{12}$$

We next give a sufficient condition for the continuity of the function $F_{\bar{y}}$ at $t = 0$, for each $\bar{y} \in C$.

LEMMA 1.1. *If the Lagrange function L of program (P) defined in (6) is continuous on $\mathbb{R}^n \times \mathbb{R}^m_+$, then for all $\bar{y} \in C$ defined in (9) the function $F_{\bar{y}}$ defined in (12) is continuous at $t = 0$. Moreover, $F_{\bar{y}}(t) < +\infty$ for all $t \in \mathbb{R}_+$.*

Proof. We firstly show the second assertion. Given $t \in \mathbb{R}_+$, since $\bar{y} \in C$ we have for all $u \in \mathbb{R}^m_+$

$$\ell_{\bar{y}}(u, t) \leqslant L(\bar{y}, u) \leqslant f_0(\bar{y}), \tag{13}$$

thus

$$F_{\bar{y}}(t) \leqslant f_0(\bar{y}) < +\infty. \tag{14}$$

We now prove the continuity of function $F_{\bar{y}}$ at $t = 0$. Since $\bar{y} \in C$, we have $F_{\bar{y}}(0) = f_0(\bar{y})$, and the above inequality therefore implies

$$\limsup_{t \to 0^+} F_{\bar{y}}(t) \leqslant F_{\bar{y}}(0),$$

that is, the upper semi-continuity of function $F_{\bar{y}}$ at $t = 0$.

On the other hand, from the definition of function $F_{\bar{y}}$ we have the inequality

$$\ell_{\bar{y}}(u, t) \leqslant F_{\bar{y}}(t) \quad \text{for all } \; u \in \mathbb{R}^m_+, \; t \in \mathbb{R}_+$$

and since, the function $\ell_{\bar{y}}$ is continuous on $\mathbb{R}^m_+ \times \mathbb{R}_+$ (see for example Auslender [1], page 54), we can write

$$F_{\bar{y}}(0) = \sup_{u \in \mathbb{R}^m_+} \ell_{\bar{y}}(u, 0) \leqslant \liminf_{t \to 0^+} F_{\bar{y}}(t)$$

that is, the lower semi-continuity of function $F_{\bar{y}}$ at $t = 0$. □

REMARK . A weaker condition ensuring the same result is the lower semi-continuity (on C) of the Lagrange function $L(\cdot, u)$ for each $u \in \mathbb{R}^m_+$, because this semi-continuity implies the lower semi-continuity at $t = 0$ of functions $\ell_{\bar{y}}(u, \cdot)$ for all $u \in \mathbb{R}^m_+$.

In order to simplify notation, for any cover $\Lambda = \{R_j : j \in J\}$ of B we define the quantities

$$\begin{aligned} \operatorname{diam}(R_j) &= \sup\{\|y - y'\| : \; y, y' \in R_j\}, \\ \delta(\Lambda) &= \sup_{j \in J} \operatorname{diam}(R_j). \end{aligned} \tag{15}$$

We now can establish the principal result of this section. *It tells us that the duality gap can be made arbitrarily small if the closed cover is fine enough.*

THEOREM 1.2. *If B is a compact set and if the Lagrange function L of program (P) defined in (6) is continuous on $\mathbb{R}^n \times \mathbb{R}^m_+$, then for all $\varepsilon > 0$ there exists $\delta > 0$ such that for any cover $\Lambda = \{R_j : j \in J\}$ of B verifying $\delta(\Lambda) \leqslant \delta$, we have*

$$(0 \leqslant)\ \alpha - \gamma(\Lambda) \leqslant \varepsilon \tag{16}$$

Proof. Given $\varepsilon > 0$, from the above lemma we know that for each $y \in C$ the function F_y (defined in (12)) is continuous at $t = 0$; there therefore exists $t(y) > 0$ such that $F_y(0) - F_y(t(y)) \leqslant \varepsilon$. On the other hand, for each $z \in B \setminus C$ there exists $t(z) > 0$ such that $F_z(t(z)) > \alpha + 1$ where α is defined in (1).

We now consider the cover $\{B_y(t(y))\}_{y \in C} \cup \{B_z(t(z))\}_{z \in B \setminus C}$ of B. The compactness of B allows us to choose a finite number of $y_j \in C$, $j \in \{1, \dots, q\}$, $z_j \in B \setminus C$, $j \in \{q+1, \dots, p\}$ such that $\Lambda = \{B_{y_j}(t(y_j)) : j \in \{1, \dots, q\}\} \cup \{B_{z_j}(t(z_j)) : j \in \{q+1, \dots, p\}\}$ is a finite cover of B.

