Journal of Global Optimization **13:** 171–206, 1998.

A Branch and Bound Method via d.c. Optimization Algorithms and Ellipsoidal Technique for Box Constrained Nonconvex Quadratic Problems

LE THI HOAI AN and PHAM DINH TAO
Mathematical Modelling and Applied Optimization Group, Laboratory of Mathematics, National Institute for Applied Sciences- Rouen, CNRS UPRES-A 60 85, BP 08, F-76131 Mont Saing Aignan, France

(Received 20 May 1996; accepted 15 October 1997)

Abstract. In this paper we propose a new branch and bound algorithm using a rectangular partition and ellipsoidal technique for minimizing a nonconvex quadratic function with box constraints. The bounding procedures are investigated by d.c. (difference of convex functions) optimization algorithms, called DCA. This is based upon the fact that the application of the DCA to the problems of minimizing a quadratic form over an ellipsoid and/or over a box is efficient. Some details of computational aspects of the algorithm are reported. Finally, numerical experiments on a lot of test problems showing the efficiency of our algorithm are presented.

Key words: Branch and bound, Ellipsoidal technique, d.c. Optimization, DCA, Ball constrained quadratic problem, Box constrained quadratic problem

1. Introduction

We shall be concerned with the well-known problem

$$\text{(QB)} \quad \min \left\{ f(x) := \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle : \right.$$
$$\left. -\infty < l_i \le x_i \le u_i < +\infty, \quad i = 1, \dots, n \right\}$$

with A being an $(n \times n)$ symmetric matrix, and b, $x \in \mathbb{R}^n$. A special case of (QB) is the unconstrained quadratic zero-one problem which has many important applications, particularly in combinatorial optimization, and is formulated as follows

$$\min \left\{ f(x) := \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle : x \in \{0, 1\}^n \right\}. \tag{1}$$

Indeed, writing (1) in the form

$$\min\left\{\frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle + \frac{t}{2}(\langle e, x\rangle - \langle x, x\rangle) : x \in \{0,1\}^n\right\}, \tag{2}$$

we have

$$(2) \Leftrightarrow \min\left\{\frac{1}{2}\langle x, (A - tI)x\rangle + \left\langle b + \frac{t}{2}e, x\right\rangle : x \in [0,1]^n\right\}, \tag{3}$$

where e is the vector of ones and t is a real scalar such that $A - tI$ is negative semidefinite.

Several algorithms have been proposed for globally solving these problems [see, e.g., 3, 8, 9, 11, 15–17, 30–32, 40, 42]. More generally, there exists in the literature a lot of methods for solving the nonconvex quadratic programming, among them branch-and-bound algorithms whose branching operation takes places only in the 'negative eigenvalues space' have been shown to be efficient [see e.g. 4, 20, 39, 41, 53].

The procedures often used in the existing methods are to relax the constraints and to perturb the objective function. The branching is usually base dupon rectangular and/or simplicial partitions.

Ellipsoidal techniques have been used to state estimations for optimal values in nonconvex quadratic programming [21–23, 53, 54]. In Vavasis [53] an approximation algorithm was proposed for finding an ε-approximate solution. It was shown that such an approximation can be found in polynomial time for fixed ε and t, where t denotes the number of negative eigenvalues of the quadratic term. In a series of paper by N. Karmarkar et al. [21–23], continuous methods for integer quadratic programs (more precisely the linear inequality $\{-1, 1\}$ satisfiability problem which is equivalently translated into maximizing the square of Euclidean norm on $[-1, 1]^n$ subject to linear inequalities) have been investigated. These methods used ellipsoidal techniques (each iteration requires solving a nonconvex quadratic program over an ellipsoid), interior point algorithm [22], and Riemannian metric for defining the search region [21, 23].

In [23] a new variant is introduced Karmarkar et al. [22] where the Riemannian metric used for finding the search region is dynamically modified. Their approaches are basically different from ours. For globally solving a nonconvex quadratic program on an ellipsoid, N. Karmarker et al. used an adapted version of Moré-Sorensen's algorithm [29]. It has been recognized that the latter is expensive to implement (because it uses Cholesky factorization procedure) and non suitable to the large scale setting. The recent powerful algorithms by Sorensen [50], Santos & Sorensen [47, an improved version of 50] and by Rendl & Wolkowicz [45] aimed to remedy the above mentioned drawback.

The method we study in this paper is a combination of the branch and bound algorithm and the d.c. optimization approach. The d.c. optimization algorithm (called DCA) introduced by Pham Dinh Tao [33], is a primal-dual subdifferential method

for solving a general d.c. program of the form

$$(P_{dc}) \quad \alpha = \inf\{F(x) = g(x) - h(x) : x \in \mathbb{R}^n\}, g, h \in \Gamma_0(\mathbb{R}^n)$$

and its dual given by

$$(D_{dc}) \quad \alpha = \inf\{h^*(y) - g^*(y) : y \in Y\}.$$

Here $\Gamma_0(\mathbb{R}^n)$ is the set of all proper lower semi-continuous convex functions on $\mathbb{R}^n$ which is equipped with the canonical inner product $\langle \cdot, \cdot \rangle$, Y is the dual space of $\mathbb{R}^n$ which can be identified with $\mathbb{R}^n$ itself, g^* is the conjugate function belonging to $\Gamma_0(Y)$ and is defined by

$$g^*(y) = \sup\{\langle x, y\rangle - g(x) : x \in \mathbb{R}^n\}.$$

Based on the d.c. duality and the local optimality, the DCA consists in the construction of two sequences $\{x^k\}$ and $\{y^k\}$ such that x^k (resp. y^k) is a solution to the convex program (P_k) (resp. (D_k)) defined by

$$\min\{g(x) - [h(x^k) + \langle x - x^k, y^k\rangle] : x \in X\}, \quad (P_k)$$

$$\min\{h^*(y) - [g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle] : y \in Y\}. \quad (D_k)$$

In fact, (P_k) (resp. (D_k)) is obtained from (P_{dc}) (resp. (D_{dc})) by replacing h (resp. g^*) with its affine minorization defined by $y^k \in \partial h(x^k)$ (resp. $x \in \partial g^*(y^{k-1})$). In other words, starting with a x^0 in dom ∂h, the DCA reads

$$y^k \in \partial h(x^k); \quad x^{k+1} \in \partial g^*(y^k).$$

By this way, the sequences $\{x^k\}$ and $\{y^k\}$ satisfy the following conditions [1, 33, 37]:

(i) The sequences $\{g(x^k) - h(x^k)\}$ and $\{h^*(y^k) - g^*(y^k)\}$ are decreasing and
- $g(x^{k+1}) - h(x^{k+1}) = g(x^k) - h(x^k)$ if and only if $y^k \in \partial g(x^k) \cap \partial h(x^k)$, $y^k \in \partial g(x^{k+1}) \cap \partial h(x^{k+1})$ and $[\rho(g) + \rho(h)]\|x^{k+1} - x^k\| = 0$.
- $h^*(y^{k+1}) - g^*(y^{k+1}) = h^*(y^k) - g^*(y^k)$ if and only if $x^{k+1} \in \partial g^*(y^k) \cap \partial h^*(y^k)$, $x^{k+1} \in \partial g^*(y^{k+1}) \cap \partial h^*(y^{k+1})$ and $[\rho(g^*) + \rho(h^*)]\|y^{k+1} - y^k\| = 0$.

  ($\rho(g)$ denotes the modulus of strong convexity of g).

(ii) Every limit point x^* (resp. y^*) of the sequence $\{x^k\}$ (resp. $\{y^k\}$) is a critical point of $g - h$ (resp. $h^* - g^*$).

(Remember that a point x is called *critical point* of $g - h$ if $\partial g(x) \cap \partial h(x) \neq \emptyset$).

Important and interesting questions in the use of the DCA are the choice of the d.c. decomposition of the objective function and the initial point x^0 to ensure convergence of the DCA to a global minimizer. They are open questions to be studied for the specific structure of the problem being considered. This d.c. optimization

approach has been improved and completed from both theoretical and numerical viewpoints after the works of the authors of this paper appeared in which the DCA has been successfully applied to many large-scale d.c. optimization problems [see, e.g., 1, 2, 37 and references therein].

The DCA plays a key role in this paper. The first use of the DCA deals with the lower bound computation scheme. We apply the DCA to solve the problem (QE) which is obtained from (QB) by replacing the box constraint with an ellipsoid containing the selected box. Our main motivation for the use of this technique is that the DCA is very efficient for globally minimizing a quadratic form over an Euclidean ball [37]. The second use concerns the upper bound computation process. For finding a good quality of feasible solutions we apply the DCA to solving (QB) from good starting points which are obtained while computing lower bounds. Having its source in the theory of d.c. optimization, DCA is a new algorithmic concept for solving d.c. programs. For both box constrained nonconvex quadratic programs and ball constrained nonconvex quadratic programs DCA requires only matrix-vector produces and proved to be very efficient and robust [1, 37]. In particular for the latter (which constitutes an important step in our algorithm) many numerical experiments have shown the efficiency and robustness of DCA with respect to Sorensen's algorithm [50], Santos–Sorensen's algorithm [47] and Rendl–Wolkowicz's algorithm [45].

In the next section we describe the branch and bound scheme for solving (QB). First, we present in Subsection 2.1 the procedure for computing lower bounds. This is based on the DCA for the ball constrained quadratic problem. In Subsection 2.2 we study the DCA for minimizing a quadratic form over a box. The branching procedure is discussed in the next subsection. In the last subsection we provide the summary of the branch and bound algorithm and its convergence. In Section 3 the implementation of our algorithm and some illustrative examples to it are given. Using an affine linear transformation we propose a very simple implementation of the algorithm which does not need a large memory storage. A variant of our branch and bound algorithm is presented in Section 4. There, we introduce a second process for lower bounding in which a convex underestimation of the objective function is considered. Finally, in Section 5 we present computation results on several classes of test problems, among them is Spin-glass problem which has an important application in physics.

2. A branch and bound algorithm via DCA and ellipsoidal technique (BDCE)

The algorithm we are going to describe starts with an ellipsoid E^0 containing the feasible region K^0 of (QB). If a minimizer of f over E^0 is feasible then we stop the process. Otherwise, we divide K^0 into two rectangles (rectangular bisection) and construct two ellipsoids such that each of them contains one of the just divided rectangles. To improve the lower bound we compute minima of f on each newly

generated ellipsoid. The procedure is then repeated by replacing K^0 by a rectangle corresponding to the smallest lower bound and E^0 with an ellipsoid containing this rectangle. As this procedure repeats infinitely many times two infinite nested sequences of rectangles and ellipsoids are generated and shrink to a singleton. The convergence of the algorithm thus is ensured.

2.1. LOWER BOUND – GLOBALLY SOLVING BALL CONSTRAINED QUADRATIC PROBLEM BY THE DCA

For general polyhedral convex sets K^k it is easier to use the technique of halving ellipsoids for constructing the ellipsoid E^k (of minimal volume) containing K^k. In our problem the simple rectangular shape of K^k makes possible the following construction of E^k which seems to be better (at least in practice) than the Löwner-John process.

Let S be the ball with the centre at origin and of radius $\sqrt{n}$ containing the rectangle $T = [-1, 1]^n$

$$S = \{x \in \mathbb{R}^n : \|x\| \leq \sqrt{n}\}.$$

Using the bijective affine transformation L such that $T = LK$, we can define an ellipsoid E containing the rectangular $K = \prod_{i=1}^n [a_i, t_i]$ as follows:

$$E = \{x \in \mathbb{R}^n : \langle D^2 x, x\rangle + 2\langle Dc, x\rangle \leq n - \langle c, c\rangle\} = L^{-1}S, \tag{4}$$

where $D = \operatorname{diag}(d_1, \ldots, d_n)$, $d, c \in \mathbb{R}^n$ and satisfy

$$d_j = 2/(t_j - a_j),\ j = 1, \ldots, n, \tag{5}$$

$$c_j = (a_j + t_j)/(a_j - t_j),\ j = 1, \ldots, n. \tag{6}$$

Moreover, we obtain the following connection between the volumes of E and S:

$$\operatorname{vol}(E) = \frac{\operatorname{vol}(S)}{d_1 d_2 \cdots d_n}. \tag{7}$$

Since the subrectangles $K^k = \prod_{i=1}^n [a_i^k, t_i^k]$ shrink to a singleton, the numbers $d_i^k = 2/(t_i^k - a_i^k)$ for $i = 1, \ldots, n$ tend to $+\infty$. Hence the sequence $\{\operatorname{vol}(E^k)\}$ decreases to zero as $k \to +\infty$.

The lower bound of f in our algorithm is improved by the solution of the problem

$$\text{(QE)} \quad \min\left\{\frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle : x \in E\right\}.$$

Again by the transformation L one has $x \in E \Leftrightarrow y = Lx \in S$, so the last problem is equivalent to

$$\min\left\{\frac{1}{2}\langle y, D^{-1}AD^{-1}y\rangle + \langle D^{-1}q - D^{-1}AD^{-1}c, y\rangle : \|y\|^2 \leq n\right\}. \tag{8}$$

Clearly, $D^{-1}AD^{-1}$ is an $(n \times n)$ symmetric matrix.

Now we study the general ball constrained quadratic problem in the form

$$\text{(QBA)} \quad \min \left\{ F(x) := \frac{1}{2}\langle x, Qx\rangle + \langle q, x\rangle : \|x\| \leq r \right\}.$$

(As before Q is a $(n \times n)$ symmetric matrix, $q \in \mathbb{R}^n$ and $r > 0$.)

First, we give a brief history of this problem which is often called the trust-region subproblem. It arises as the main subproblems in a class of nonlinear programming algorithms called model trust region methods which have been introduced by Levenberg [25], Marquardt [28] and have been successfully developed in the works of Gay [13], Moré & Sorensen [29, 48], Pham Dinh Tao and his coworkers [34–37]. To our knowledge the first efficient (primal-dual) algorithm for solving (QBA) is due to Moré & Sorensen [29]. These authors used safeguarding technique combined with an adapted Newton method (due to Hebden) for computing a saddle-point of the Lagrangian function $L(x, \lambda)$. In the same primal-dual approach Pham Dinh Tao et al. proposed in 1989 [34, 35] a dichotomy process as an alternative for the safeguarding technique of Moré–Sorensen. According to many comparative numerical simulations [34, 35], the dichotomy algorithm seems to be less expensive than the safeguarding one. A simple bisection method was proposed by Ye [54] and Vavasis & Zippel [51]. Recently, matrix-free methods [45, 47, 50] have been introduced to solve large-scale problems. These methods recast the trust-region problems in terms of a parametrized eigenvalue problem and only require matrix-vector products.