Since $F_{z_j}(t(z_j)) > \alpha + 1$ for all $j \in \{q+1, \dots, p\}$ we have that

$$\gamma(\Lambda) = \min_{j \in \{1, \dots, q\}} F_{y_j}(t(y_j))$$

Let $j_o \in \{1, \dots, q\}$ be such that $\gamma(\Lambda) = F_{y_{j_o}}(t(y_{j_o}))$. Since $\alpha = \min\limits_{y \in B} F_y(0)$ we can conclude that

$$\alpha - \gamma(\Lambda) \leqslant F_{y_{j_o}}(0) - F_{y_{j_o}}(t(y_{j_o})) \leqslant \varepsilon,$$

and this will be true for any cover having a subset $R_j \subset B_{y_{j_o}}(t(y_{j_o}))$. □

2. A Branch and Bound Method

In this section we describe the simple Branch and Bound method implicit in Theorem 1.2 of Section 1. Moreover, we establish its convergence in a finite number of iterations.

In the method that we propose, *branching* is the refinement of Λ and *bounding* is the determination of lower and upper bounds for the value α defined in (1).

We start with defining $\Lambda = \{B\}$. If we initially know a feasible point $\bar{y} \in B$, i.e. such that $f_i(\bar{y}) \leqslant 0$ for all $i \in \{1, \dots, m\}$, we start with defining $\alpha(\Lambda) = f_0(\bar{y})$, otherwise $\alpha(\Lambda) = +\infty$.

Now, at each iteration we define new local refinements of Λ and we improve the lower bound of the value α by solving dual programs of the type

$$\sup_{u \in \mathbb{R}^m_+} \ell(u), \tag{17}$$

where $\ell(u) = \min\limits_{y \in R} L(y, u)$ and R is a set in the closed cover Λ of B.

Let us now describe precisely this Branch and Bound method:

The Bounding. At the current iteration we have a closed cover $\Lambda = \{R_j : j \in J\}$ of B. We solve the dual programs

$$\gamma_j = \sup_{u \in \mathbb{R}^m_+} \ell_j(u) \qquad \text{for all} \quad j \in J, \tag{18}$$

where $\ell_j(u) = \min_{y \in R_j} L(y, u)$. Moreover, for each $j \in J$ we take some $y_j \in R_j$; if $f_i(y_j) \leqslant 0$ for all $i \in \{1, \ldots, m\}$ we put

$$\alpha(\Lambda) = \min\{\alpha(\Lambda),\ f_0(y_j)\} \tag{19}$$

We define $\gamma_{\hat{j}} = \min_{j \in J} \gamma_j$. If $\alpha(\Lambda) - \gamma_{\hat{j}} \leqslant \varepsilon$ we stop the algorithm, otherwise we branch.

The Branching. We take the set $R_{\hat{j}}$ defined by the bounding, and we design a closed cover $\{R_j : j \in J'\}$ of it. We define the new closed cover $\{R_j : j \in J\}$ of B by redefining $J \leftarrow J \cup J' \setminus \{\hat{j}\}$.

In the first section, in Theorem 1.2, we have established (under weak assumptions) that there always exists a finite closed cover Λ of B such that the gap $\alpha - \gamma(\Lambda)$ is arbitrarily small. We now wish to show that the above upper bounding procedure has similarly properties, that is, there always exists a finite closed cover Λ of B such that the quantity $\alpha(\Lambda) - \alpha$ is arbitrarily small. Of course, we must append some additional assumptions on the data of our program (P) defined in (1).

THEOREM 2.1. *If B is a compact set and*
(i) the function f_0 is continuous on $\mathbb{R}^n$,
(ii) for all $i \in \{1, \ldots, m\}$ the function f_i is convex finite on $\mathbb{R}^n$,
(iii) there exists $y_0 \in B$ such that $f_i(y_0) < 0$ for all $i \in \{1, \ldots, m\}$.
Then, for all $\varepsilon > 0$ there exists $\delta > 0$ such that for any closed cover $\Lambda = \{R_j : j \in J\}$ of B verifying $\delta(\Lambda) \leqslant \delta$, we have

$$\alpha(\Lambda) - \alpha \leqslant \varepsilon. \tag{20}$$

Proof. Let $y^* \in C$ (defined in (9)) such that $f_0(y^*) = \alpha$, that is, y^* is a global solution of problem (P) (defined in (1)).

From assumptions i), ii) and iii), it is easy to see that there exist $\tilde{y} \in C$ and $\delta' > 0$ such that the set $B_{\tilde{y}}(\delta')$ is a subset of the interior of C and

$$f_0(y) - f_0(y^*) \leqslant \varepsilon \text{ for all } y \in B_{\tilde{y}}(\delta').$$

We now take a closed cover $\Lambda = \{R_j : j \in J\}$ of B such that there exists $j \in J$ verifiying $R_j \subset B_{\tilde{y}}(\delta')$. Then, since $f_0(y^*) = \alpha$, we can conclude that

$$\alpha(\Lambda) - \alpha \leqslant \max_{y \in R_j} f_0(y) - \alpha \leqslant \max_{y \in B_{\tilde{y}}(\delta')} f_0(y) - \alpha \leqslant \varepsilon,$$

and this will be true for any cover having a subset $R_j \subset B_{\tilde{y}}(\delta')$. □

Consequently, we now can derive a result ensuring the convergence of the above method in a finite number of iterations. Of course, we must append some additional assumptions on The Branching step. In view of a more implementable condition see the remark following the theorem below.