While minimizing a nonpositive semi-definite quadratic function over a polyhedral convex set has been shown to be NP-hard, minimizing the same objective function on an Euclidean ball can be solved effectively by some polynomial time algorithms [see 52 and the references therein]. We can explain this 'luck' by the fact that Problem (QBA) is one of few nonconvex programs which possess, as in convex case, a complete characterization of its solutions.

THEOREM 1 [13, 29, 34, 36]. *x^* is a global optimal solution to (QBA) if and only if there is a real number $\lambda^* \geq 0$ such that:*

(i) $(Q + \lambda^* I)x^* = -q$,

(ii) $\|x^*\| \leq r, \lambda^*(\|x^*\| - r) = 0$,

(iii) $(Q + \lambda^* I)$ *is positive semi-definite.*

Such a λ^* is unique.

A point x^* satisfying the first two conditions is called Kuhn–Tucker point for (QBA), the corresponding nonnegative number λ^* is called Lagrangian multiplier associated with x^*. It is uniquely defined by

$$\lambda^* = -[\langle Qx^*, x^*\rangle + \langle q, x^*\rangle]/r^2. \tag{9}$$

Finally (λ^*, x^*) is called Kuhn–Tucker pair for (QBA).

In this paper, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the eigenvalues of Q and ϕ is a function defined on $\mathbb{R}\backslash\{\lambda_i : i = 1, \ldots, n\}$ by $\phi(\lambda) = \|x(\lambda)\|$ where $x(\lambda)$ is the solution of $(Q + \lambda I)x = -q$.

The following result of Martinez [27] emphasizes the relationship between local and global solutions to (QBA).

THEOREM 2.

(i) *If x^* is local-nonglobal minimum for (QBA), then $(Q + \lambda^* I)x^* = -q$ with nonnegative $\lambda^* \in]-\lambda_2, -\lambda_1[$ and $\phi'(\lambda^*) \geq 0$.*

(ii) *There exists at most one local-nonglobal minimum for (QBA).*

(iii) *If $\|x^*\| = r$, $(Q + \lambda^* I)x^* = -q$ for some nonnegative $\lambda^* \in]-\lambda_2, -\lambda_1[$ and $\phi'(\lambda^*) > 0$, then x^* is a strict local minimum for (QBA).*

(iv) *If q is orthogonal to some eigenvector associated with λ_1, then there is no local-nonglobal minimum for (QBA).*

In d.c. optimization approach we propose here a new algorithm for globally solving (QBA). Although the DCA is not a polynomial time algorithm, numerical experiments [1, 2, 37] indicated that this method works very well in practice. The DCA is an iterative method which is quite different from known related algorithms. Thanks to the particular structure of the problem under consideration, normally the DCA becomes very simple (it requires only matrix – vector products) and in practice converges to a global solution. For checking the global optimality of solutions provided by the DCA we use the inexpensive *Implicit Restarted Lanczos Method* (IRLM) of Sorensen [49] to compute the smallest eigenvalue of the related matrix Q and a corresponding eigenvector. A simple numerical procedure has been introduced (in case of nonglobal solutions) to find a feasible point having a smaller objective and to restart the DCA with this point. It has been stated that in nonconvex case (Q is nonpositive semidefinite) the DCA with at most $2m + 2$ restarts (m is the number of distinct negative eigenvalues of Q) requires only matrix-vector products too and converges to a global solution. This fact (see Remark 3) is guaranteed by Theorem 3.

To explain the nature of DCA's convergence, let us state the following features of Kuhn–Tucker points for (QBA).

LEMMA 1. *If $F(x) = 1/2\langle x, Qx\rangle + \langle q, x\rangle$ with Q being symmetric and $(Q + \mu I)y = -q$, then $F(x) = F(y) - \mu/2(\|x\|^2 - \|y\|^2) + 1/2\langle x - y, (Q + \mu I)(x - y)\rangle$, $\forall x$.*

Proof. We have

$$\begin{aligned} F(x) =& F(y) + \langle x - y, Qy + q\rangle + \frac{1}{2}\langle x - y, Q(x - y)\rangle \\ =& F(y) - \mu\langle x - y, y\rangle + \frac{1}{2}\langle x - y, (Q + \mu I)(x - y)\rangle \\ & - \frac{\mu}{2}\langle x - y, x - y\rangle. \end{aligned}$$

Since $\mu/2\langle x-y, x-y\rangle = \mu/2(\|x\|^2 + \|y\|^2) - \mu\langle x, y\rangle$, we get

$$-\frac{\mu}{2}\langle x-y, x-y\rangle - \mu\langle x-y, y\rangle = -\frac{\mu}{2}(\|x\|^2 + \|y\|^2) + \mu\|y\|^2 = -\frac{\mu}{2}(\|x\|^2 - \|y\|^2),$$

which proves the lemma. □

LEMMA 2. *Let (μ_1, x_1) and (μ_2, x_2) be Kuhn–Tucker pairs for (QBA), then $\mu_1 = \mu_2$ if and only if $F(x_1) = F(x_2)$.*

Proof. If $\mu_1 = \mu_2 = \mu$, then according to Lemma 1

$$F(x_2) = F(x_1) - \frac{\mu}{2}(\|x_2\|^2 - \|x_1\|^2) + \frac{1}{2}\langle x_2 - x_1, (Q + \mu I)(x_2 - x_1)\rangle.$$

But $(Q + \mu I)(x_2 - x_1) = 0$ because $(Q + \mu I)x_1 = -q = (Q + \mu I)x_2$. Hence

$$F(x_2) + F(x_1) - \frac{\mu}{2}(\|x_2\|^2 - \|x_1\|^2).$$

Since either $\mu = 0$ or $\|x_1\| = \|x_2\| = r$, we have $F(x_1) = F(x_2)$.

Conversely suppose that $F(x_1) = F(x_2)$. We can assume without loss of generality that $x_1 \neq x_2$ because $x_1 = x_2$ implies $\mu_1 = \mu_2$ in virtue of (9). Consider first the case where μ_1 and μ_2 are positive. It follows from Lemma 1 that

$$F(x_2) = F(x_1) + \frac{1}{2}\langle x_2 - x_1, (Q + \mu_1 I)(x_2 - x_1)\rangle$$

because $\mu_1 > 0$ and $\mu_2 > 0$ imply $\|x_1\| = r = \|x_2\|$. Hence

$$\mu_1 = \frac{-\langle x_2 - x_1, Q(x_2 - x_1)\rangle}{\|x_2 - x_1\|^2}. \tag{10}$$

Likewise we get

$$\mu_2 = \frac{-\langle x_1 - x_2, Q(x_1 - x_2)\rangle}{\|x_1 - x_2\|^2}. \tag{11}$$

So $\mu_1 = \mu_2$.

It remains to examine the case where μ_1 or μ_2 is zero, say μ_1 for instance. As above we have (10) and hence

$$\langle x_2 - x_1, Q(x_2 - x_1)\rangle = 0.$$

On the other hand we can write

$$F(x_1) = F(x_2) - \frac{\mu_2}{2}(\|x_1\|^2 - \|x_2\|^2) + \frac{1}{2}\langle x_1 - x_2, (Q + \mu_2 I)(x_1 - x_2)\rangle.$$

That is reduced to

$$\mu_2(\|x_2\|^2 - \|x_1\|^2) + \mu_2\|x_2 - x_1\|^2 = 0.$$

If $\mu_2 > 0$, then $\|x_2\| = r$ and

$$\mu_2(r^2 - \|x_1\|^2) + \mu_2\|x_2 - x_1\|^2 > 0.$$

The proof is then complete. □

Let us indicate here that the statement in Lemma 2 has been earlier stated in [26]. '$F(x_1) = F(x_2)$ for Kuhn–Tucker pairs (μ, x_1) and (μ, x_2)'.

We shall state now the main result of Kuhn–Tucker points for (QBA) which is useful to appreciate DCA's convergence:

THEOREM 3. *The number p of distinct multipliers associated with Kuhn–Tucker points for (QBA) is equal to the number of distinct values of the objective function $F(x)$ at Kuhn–Tucker points. Moreover p is at most* $\min\{2m + 2, 2n + 1\}$, *where m is the number of distinct negative eigenvalues of Q.*

Proof. According to the above displayed results, it suffices to prove that $p \leq \min\{2m + 2, 2n + 1\}$. As in [26], the idea follows from a result of Forsythe and Golub [12]. For the sake of completeness we give below a sketch of the proof.

Let V be an orthogonal matrix composed of eigenvectors associated with the eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ of Q and Δ be the diagonal matrix whose entries are $\lambda_1, \ldots, \lambda_n$ such that $V^T Q V = \Delta$. Let λ be a positive multiplier for (QBA). The linear equation

$$(Q + \lambda I)x = -q \tag{12}$$

must have at least a solution such that $\|x\| = r$. In other words, with the change of variables $u = V^T x$ and $d = V^T q$, the problem

$$(\Delta + \lambda I)u = -d, \quad \|u\| = r \tag{13}$$

must have a solution.

Finally, λ must be a solution of the nonlinear equation

$$\phi(\lambda) = r, \quad \lambda \geq 0, \tag{14}$$

where

$$\phi(\lambda) = \left[\sum_{i=1}^{n} \frac{d_i^2}{(\lambda_i + \lambda)^2}\right]^{1/2}.$$

Main properties of $\phi(\lambda)$ have been investigated in [36]. It tells us, in particular, that $\phi \in C^\infty(\mathbb{R}\backslash\{-\lambda_i : i = 1, \ldots, n\})$ and is strictly convex on each open interval

contained in $\mathbb{R}\backslash\{-\lambda_i : i = 1, \ldots, n\}$. The function $\phi(\lambda)$ has poles at $-\lambda_i$, $(i = 1, \ldots, n)$ and tends to zero as λ tends to $\pm\infty$. Thus the equation $\phi(\lambda) = \Gamma$ has at most 2 roots in each interval $]-\lambda_i, -\lambda_{i-1}[$ and one root in each extreme interval. If $\lambda_i > 0$ for all $i = 1, \ldots, n$, then there is at most one nonnegative root; if $\lambda_i < 0$ for all $i = 1, \ldots, n$, then there are at most $2m + 1$ nonnegative roots. Hence the problem (14) has at most $\min\{2m + 1, 2n\}$ nonnegative roots. Taking into account the possible case of Kuhn–Tucker pairs (λ, x) with $\lambda = 0$, we can conclude that there are at most $\min\{2m + 2, 2n + 1\}$ multipliers for (QBA). □

The next result follows from the preceding ones:

COROLLARY 1. *Let $0 \leq \mu_1 < \cdots < \mu_p$ be the set of Lagrangian multipliers for (QBA). Let $\mathcal{KT}$ (resp. $\mathcal{KT}(\mu_i)$) be the set of Kuhn–Tucker points for (QBA) (resp. the set of Kuhn–Tucker points associated with μ_i). We have*

(i) $\mathcal{KT} = \cup_{i=1}^{p} \mathcal{KT}(\mu_i)$.

(ii) *$\mathcal{KT}(\mu_i)$ is compact and identical with the set of Kuhn–Tucker points at which the objective function $F(x)$ takes the same value α_i*

$$\mathcal{KT}(\mu_i) = \{x \in \mathcal{KT} : \quad F(x) = \alpha_i\};$$

it is reduced to a singleton if $-\mu_i \notin \{\lambda_j : j = 1, \ldots, n\}$ (the set of eigenvalues of Q).

(iii) $\alpha = \min\{F(x) : \|x\| \leq r\} = \min\{F(x) : x \in \mathcal{KT}\} = \min\{\alpha_i : i = 1, \ldots, p\} = \alpha_p$

and $\mathcal{KT}(\mu_p)$ is exactly the solution set to (QBA).

(iv) *The sequence $\{x^k\}$ generated by DCA has all its limit points belonging to one and only $\mathcal{KT}(\mu_i)$. If x^* is a limit point of $\{x^k\}$ then its associated Lagrangian multiplier λ^* is given by*

$$\lambda^* = -[\langle x^*, Qx^*\rangle + \langle q, x^*\rangle]/r^2$$

and we have $\lambda^* = \mu_i$.

(v) *The whole sequence $\{\lambda^k = -[\langle x^k, Qx^k\rangle + \langle q, x^k\rangle]/r^2\}$ converges to λ^*.*

For solving (QBA) we use the following d.c. decomposition which has been shown to be an efficient one from the computational viewpoint:

$$g(x) = \frac{1}{2}\rho\|x\|^2 + q^T x + \chi_C(x); \quad h(x) = \frac{1}{2}x^T(\rho I - Q)x, \tag{15}$$

where χ_C stands for the indicator function of the ball $C := \{x \in \mathbb{R}^n : \|x\| \leq r\}$ and ρ is a positive number such that the matrix $(\rho I - Q)$ is positive semidefinite. With this decomposition Problem (QBA) takes the form

(QBA_{dc}) $\quad \min\{g(x) - h(x) : x \in \mathbb{R}^n\}$.

Since $\nabla h(x) = (\rho I - Q)x$ for any x in $\mathbb{R}^n$, the DCA applied to (QBA$_{dc}$) is reduced to construction of two sequences $\{x^k\}$ and $\{y^k\}(k \geq 0)$ satisfying

$$y^k = (\rho I - Q)x^k, \tag{16}$$

$$x^{k+1} \in \arg\min\left\{\frac{\rho}{2}\|x\|^2 + x^T(q - y^k) + \chi_C(x) : x \in \mathbb{R}^n\right\}, \tag{17}$$

where $x^0 \in C$ is given in advance.

Clearly, x^{k+1} is a solution of the following problem

$$\min_{x \in C}\left\|x - \frac{y^k - q}{\rho}\right\|^2.$$

Hence, x^{k+1} is just the projection of $(y^k - q)/\rho$ on C, i.e.,

$$x^{k+1} = P_C\left(x^k - \frac{1}{\rho}(Qx^k + q)\right). \tag{18}$$

ALGORITHM DCQBA (DCA for computing a Kuhn–Tucker point of Problem (QBA)). *Let $\epsilon > 0$ be small enough, ρ be a positive number such that the matrix $(\rho I - Q)$ is positive semi-definite and $x^0 \in C$ be given. Set $k \leftarrow 0$, $er \leftarrow 1$.*

While $er > \epsilon$ **do**

If $\|(\rho I - Q)x^k - q\| \leq \rho r$, *then take* $x^{k+1} = [(\rho I - Q)x^k - q]/\rho$

else *take* $x^{k+1} = r[(\rho I - Q)x^k - q]/\|(\rho I - Q)x^k - q\|$

endif

$$\lambda^{k+1} = \frac{-\langle x^{k+1}, Qx^{k+1}\rangle - \langle x^{k+1}, q\rangle}{r^2}$$

$$er = \max\{\|(Q + \lambda^{k+1}I)x^{k+1} - q\|, \lambda^{k+1}(r - \|x^{k+1}\|)\}$$

$$k \leftarrow k + 1$$

endwhile

REMARK 1.