THEOREM 2.2. *Suppose, in addition to assumptions of Theorem 2.1 above, that The Branching step is a process such that the diameter $\delta(\Lambda)$ of the closed cover Λ of B converges to zero. Then, the Branch and Bound method described above converges in a finite number of iterations. That is to say, we obtain a point $y \in C$ such that*

$$f_0(y) - \alpha \leqslant \varepsilon. \tag{21}$$

Proof. It is a direct consequence of Theorems 1.2 and 2.1 above. □

REMARK . From proofs of Theorems 1.2 and 2.1 we see that the condition assumed for the Branching step can be weakened. In fact, it is sufficient that there exists $0 < \eta < 1$ such that, if the set $R_{\hat{j}}$ should be branched, then the new closed cover $\{R_j : j \in J'\}$ of it, verifies

$$\mathrm{diam}(R_j) \leqslant \eta \, \mathrm{diam}(R_{\hat{j}}) \text{ for all } j \in J'. \tag{22}$$

On the other hand, the performance of this method is that when one makes the Bound step, all the subsets R_j for which $\gamma_j > \alpha(\Lambda) + \varepsilon$ will never be considered in the branch. In fact, we will only make local refinements of the closed cover Λ. Of course, this is the philosophy of all Branch and Bound methods.

To end this section, we prove a result that tells us that, after a finite number of iterations, the only sets that remain are those containing an approximated global solution of problem.

PROPOSITION 2.3. *Let $\Lambda = \{R_j : j \in J\}$ be a closed cover of B and let $R_{\hat{j}} \in \Lambda$ be such that*

$$R_{\hat{j}} \cap \{\, y^* \in C : f_0(y^*) = \alpha \,\} = \emptyset. \tag{23}$$

If $R_{\hat{j}}$ is a compact set and

(i) if $R_{\hat{j}} \cap C = \emptyset$, then $R_{\hat{j}}$ will be leaved out after a finite number of refinements of Λ.

(ii) if $R_{\hat{j}} \cap C \neq \emptyset$, then we will have after a finite number of refinements of Λ the following inequality

$$\sup_{u \in \mathbb{R}^m_+} \min_{y \in R_j} L(y, u) > \alpha \quad \text{for all} \quad R_j \subset R_{\hat{j}}. \tag{24}$$

Proof. i) Given $\eta \in \mathbb{R}$, for all $z \in R_{\hat{j}}$ there exists $t(z) > 0$ such that $F_z(t(z)) > \eta$.

From the compactness of the set $R_{\hat{j}}$, we can choose a finite number of $z_k \in R_{\hat{j}}$, $k \in \{1, \dots, q\}$ such that $\{B_{z_k}(t(z_k)/3) : k \in \{1, \dots, q\}\}$ cover $R_{\hat{j}}$.

We set $\delta = \frac{1}{3} \min_{k\in\{1,\dots,q\}} t(z_k)$. Let Λ' be a closed cover of $R_{\hat{j}}$ such that

$$\operatorname{diam}(R_{j'}) \leqslant \delta \quad \text{for all} \quad R_{j'} \in \Lambda'.$$

Since for all $R_{j'} \in \Lambda'$ there exists $k \in \{1, \dots, q\}$ verifying $R_{j'} \subset B_{z_k}(t(z_k))$, we have

$$\sup_{u\in\mathbb{R}^m_+} \min_{y\in R_{j'}} L(y,u) \geqslant \sup_{u\in\mathbb{R}^m_+} \min_{y\in B_{z_k}(t(z_k))} L(y,u) > \eta.$$

Therefore, to leave out $R_{\hat{j}}$ it is sufficient to choose η greater than $\alpha + \varepsilon$ (for example $\eta = \alpha(\Lambda) + 2\varepsilon$).

(ii) We define the auxiliary problem (P$'$) as follows:

$$\text{(P')} \qquad \eta = \min\{f_0(y) : f_i(y) \leqslant 0, i \in \{1, \dots, m\}, y \in R_{\hat{j}}\}.$$

From the compactness of the set $R_{\hat{j}}$ and equality (23), we have $\alpha < \eta$. Moreover, since $R_{\hat{j}} \cap C \neq \emptyset$, we then have $\eta < +\infty$.