- In the case $q = 0$, exceptionally if Q is positive semi-definite (for which zero is a solution), we choose x^0 such that $y^0 = (\rho I - Q)x^0 \neq 0$, since $y^0 = 0$ implies $x^k = 0, \forall k \geq 1$
- In practice, the convergence rate very much depends on the value ρ. Numerical experiments indicated that in general the closer to λ_n the positive parameter ρ is, the better the DCA converges. Such "good" parameters ρ belong to $\{\lambda > 0 : \lambda \in [\rho^*, \rho^* + \varepsilon]\}$ where $\varepsilon > 0$ is sufficiently small and ρ^* defined by $\rho^* = 0$ if $\lambda_n < 0$, λ_n otherwise. Such a ρ can be computed by using the quite inexpensive *Implicit Restarted Lanczos Method* [49].

THEOREM 4.

(i) $F(x^{k+1}) \leq F(x^k) - \frac{1}{2}(\rho + \bar{\lambda})\|x^{k+1} - x^k\|^2$, *where* $\bar{\lambda}$ *is the smallest eigenvalue of* $(\rho I - Q)$.

(ii) $F(x^k) \searrow \alpha' \geq \alpha$ *and* $\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0$.

(iii) Either there is $p \geq 0$ *such that* x^p *is a Kuhn– Tucker point for (QBA), or every limit point* x^* *of the sequence* $\{x^k\}$ *is a Kuhn–Tucker point for (QBA).*

Proof. Theorem 4 is an immediate consequence of the DCA's convergence theorem for the general d.c. programming [see 1, 33, 37]. However, we wish to present a direct proof of Theorem 4 that is better suited to this setting.

We have for $x \in C$

$$F(x) = F(x^k) + \langle Qx^k + q, x - x^k\rangle + \frac{1}{2}\langle x - x^k, Q(x - x^k)\rangle.$$

So

$$F(x^{k+1}) = F(x^k) + \langle Qx^k + q, x^{k+1} - x^k\rangle + \frac{1}{2}\langle x^{k+1} - x^k, Q(x^{k+1} - x^k)\rangle. \tag{19}$$

By definition (18) of x^{k+1} we get

$$\left\langle x^k - \frac{1}{\rho}(Qx^k + q) - x^{k+1}, x^k - x^{k+1}\right\rangle \leq 0, \quad \textit{i.e.,}$$

$$\langle Qx^k + q, x^{k+1} - x^k\rangle \leq -\rho\|x^{k+1} - x^k\|^2. \tag{20}$$

Combining (19) and (20) gives

$$F(x^{k+1}) \leq F(x^k) - \frac{\rho}{2}\|x^{k+1} - x^k\|^2 + \frac{1}{2}\langle x^{k+1} - x^k, Q(x^{k+1} - x^k)\rangle.$$

Finally we obtain

$$F(x^{k+1}) \leq F(x^k) - \rho\|x^{k+1} - x^k\| - \frac{1}{2}\langle x^{k+1} - x^k, (\rho I - Q)(x^{k+1} - x^k)\rangle. \tag{21}$$

Now let $\bar{\lambda}$ be the smallest eigenvalue of the positive semi- definite matrix $\rho I - Q$. Then we have

$$F(x^{k+1}) \leq F(x^k) - \frac{\rho + \bar{\lambda}}{2}\|x^{k+1} - x^k\|^2.$$

The sequence $\{F(x^k)\}$ being decreasing and bounded below then converges to $\alpha' \geq \alpha$. Hence $\{\|x^{k+1} - x^k\|\}$ converges to zero since $\rho > 0$ and $\bar{\lambda} \geq 0$. It remains to prove property (iii).

Let x^* be a limit point of $\{x^k\}$ (it exists since the sequence $\{x^k\}$ is contained in C). Since $\lim_{k\to+\infty} \|x^{k+1} - x^k\| = 0$ we get

$$x^* = P_C\left(x^* - \frac{1}{\rho}(Qx^* + q)\right). \tag{22}$$

It is clear that (22) is the geometrical translation of the Kuhn–Tucker conditions: there is $\lambda^* \geq 0$ such that $(Q + \lambda^* I)x^* = -q$, $\|x^*\| \leq r$ and $\lambda^*(\|x^*\| - r) = 0$. □

REMARK 2. It is clear that Theorem 2 strengthens the possibility for DCQBA to reach a global minimizer to (QBA). In our numerical experiments DCQBA converges always to a global solution.

Using DCQBA for globally solving (QBA)

From property (iii) of Theorem 1 we see that checking the globality of a solution x^* provided by DCQBA amounts to check the positive semi- definiteness of the matrix $Q + \lambda^* I$. For this purpose, we use also IRLM for computing λ_1, its corresponding eigenvector u and then compare $-\lambda_1$ with

$$\lambda^* = \frac{-\langle x^*, Qx^*\rangle - \langle x^*, q\rangle}{r^2} \tag{23}$$

(following Theorem 1). In case if $\lambda^* < -\lambda_1$ (it means that $Q + \lambda^* I$ is nonpositive semi-definite, i.e., $u^T(Q + \lambda^* I)u = (\lambda^* + \lambda_1)u^T u < 0$) we can restart DCQBA from a new point $\bar{x}$ for which $F(\bar{x}) < F(x^*)$. Such a point can be computed as follows.

a) If $\langle q, x^*\rangle > 0$, then taking $\bar{x} = -x^*$ we have $F(\bar{x}) < F(x^*)$. Indeed, in virtue of Lemma 1,

$$F(x) = F(x^*) - \frac{\lambda^*}{2}(\|x\|^2 - \|x^*\|^2) + \frac{1}{2}\langle x - x^*, (Q + \lambda^* I)(x - x^*)\rangle. \tag{24}$$

This implies

$$F(\bar{x}) = F(x^*) + 2\langle (Q + \lambda^* I)x^*, x^*\rangle = F(x^*) - 2\langle q, x^*\rangle < F(x^*).$$

So we can restart DCQBA from the initial point $\bar{x}$

b) If $\langle q, x^*\rangle \leq 0$, then we distinguish two cases:

(b.1) If either $\|x^*\| < r$ or $\|x^*\| = r$ and $u^T x^* \neq 0$, then we take $\bar{x} = x^* + \gamma u$, where $\gamma \neq 0$ is chosen such that $\|\bar{x}\| = r$. More precisely, γ is a nontrivial solution of the equation

$$\|u\|^2\gamma^2 + 2u^T x^*\gamma + \|x^*\|^2 - r^2 = 0$$

which is given explicitly by

$$\gamma = \begin{cases} -2(u^T x^*)/\|u\|^2 & \text{if } \|x^*\| = r \\ (-(u^T x^*) \pm \sqrt{\Delta})/\|u\|^2 & \text{if } \|x^*\| < r, \end{cases} \tag{25}$$

where $\Delta = (u^T x^*)^2 - \|u\|^2(\|x^*\|^2 - r^2)$.

So we can choose $\bar{x}$ as the initial point for restarting DCQBA, since

$$F(\bar{x}) = F(x^*) + \frac{\gamma^2}{2}\langle u, (Q + \lambda^* I)u\rangle < F(x^*).$$

Note that in (25) the greater γ^2 is, the smaller $F(\bar{x})$ becomes. Therefore, a better value of γ is given by $\gamma = (-(u^T x^*) + \sqrt{\Delta})/\|u\|^2$ if $u^T x^* < 0$, $(-(u^T x^*) - \sqrt{\Delta})/\|u\|^2$ otherwise.

(b.2) If $\|x^*\| = r$, and $u^T x^* = 0$ (the latter implies $q^T u = 0$ since

$$\langle q, u\rangle = -\langle u, (Q + \lambda^* I\rangle = -\langle x^*, (Q + \lambda^* I)u\rangle = -(\lambda^* + \lambda_1)u^T x^*),$$

then we find a new vector $v = u + \tau x^*$ with $\tau \neq 0$ such that $v^T(Q + \lambda^* I)v < 0$ and $v^T x^* \neq 0$. Return to (b.1) using v instead of u.

To prove the existence of such a vector v we will proceed as follows. We have $v^T x^* = u^T x^* + \tau\|x^*\|^2 = \tau r^2$. So $v^T x^* \neq 0$ if and only if $\tau \neq 0$. Since

$$\begin{aligned} \langle v, (Q + \lambda^* I)v\rangle &= \tau^2\langle x^*, (Q + \lambda^* I)x^*\rangle + 2\tau\langle u, (Q + \lambda^* I)x^*\rangle \\ &\quad + \langle u, (Q + \lambda^* I)u\rangle \\ &= -\langle q, x^*\rangle\tau^2 - 2\langle q, u\rangle\tau + \langle u, (Q + \lambda^* I)u\rangle \\ &= -\langle q, x^*\rangle\tau^2 + \langle u, (Q + \lambda^* I)u\rangle, \end{aligned}$$

we deduce after several suitable computations the following properties:

- If $q^T x^* = 0$, then $v^T(Q + \lambda^* I)v < 0$ so all $\tau \neq 0$ is possible and optimal choices of τ (which correspond to the best decrease of the objective function $F(x)$ while passing from x^* to $\bar{x}$) are given by $\tau \in \{\tau' = -\|u\|/r, \tau'' = \|u\|/r\}$. Moreover we have

$$\bar{x}(\tau') = -\frac{r}{\|u\|}u, \quad \bar{x}(\tau'') = \frac{r}{\|u\|}u$$

and

$$F(\bar{x}(\tau') = F(\bar{x}(\tau'') = F(x^*) + \frac{r^2}{2\|u\|^2}\langle u, (Q + \lambda^* I)u\rangle.$$

- If $q^T x^* < 0$, then $v^T(Q + \lambda^* I)v < 0$ if and only if $\tau \in \{]\tau_1, 0[\cup]0, \tau_2[\}$, where τ_1 and τ_2 given by

$$\tau_1 = -\frac{[-u^T(Q + \lambda^* I)u]^{1/2}}{(-q^T x^*)^{1/2}}, \quad \tau_2 = -\tau_1.$$

Optimal choices of τ, in this case, are

$$\tau \in \left\{ \tau' = \frac{-\|u\|[-u^T(Q+\lambda^* I)u]^{1/2}}{[-u^T(Q+\lambda^* I)ur^2 - 2q^T x^* \|u\|^2]^{1/2}}, \quad \tau'' = -\tau' \right\}.$$

Remark that $\tau_1 < \tau' < 0 < \tau'' < \tau_2$ and actually the case $q^T x^* = 0$ can be incorporated into the case $q^T x^* < 0$ by taking $\tau_1 = -\infty$ and $\tau_2 = +\infty$ when $q^T x^* = 0$.

Summing up we conclude that if a Kuhn–Tucker point x^* produced by DCQBA is not a global solution to (QBA) then another point $\bar{x}$ (such that $\|\bar{x}\| \leq r$ and $F(\bar{x}) < F(x^*)$) is already available for restarting DCQBA.

Finite convergence of DCA with restarting to an approximate global solution of (QBA). According to Corollary 1, the sequence $\{x^k\}$ generated by DCA converges infinitely to a Kuhn–Tucker point for (QBA) and the set of limit points of $\{x^k\}$ is contained exactly in one $\mathcal{KT}(\mu_i)$. Let x^* be a limit point of $\{x^k\}$ then the whole sequence $\{\lambda^k = -[\langle x^k, Qx^k\rangle + \langle q, x^k\rangle]/r^2\}$ converges to $\lambda^* = -[\langle x^*, Qx^*\rangle + \langle q, x^*\rangle]/r^2$ and $\lambda^* = \mu_i$.

(i) If $x^* \in \mathcal{KT}(\mu_i)$ with $i = p$ then $\lambda^* = \mu_p$ and x^* is a global solution to (QBA). In this case, DCA provides after finitely many iterations an approximate global solution x^k to (QBA).

(ii) If x^* is a non global solution to (QBA), (i.e., $x^* \in \mathcal{KT}(\mu_i)$ with $i \neq p$) then the aforementioned procedure provides a feasible point $\bar{x}(x^*)$ such that $F(\bar{x}(x^*)) < F(x^*) = F(x) < F(x^p)$ for all p and all $x \in \mathcal{KT}(\mu_i)$.

The crucial problem then is to leave the set $\mathcal{KT}(\mu_i)$. According to the restarting procedure, DCA provides after finitely many iterations an approximation x^k of x^* such that $\bar{x}(x^k)$ is near to $\bar{x}(x^*)$ and so $F(\bar{x}(x^k)) < F(x^*)$. Restart now DCA from $\xi^0 = \bar{x}(x^k)$, it follows from the strict decrease of $\{F(\xi^k)\}$ (Theorem 4) that $\mathcal{KT}(\mu_i)$ contains no ξ^k neither does the sequence $\{x^k\}$.

$$F(\xi^l) < \cdots < F(\xi^0) < F(x^*) = F(x) < F(x^p)$$

for all l and p and for all $x \in \mathcal{KT}(\mu_i)$.

In practice such an approximate Kuhn–Tucker point x^k can be computed by DCA as the first x^p satisfying

$$\|(Q+\lambda^p I)x^p - q\| \leq \varepsilon, \lambda^p \geq 0, \lambda^p(r - \|x^p\|) \leq \varepsilon$$

with $\varepsilon > 0$ is small enough.

We now describe our global algorithm based on DCQBA for solving (QBA):

ALGORITHM 1 (globally solving (QBA)):
Initialization. *Compute λ_1 and u by IRLM.*
Compute an approximation $\tilde{\lambda}_n$ to λ_n by IRLM.

Let $\rho := \tilde{\lambda}_n + 0.1$, $\bar{x} \in C$, *stop* := *false.*

While *stop* = *false* **do**

1. Apply DCQBA from the starting point $x^0 := \bar{x}$ *to obtain* x^*.

2. Let $\lambda^* = (-\langle x^*, Qx^*\rangle - \langle x^*, q\rangle)/r^2$.

If $\lambda^* \geq -\lambda_1$, **then** *stop* := *true,* x^* *is a global solution of (QBA)*

else *compute* $\bar{x}$ *such that* $F(\bar{x} < F(x^*)$ *and return to 1.*

end while

REMARK 3. According to Theorem 3, for nonconvex program (QBA) Algorithm 1 requires only matrix-vector products and converges to a global solution after at most $2m + 2$ restarting procedures. In general we rarely have recourse to this procedure.