We now apply Theorem 1.2 to problem (P’) with $\varepsilon = \frac{1}{2}(\eta - \alpha) > 0$; then there exists $\delta' > 0$ such that for any closed cover Λ' of $R_{\hat{j}}$ verifying

$$\operatorname{diam}(R_j) \leqslant \delta' \quad \text{for all} \quad R_j \in \Lambda',$$

we have: $\eta - \min_{R_j\in\Lambda'} \sup_{u\in\mathbb{R}^m_+} \min_{y\in R_j} L(y,u) \leqslant \varepsilon$. But, that is equivalent to $\frac{1}{2}(\eta + \alpha) \leqslant \min_{R_j\in\Lambda'} \sup_{u\in\mathbb{R}^m_+} \min_{y\in R_j} L(y,u)$.

Therefore, for all $R_j \in \Lambda'$ we get $\alpha < \sup_{u\in\mathbb{R}^m_+} \min_{y\in R_j} L(y,u)$. $\square$

3. Global Minimization of Linearly Constrained Quadratic Functions

In this section we consider the linearly constrained quadratic problem

$$\text{(QP}'\text{)} \qquad \min_{x\in\Omega'} \phi'(x) = < Qx, x > + < c', x >, \tag{25}$$

where $\Omega' = \{x \in \mathbb{R}^n : A'x \leqslant b'\}$ is a bounded polyhedral set, $Q \in \mathbb{R}^{n\times n}$ is a symmetric matrix and $A' \in \mathbb{R}^{m\times n}$, $b' \in \mathbb{R}^m$, $c' \in \mathbb{R}^n$, and $< \cdot, \cdot >$ denotes the usual inner product on $\mathbb{R}^n$.

First of all, we use either the so-called Gauss’ method (diagonalization by completion of the squares) or the diagonalizing procedure of Rosen et al. [16]. It is used to obtain an equivalent problem with separable cost quadratic functions, and in this way, we will be able to use the method described in the above section, since we will explicitly compute the value function ℓ defined in (17).

Therefore, we can suppose that the quadratic problem to solve has the following structure

$$\text{(QP)} \qquad \min_{y\in\Omega} \phi(y) =< Dy, y > + < c, y >, \tag{26}$$

where $\Omega = \{y \in \mathbb{R}^n : Ay \leqslant b\}$ is a bounded polyhedral set; $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix and $A \in \mathbb{R}^{m\times n}$; $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$.

On the other hand, from the compactness of the set Ω, we can suppose that we know a rectangular set $B = \{y \in \mathbb{R}^n : \underline{b} \leqslant y \leqslant \bar{b}\}$ such that $\Omega \subset B$; indeed if we define

$$\underline{b}_i = \min_{y\in\Omega} y_i \qquad \bar{b}_i = \max_{y\in\Omega} y_i \text{ for all } i \in \{1, \ldots, n\}, \tag{27}$$

B turns out to be the rectangular set of minimum volume that contains Ω.

We continue by using the Branch and Bound method described in Section 2. From now on, due to the separability of the quadratic cost function, we consider finite covers $\Lambda = \{R_j : j \in J\}$ of B such that R_j is a rectangular set for all $j \in J$. Branching refers to the partition of the feasible domain by bisection in certain directions, and bounding refers to the determination of lower and upper bounds for the global optimum value.

The Bounding. At the current iteration we have a finite rectangular cover $\Lambda = \{R_j : j \in J\}$ of B. We solve the dual programs

$$\gamma_j = \sup_{u\in\mathbb{R}^m_+} \ell_j(u) \text{ for all } j \in J, \tag{28}$$

where $\ell_j(u) = \min_{y\in R_j} L(y, u)$ and $L(y, u) = \phi(y)+ < u, Ay - b >$.

It is easy to see that

$$\ell_j(u) = - < u, b > + \sum_{i=1}^{n} \min_{y_i\in[l_i,L_i]} \{d_i y_i^2 + z_i(u) y_i\}, \tag{29}$$

where $z(u) = c + A^T u$, $D = \text{diag}(d_i)$ and $R_j = \prod_{i=1}^{n} [l_i, L_i]$.

For obtaining upper bounds we systematically test the middle point of the rectangular set $R_j = \prod_{i=1}^{n} [l_i, L_i]$, that is, if $\bar{y}_j = \frac{1}{2}(l + L) \in R_j$ verifies $A\bar{y}_j - b \leqslant 0$, then we set

$$\alpha(\Lambda) = \min\{\phi(\bar{y}_j), \alpha(\Lambda)\} \tag{30}$$

On the other hand, while we solve the programs (28), it is necessary to determine the value of $\ell(u)$, that is, the point $y' \in R_j$ such that $\ell(u) = \phi(y') + < u, Ay' - b >$, we then can test if this $y' \in R_j$ is feasible or not.