2.2. UPPER BOUND – DCA TO BOX CONSTRAINED QUADRATIC PROBLEM

For finding a good feasible point we use again the DCA to Problem (QB), since the DCA applied to this problem (with a suitable d.c. decomposition) is very simple and usually leads to a point which is a good approximation to a global solution.

Using a similar decomposition to (15) for F we obtain Algorithm 2a which will be indicated below. This seems to be the most efficient decomposition since the projection of a point over a rectangle is easily computed. More precisely, taking

$$g_1(x) = \frac{1}{2}\rho\|x\|^2 + b^T x + \chi_{K^0}(x); \quad h_1(x) = \frac{1}{2}x^T(\rho I - A)x \tag{26}$$

(ρ is a positive number such that the matrix $(\rho I - A)$ is positive semi-definite) we can write (QB) in the form of d.c. program

$$\text{(QB)} \Leftrightarrow \min\{g_1(x) - h_1(x) : x \in \mathbb{R}^b\}. \quad (\text{QB}_{dc})$$

Then, as in Algorithm DCQBA, the DCA applied to (QB_{dc}) consists in the construction of two sequences $\{x^k\}$ and $\{y^k\}$ as follows

$$y^k = (\rho I - A)x^k; \quad x^{k+1} = P_{K^0}((y^k - b)\rho). \tag{27}$$

ALGORITHM 2a (DCA for computing a Kuhn–Tucker point of Problem (QB)).
Let $\epsilon > 0$ *be small enough,* ρ *be a positive number such that the matrix* $(\rho I - A)$ *is positive semidefinite and* $x^0 \in K^0$ *be given. Set* $k \leftarrow 0$, $er := 1$.
While $er > \epsilon$ **do**

$y^k \leftarrow (\rho I - A)x^k$

For $i = 1, \ldots, n$ **do**

If $l_i \le (y_i^k - b_i)/\rho \le u_i$, *then take* $x_i^{k+1} := (y_i^k - b_i)/\rho$
else
if $(y_i^k - b_i)/\rho < l_i$, *then take* $x_i^{k+1} := l_i$
else *take* $x_i^{k+1} := u_i$
endif
endif
endfor
$er \leftarrow \|x^{k+1} - x^k\|$
$k \leftarrow k + 1$
endwhile

Consider now a special case of (QB) where A is negative semi-definite matrix, i.e, one is faced with a concave minimization problem. We propose another d.c. decomposition defined by

$$g_1(x) = \chi_{K^0}(x); \quad h_1(x) = \frac{1}{2}\langle x, -Ax\rangle - \langle b, x\rangle. \tag{28}$$

The DCA applied to (QB_{dc}) is written as follows:

$$y^k = -Ax^k - b; \quad x^{k+1} \in \operatorname{argumin}\{\langle x, -y^k\rangle : x \in K^0\}. \tag{29}$$

Since $\min\{\langle x, -y^k\rangle : x \in K^0\} = \sum_{i=1}^n \min\{-x_i y_i^k : l_i \le x_i \le u_i\}$ we get

ALGORITHM 2b (for convex maximization program). *Let* $\epsilon > 0$ *be small enough and* $x^0 \in K^0$ *be given. Set* $k \leftarrow 0$, $er \leftarrow 1$.
While $er > \epsilon$ **do**
$y^k \leftarrow -Ax^k - b$
For $i = 1, \ldots, n$ **do**
If $y_i^k \le 0$, *then take* $x_i^{k+1} := l_i$
else *take* $x_i^{k+1} := u_i$
endif
endfor
$er \leftarrow \|x^{k+1} - x^k\|$
$k \leftarrow k + 1$
endwhile

REMARK 4.

- Algorithms 2a and 2b require only matrix-vector products, thus they may be applied to large-scale problems.

- A d.c. program (P_{dc}) is called *d.c. polyhedral* if either g or h is polyhedral. This class of d.c. optimization problems, which is frequently encountered in practice and has been extensively developed in our previous works (see e.g. [1] and references therein), enjoys interesting properties (from both theoretical and practical viewpoints) concerning the local optimality and the convergence of the DCA. Clearly, with the decomposition (28) (QB_{dc}) is a polyhedral d.c. program since $g_1 = \chi_{K^0}$ is a polyhedral function. In this case the DCA converges after a finite number of iterations (Theorem 9 of [1]). Moreover, from Proposition 1 in [1] it follows that if Algorithm 2b provides a solution x^* such that $-Ax^* - b \in \text{int}\, \partial K^0(x^*)$ (this condition is in many cases satisfied), then x^* is a local solution of (QB).
- In our experimentations both Algorithm 2a and 2b converge very rapidly (when $n \leq 500$ the maximal number of iterations is 3) to a local minimizer of (QB). Moreover they provide a global solution if the initial point x^0 is near enough to a global one.

How to choose a good initial point for Algorithms 2a and 2b? It is well-known that Problem (QB) may have many local minimizers. Thus, to obtain a good approximation to its global solution we start Algorithm 2a (or Algorithm 2b) at each iteration k from a suitable initial point x_0^k. The point x_0^k is chosen as follows: at iteration k one considers the ellipsoid E^k corresponding to the smallest lower bound. E^k is a candidate to ellipsoids containing a global solution. So starting Algorithm 2a (or Algorithm 2b) from a point in $E^k \cap K^0$ can detect a global one. A good point can be taken by the projection of $\bar{x}^k$ on K^0 which is a solution obtained by applying Algorithm 1 to the problem

$$(QE^k) \quad \beta(K^k) := \min \left\{ \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle : x \in E^k \right\}.$$

More precisely we have the following procedure:

Procedure IP Let $\bar{x}^k$ be an optimal solution of (QE^k).

For $i = 1, \ldots, n$ **do**

If $l_i \leq \bar{x}_i^k \leq u_i$ **then** take $x_{0i}^k := \bar{x}_i^k$

else

If $\bar{x}_i^k < l_i$ **then** take $x_{0i}^k := l_i$

else take $x_{0i}^k := u_i$

endif

endif

endfor

Numerical tests indicated that (Section 5) by this way one can quickly reach a good feasible point and thereby a lot of numbers of generated ellipsoids can be eliminated from further consideration.

2.3. BRANCHING PROCEDURE

For rectangular subdivision, the bisection process via a longest edge as thoroughly discussed in Horst & Tuy [19], is often used. For example, in [20] and [38] the following rule was proposed:

Rule 1: Let $K^k = \{x : a_i^k \leq x_i \leq t_i^k\}$ be a rectangle corresponding to smallest lower bound which will be divided. Let j_k be the index of the longest edge of K^k, i.e., such that

$$t_{j_k}^k - a_{j_k}^k = \max\{t_i^k - a_i^k, \quad i = 1, \ldots, n\}.$$

Let $\bar{\alpha} = 1/2(t_{j_k} + a_{j_k})$. Then K^k is bisected into two subrectangles:

$$K^{k_1} = \{x \in K^k : x_{j_k} \leq \bar{\alpha}\}, \quad K^{k_2} = \{x \in K^k : x_{j_k} \geq \bar{\alpha}\}. \tag{30}$$

This process seems to be not well adapted to the ellipsoidal technique used in our branch and bound algorithm. We propose an alternative bisection which uses the solution of a current ellipsoidal constrained problem (QE^k). This is based on the fact that if $\bar{x}^k \in K^k$ then one has an exact evaluation over K^k and therefore the rectangle K^k will not be divided further. The following rule forces $\bar{x}^k$ to enter in K^k.