REMARK . Of course, we could obtain a feasible point $y \in R_j$ (if there exists) by solving the problem

$$\text{(LP)} \qquad \min\{h(y) : y \in R_j, y \in \Omega\} \tag{31}$$

where the function $h : \mathbb{R}^m \to \mathbb{R}$ is a linear (or piecewise linear) underestimation of ϕ on R_j. But, it introduces an additional computational cost.

The Branching. We take the set $R_{\hat{j}}$ defined by the bounding, that is, $R_{\hat{j}}$ verifies $\gamma_{\hat{j}} = \min_{j \in J} \gamma_j$. We bisect $R_{\hat{j}}$ on its largest face, that is, if $R_{\hat{j}} = \prod_{i=1}^{n} [l_i, L_i]$, we set $i_o \in \{1, \ldots, n\}$ such that

$$L_{i_o} - l_{i_o} = \max_{i \in \{1, \ldots, n\}} (L_i - l_i); \tag{32}$$

we then define the new cover Λ' of B by

$$\Lambda' = \Lambda \cup \{R^1, R^2\} \setminus \{R_{\hat{j}}\}, \tag{33}$$

where

$$R^1 = [l_{i_o}, \frac{l_{i_o} + L_{i_o}}{2}] \times \prod_{\substack{i=1 \\ i \neq i_o}}^{n} [l_i, L_i], \quad R^2 = [\frac{l_{i_o} + L_{i_o}}{2}, L_{i_o}] \times \prod_{\substack{i=1 \\ i \neq i_o}}^{n} [l_i, L_i]. \tag{34}$$

To end this section we establish a result showing that, first the above branching procedure verifies the remark following Theorem 2.2; second, giving two ways to leave out an element R_j of cover Λ of set B.

PROPOSITION 3.1. *Let $\Lambda = \{R_j : j \in J\}$ be a rectangular cover of the set B.*

(i) If we suppose that the set $R_{\hat{j}} = \prod_{i=1}^{n} [l_i, L_i] \in \Lambda$ must be branched, then the set R^k $k \in \{1, 2\}$ defined in (34) verifies the inequality

$$\operatorname{diam}(R^k) \leqslant \left(\frac{4n - 3}{4n}\right)^{1/2} \operatorname{diam}(R_{\hat{j}}). \tag{35}$$

(ii) Every set $R_j \in \Lambda$ such that $\gamma_j > \alpha(\Lambda) + \varepsilon$ may be left out since it does not contain global solutions of problem (QP).

(iii) If there exist $k_o \in \{1, \ldots, m\}$ *such that the set* $R_j = \prod_{i=1}^{n} [l_i, L_i] \in \Lambda$ *verifies*

$$\sum_{i=1}^{n} \min\{l_i a_i^{k_o}, L_i a_i^{k_o}\} > b_{k_o}, \tag{36}$$

where a^{k_o} *is the* k_o*-th row of matrix* A *and* b_{k_o} *is the* k_o*-th component of vector* $b \in \mathbb{R}^m$. *Then, the set* R_j *may be left out since it does not contain feasible points of problem* (QP).

Proof. It is a simple exercise. □

4. The Algorithm and the Computational Results

Based on the above theorical results, we now present the computational algorithm.

ALGORITHM . The elements of cover Λ are stored (in a growing order with respect to their lower bounds) in a two way-linked list **L**.

1. Initialization
   - 1.1. We use either Gauss' method or the diagonalizing procedure of Rosen et al. [16] to obtain an equivalent problem with a separable quadratic cost function.
   - 1.2. We solve the $2n$ linear programming problems

$$\underline{b}_i = \min_{y \in \Omega} y_i, \qquad \bar{b}_i = \max_{y \in \Omega} y_i. \tag{37}$$

   - 1.3. We define

     $\Lambda = \{B\}$ where $B = \prod_{i=1}^{n} [\underline{b}_i, \bar{b}_i]$,

     $\alpha(\Lambda) = \min_{i \in \{1,\ldots,n\}} \{\phi(\underline{y}_i), \phi(\bar{y}_i)\}$ where $\underline{y}_i$, $\bar{y}_i$ are (respectively) the solutions of linear programs in (37),

     $\gamma = \sup_{u \in \mathbb{R}^m_+} \ell(u)$ with ℓ defined in (28),

     $\mathbf{L} \leftarrow \{B\}$.

2. We bisect the first element (along its largest face) of list **L**, we obtain two elements R^1 and R^2 (defined in (34)).
3. For each $k \in \{1, 2\}$, we test if R^k contains feasible points (see (36)). If it is the case we compute $\gamma = \sup_{u \in \mathbb{R}^m_+} \ell(u)$, and the middle point $\bar{y}^k$ of R^k.

   If $\bar{y}^k \in R^k$ is a feasible point then

$$\alpha(\Lambda) \leftarrow \min\{\alpha(\Lambda), \phi(\bar{y}^k)\}. \tag{38}$$

4. We delete the first element of the list **L** and those elements of **L** such that their lower bound is greater than $\alpha(\Lambda) + \varepsilon$.
5. For each $k \in \{1, 2\}$, if necessary, we insert R^k in the list **L**.
6. If $\alpha(\Lambda) - \varepsilon$ is greater than the lower bound of the first element of **L** then we go to 2, otherwise we have an approximate solution $y^* \in \Omega$ of problem (QP) within ε, that is, $\phi(y^*) - \alpha \leqslant \varepsilon$.

REMARK . The dual problems

$$\gamma = \sup_{u \in \mathbb{R}^m_+} \min_{y \in R} L(y, u),$$

where L is defined in (28), are not solved in an exact form. For searching an underestimating of γ we use the fact that ℓ is a concave (continuous) piecewise quadratic function when R is a rectangular set, that is, $R = \prod_{i=1}^{n} [l_i, L_i]$; and D is a diagonal matrix. In fact, we recall that

$$\ell(u) = -< u, b > + \sum_{i=1}^{n} \min_{y_i \in [l_i, L_i]} \{d_i y_i^2 + z_i(u) y_i\},$$

where $z(u) = c + A^T u$ and $D = \mathrm{diag}(d_i)$.

If $d_i > 0$ and $l_i \leqslant \frac{z_i(u)}{2d_i} \leqslant L_i$, we then have the equality

$$\min_{y_i \in [l_i, L_i]} \{d_i y_i^2 + z_i(u) y_i\} = -\frac{(c_i + < u, a^i >)^2}{4d_i},$$

where $a^i \in \mathbb{R}^m$ is the i-th column of matrix A. It is then a quadratic function on a polyhedral domain in $\mathbb{R}^m$.

Otherwise, we have the equality

$$\min_{y_i \in [l_i, L_i]} \{d_i y_i^2 + z_i(u) y_i\} = \min\{d_i l_i^2 + z_i(u) l_i, d_i L_i^2 + z_i(u) L_i\}.$$

It is then a piecewise linear function on $\mathbb{R}^m$.

Next, we make no more than 30 iterations with a Uzawa's method. But since ℓ is a piecewise quadratic function, we use as ascent direction the variant of Polak and Ribière. Finally, the line search is realized with a bisection method. For details on these methods see Minoux [9], Part I, chapters 3 (p. 75), 4 (p. 110) and 6 (p. 253).

Computational Results

The computational results presented below were all obtained using a DEC ALPHA 3000/300L. The DEC ALPHA has a DECchip 21064 RISC-style microprocessor,

Table I.

	n=100	n=300	n=500	n=1000	n=2000	n=3000	n=4000
CPU time (secs.)	0.0	0.1	0.3	1.5	8.2	17.6	32.0

with a 256 KB secondary cache, a SIMM memory of 128 MB and a CPU speed of 100 Mhz.

TEST PROBLEM 1

It is the following concave separable quadratic problem:

$$\min\left\{ -\sum_{i=1}^{n}\left[x_i^2+\frac{1}{2}x_i\right]:\ -1.0\leqslant x_i\leqslant 1.0, i\in\{1,\ldots,n\}\right\}.$$

It is easy to see that this problem has 3^n Karush-Kuhn-Tucker points, 2^n local minima points and only one global solution.

For every problems the number of iterations was 1 and the number of cost function evaluations was 5.

Table 1 summarizes the computational results with $\varepsilon = 5\cdot 10^{-7}$.

TEST PROBLEM 2

It is the following integer 0-1 linear problem:

$$\max\left\{\sum_{i=1}^{n} c_i y_i:\ \sum_{i=1}^{n} y_i\leqslant b,\ y_i\in\{0,1\},\ i\in\{1,\ldots,n\}\right\},$$

where $c_i = 1.0$ for all $i\in\{1,\ldots,N\}$; $c_i=-1.0$ for all $i\in\{N+1,\ldots,n\}$; and $b=\max\{q : q\in\mathbb{N}, q\leqslant N/2.0\}$. It is easy to see that the number of solutions of this problem is greater than 2^b.

We solve the following concave separable quadratic problem:

$$-\min\left\{\sum_{i=1}^{n}[\theta y_i(1-y_i)-c_i y_i]:\sum_{i=1}^{n} y_i\leqslant b,\ 0\leqslant y_i\leqslant 1,\ i\in\{1,\ldots,n\}\right\},$$

where θ is a constant positive. It is not difficult to see that if we wish to obtain one solution to distance less or equal that $\varepsilon' > 0$ of the integer 0-1 solution, it is sufficient that θ verify

$$\theta\geqslant\frac{\varepsilon}{\varepsilon'(1-\varepsilon')}$$

Table II.

N	n=1000 time	n=2000 time	n=3000 time	fn	dual	iter	node
100	6.0	17.1	31.0	354	1057	75	76
200	11.5	31.1	51.5	755	2457	175	176
300	13.1	31.8	50.3	652	2100	150	151
400	22.5	56.1	90.6	1555	5257	375	376
500	21.1	52.8	75.2	1052	3500	250	251
600	25.2	62.2	90.1	1252	4200	300	301
700	39.0	84.4	149.0	10885	17631	1258	1259
800	44.5	107.7	168.6	12085	19844	1416	1417
900	37.3	93.6	123.1	1853	6300	450	451
1000	41.4	99.9	136.3	2052	7000	500	501
1500	——	128.6	207.6	3053	10500	750	751
2000	——	174.7	259.0	4055	14002	1000	1001

Note: time = CPU time (secs), fn = maximum number of cost function evaluations, dual = maximum number of dual function evaluations, iter = maximum number of iterations, node = maximum number of stored nodes.

where $\varepsilon > 0$ is the threshold for the approximated solution.

Table 2 summarizes the computational results with $\theta = 10^3$ and $\varepsilon = 5 \cdot 10^{-7}$.

TEST PROBLEM 3

It is the following concave separable quadratic problem:

$$\min \left\{ -\sum_{i=1}^{n} y_i^2 : \quad \sum_{i=1}^{j} y_i \leqslant j, \; j \in \{1, \ldots, n\}, \; y_i \geqslant 0, i \in \{1, \ldots, n\} \right\}.$$

This problem is taken from Strekalovskii [18]. In that paper the CPU time reported to solve the above problem is 40 min for 100 variables.

It is important to note that the diameter of the feasible set and the norm of the global solution are equals to the dimension of the problem.

Table 3 below summarizes the computational results with $\varepsilon = 5 \cdot 10^{-7}$

TEST PROBLEM 4

It is the nonconvex separable quadratic problem:

$$\min_{y \in \Omega} \left\{ -\frac{1}{2} \sum_{i=1}^{k} \lambda_i (y_i - \bar{y}_i)^2 + \frac{1}{2} \sum_{i=k+1}^{k+l} \lambda_i (y_i - \bar{y}_i)^2 \right\},$$

Table III.

n	time	fn	dual	iter	node
100	0.3	26	46	1	1
200	3.8	80	107	2	2
300	7.7	63	84	2	2
400	12.5	57	71	1	1
500	30.1	71	102	3	2
600	55.9	100	133	4	2
700	33.0	28	55	1	1
800	74.8	78	93	1	1
900	77.8	50	70	1	1
1000	125.2	63	95	2	2
1500	346.4	87	103	1	1
2000	647.9	76	106	2	2

Note: See note Table 2.

where $\Omega = \{ y \in \mathbb{R}^n : \ Ay \leqslant b, \ y \geqslant 0 \}$, $n = k + l$, $A = (a_{ji}) \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. The data $a_{ji} \in [0, 9]$, $b_j \in [0, n]$ and $\bar{y}_i, \ \lambda_i \in [0, 99]$ are randomly generated integers. We solved 17 problems (5 for each dimension), the smallest of dimension 60 (10 concave and 50 convex variables) and the largest of dimension 330 (30 concave and 300 convex variables). The average CPU time was 2.6 secs for n=60, to 723.6 secs for n=170, problem.

The PGR column contains the CPU times (obtained using a Cray 1S supercomputer) given in Pardalos et al. [12] to solve problems of the same dimension and data generated similarly.

Table 4 below summarizes the computational results with $\varepsilon = 5 \cdot 10^{-3}$.

5. Concluding Remarks

In our approach, we note that the main computational effort required to obtain good approximations of the global solution of the quadratic problem depends mainly on our capacity to store the information, that is, on our capacity to store the cover generated by the branching; and on our efficiency to calculate the dual function values.

The techniques developed above can be extended to the solution of:

(1) 5.1 Large scale nonconvex quadratic problems of the form

$$\min\{< Qy, y > + < c, y > + < d, x >: \ (y, x) \in \Omega\},$$

Table IV.

k	l	n	min	max	aver	fn	dual	iter	node	PGR
10	50	30	0.0	9.2	2.6	938	2570	49	4	54.6
10	50	20	2.2	89.4	47.1	24534	83217	1777	61	56.6
10	60	20	0.1	64.6	14.1	17738	61018	2594	51	154.0
10	60	30	0.9	67.4	15.5	16803	39431	1460	87	48.5
10	70	30	2.1	274.5	115.7	46545	158750	3556	139	80.5
10	80	40	0.2	21.5	7.1	2471	7204	132	4	77.8
10	100	50	0.1	19.3	6.6	1442	4696	93	4	89.5
20	40	20	0.0	66.0	21.7	16578	67439	2528	85	97.5
20	50	20	0.0	30.7	6.2	14623	27085	1330	27	305.5
20	60	30	0.0	215.6	98.6	32393	129571	2381	113	208.8
20	70	30	0.2	479.3	123.8	186805	171493	22818	554	199.0
40	80	40	0.1	273.1	86.4	27248	73525	4650	68	457.0
20	150	30	6.5	725.9	177.3	41625	154246	3358	31	325.0
20	150	50	0.5	2224.8	723.6	90718	312297	8361	175	436.0
30	150	50	0.4	1117.8	327.5	53096	143617	6166	59	1000.0
20	200	50	0.4	363.2	154.0	13381	38141	1096	16	178.0
30	300	50	9.6	1889.0	452.9	57902	108567	7393	77	1500.0

Note: k = number of concave variables, l = number of convex variables, m = number of constraints, min = minimum CPU time (secs), max = maximum CPU time (secs), aver = average CPU time (secs), fn = maximum number of cost function evaluations, dual = maximum number of dual function evaluations, iter = maximum number of iterations, node = maximum number of stored nodes.

where $\Omega = \{(y, x) \in \mathbb{R}^{n+k} : Ay + Bx \leqslant b\}$ is a bounded polyhedral set, $Q \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $y \in \mathbb{R}^n$ and $x \in \mathbb{R}^k$. Here k may be much larger than n.

(2) 5.2 Integer 0-1 (or mixed) nonconvex quadratic problems of the form

$$\min\{< Qy, y > + < c, y >: \; y \in \Omega, \; y_i \in \{0, 1\}, i \in I\},$$

where $\Omega = \{y \in \mathbb{R}^n : Ay \leqslant b\}$ is a bounded polyhedral set, $Q \in \mathbb{R}^{n\times n}$ is a symmetric matrix and $I \subset \{1, \dots, n\}$.

References

1. Auslender, A. (1976), *Optimisation Méthodes Numériques*, Masson, France.
2. Ben-Tal, A., Eiger, G. and Gershovitz, V. (1994), Global minimization by reducing the duality gap, Mathematical Programming 63: 193–212.
3. Bomze, I. and Danninger, G. (1994), A finite algorithm for solving general quadratic problem, *Journal of Global Optimization* 4: 1–16.
4. Falk, J.E. (1969), Lagrange Multipliers and Nonconvex Programs, *SIAM J. Control* 7: 534–545.
5. Geoffrion, A. (1972), Generalized Benders' decompositions, *Journal of Optimization, Theory and Its Applications* 10: 237–260.

6. Horst, R. and Tuy, H. (1993), *Global Optimization (Deterministic Approaches)*, 2nd edition. Springer-Verlag, Berlin.
7. Kalantari, B. (1984), *Large scale concave quadratic minimization and extensions*, PhD thesis. Computer Sci. Dept., University of Minnesota.
8. Kough, P.F. (1979), The indefinite quadratic programming problem, *Operations Research* 27/3: 516–533.
9. Minoux, M. (1983), *Programmation mathématique: Théorie et algorithmes*, Vol. 1, Dunod, France.
10. Muller, R.K. (1979), A method for solving the indefinite quadratic programming A problem, *Management Science* 16/5: 333–339.
11. Murty, K.G. and Kabadi, S.N. (1987), Some NP-complete in quadratic and nonlinear programming, *Mathematical Programming* 39: 117–129.
12. Pardalos, P.M., Glick, J.H. and Rosen, J.B. (1987), Global minimization of indefinite quadratic problems, *Computing* 39: 281–291.
13. Pardalos, P.M. and Rosen, J.B. (1987), *Constrained global optimization: Algorithms and Applications*, Lecture Notes in Computer Science 268. Springer-Verlag, Berlin.
14. Pardalos, P.M. and Schnitger, G. (1988), Checking local optimality in constrained quadratic programming is NP-hard, *OR Letters* 7: 33–35.
15. Phillips, A.T. and Rosen, J.B.(1988), A parallel algorithm for constrained concave quadratic global minimization, *Mathematical Programming* 42: 421–488.
16. Rosen, J.B. and Pardalos, P.M. (1986), Global minimization of large scale constrained concave quadratic problems by separable programming, *Mathematical Programming* 34: 163–174.
17. Shectman, J.P. and Sahinidis, N.V.(1996), A finite algorithm for global minimization of separable concave programs, in: Floudas, C.A. and Pardalos, P.M. (eds.), *State of the Art in Global Optimization: Computational Methods and Applications*, Kluwer Academic Publishers, Boston, MA, 303–340.
18. Strekalovskii, A.S. (1993), The search for a global maximum of a convex functional on a admissible set, *Comput. Maths. Math. Phys.* Vol. 33, No. 3: 315–328.
19. Tuy, H. (1983), Global minimization of the difference of two convex functions, *Select Topics in Operations Research and Mathematical Economics, Lecture Notes Econ. Math. Syst.* 226: 98–118.