Rule 2: For each selected K^k we consider the solution $\bar{x}^k$ of Problem (QE^k). Let j_k be the index satisfying

$$\max(\bar{x}_{j_k}^k - t_{j_k}^k, a_{j_k}^k - \bar{x}_{j_k}^k) = \max_{i=1,\ldots,n} \max(\bar{x}_i^k - t_i^k, a_i^k - \bar{x}_i^k). \tag{31}$$

and $\bar{\alpha} = 1/2(t_{j_k} + a_{j_k})$. Then K^k is bisected into two subrectangles as in (30).

This procedure works very well in our numerical experiments. Nevertheless using the Rule 2 one may meet an undesired fact: the length of j_k^{th}-edge is near to zero, i.e., the volume of the ellipsoid E^k containing K^k shrinks to zero but E^k may not shrink to a singleton. In this case the convergence of the branch and bound algorithm is not ensured. In order to prove the convergence of our algorithm we combine Rule 2 with Rule 1. More precisely, we consider first the index j_k satisfying (31). If

$$t_{j_k}^k - a_{j_k}^k > \delta (\delta \text{ being a given small positive number}), \tag{32}$$

then Rule 2 is used, otherwise we apply Rule 1.

By this way, our algorithm generates two whole sequences $\{E^k\}$ and $\{K^k\}$ which shrink to a singleton as k shrinks to $+\infty$. We observe that in our numerical experiments Rule 2 is always used, i.e., (32) is always satisfied.

2.4. THE DESCRIPTION OF THE BRANCH AND BOUND ALGORITHM AND ITS CONVERGENCE

Using notions and procedures presented above we can now establish a branch and bound algorithm for Problem (QB) as follows.

ALGORITHM BDCE
Initialization
Solve the following problem

$$(QE^0) \quad \beta(K^0) := \min\left\{\frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle : x \in E^0\right\}$$

to obtain an optimal solution $\bar{x}^0$ and lower bound $\beta^0 := \beta(K^0)$.
If $\bar{x}^0 \in K^0$ then *stop* $\leftarrow$ *true*, $\bar{x}^0$ is a global solution of (QB)
else

Solve the following problem by Algorithm 2a (or Algorithm 2b)

$$(QB) \quad \min\left\{f(x) := \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle : l_i \le x_i \le u_i, \; i = 1, \ldots, n\right\}$$

from the starting point supplied by Procedure IP to obtain the best point x^0 and upper bound γ^0, i.e., $f(x^0) = \gamma^0$.

If $\gamma^0 = \beta(K^0)$, then *stop* $\leftarrow$ *true*, x^0 is a global solution of (QB)

else *stop* $\leftarrow$ *false*

endif

endif

Set $\mathcal{R} \leftarrow \{K^0\}, \quad k \leftarrow 0$.

While *stop* = *false* **do**

Select a rectangle $K^k \in \mathcal{R}$ such that $\beta^k := \beta(K^k) = \min\{\beta(K) : K \in \mathcal{R}\}$.

Update upper bound γ^k by applying Algorithm 2a (or Algorithm 2b) to (QB) from a starting point chosen as in Procedure IP.

Let x^k be the known feasible point such that $\gamma^k = f(x^k)$.

Bisect K^k into K^{k_1} and K^{k_2}.

Compute $\beta(K^{k_i})$ by solving Problems (QE^{k_i}), $(i = 1, 2)$

$$(QE^{k_i}) \quad \beta(K^{k_i}) := \min\left\{\frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle : x \in E^{k_i}\right\}.$$

Let $\bar{x}^{k_i}$ be an optimal solution of (QE^{k_i}).

If $\bar{x}^{k_i} \in K^{k_i}$ and $f(\bar{x}_{k_i}) < \gamma^k$ **then**

$\gamma^k := f(\bar{x}_{k_i})$; $x^k := \bar{x}_{k_i}$

endif

If $\beta(K^{k_i}) < \beta(K^k)$ then set $\beta(K^{k_i}) \leftarrow \beta(K^k)$.

Set $\mathcal{R} \leftarrow \mathcal{R} \cup \{K_i^k : \beta(K_i^k) < \gamma^k, i = 1, 2\} \backslash \{K^k\}$.

If $\mathcal{R} = \emptyset$ then *stop* $\leftarrow$ *true*, x^k is an optimal solution.

else Set $k \leftarrow k + 1$.

endif

endwhile

THEOREM 5 (convergence of algorithm BDCD).

(i) If the algorithm terminates at iteration k, then x^k is a global optimal solution to problem (QB).

(ii) If the algorithm is infinite, then it generates a bounded sequence $\{x^k\}$, every accumulation point of which is a global optimal solution of (QB), and

$$\gamma^k \searrow f_*, \quad \beta^k \nearrow f_*$$

(f_* denotes the optimal value of f on K^0).

Proof. If the algorithm terminates at iteration k then $\beta^k := \beta(K^k) = \gamma^k := f(x^k)$. Thus the part (i) follows from the definition of β^k, γ^k, x^k.

Assume now the algorithm is infinite, then it must generate an infinite sequence $\{E^k\}$ of ellipsoids whose volumes decrease to zero. By the branching procedure, the whole sequence $\{E^k\}$ and $\{K^k\}$ shrink to a singleton. On the other hand, from the construction one sees that the sequence γ^k is nonincreasing, while the sequence β^k is nondecreasing and therefore $\{\gamma^k - \beta^k\}$ is a nonincreasing sequence of positive numbers. Hence, the whole sequence $\gamma^k - \beta^k$ must tend to zero. This and $\beta^k \leq f_* \leq \gamma^k$ for every k imply

$$\gamma^k \searrow f_*, \quad \beta^k \nearrow f_*.$$

Since $x^k \in K^0$ and $\gamma^k = f(x^k)$, any cluster point of the sequence $\{x^k\}$ belongs to K^0 and has the function value f_*, i.e., solves problem (QB). The theorem is proved. □

REMARK 5.

- In practice the following optimality criterion is used: $\beta^k \geq \gamma^k - \varepsilon|\gamma^k|$ with $0 < \varepsilon < 1$. So the step "Set $\mathcal{R} \leftarrow \mathcal{R} \cup \{K_i^k : \beta(K_i^k) < \gamma^k, i = 1, 2\} \backslash \{K^k\}$" in Algorithm BDCE is actually replaced by "Set $\mathcal{R} \leftarrow \mathcal{R} \cup \{K_i^k : \beta(K_i^k) < \gamma^k - \varepsilon|\gamma^k|, i = 1, 2\} \backslash \{K^k\}$". Hence, when the algorithm stops, we obtain an ε-optimal solution of (QB).

- In the step *Initialization*, the case where $\bar{x}^0 \notin K^0$ and $\gamma^0 = \beta^0$ may occur if the optimal solution of (QE^0) is not unique. This corresponds to the *hard case* of Problem (QE^0) where q is perpendicular to $\mathrm{Ker}(Q - \lambda_1 I)$ (see [29], [34]).

3. Implementation and illustrative example

We give below some details of an implementation of Algorithm BDCE.

The most expensive operation in Algorithm BDCE is solving subproblem (QE) at each iteration. Since (QE) is equivalent to (8), we define the objective function of Problem (8) and then use Algorithm 1 to solve the last. So at each iteration we have to solve two problems of the form (8) by Algorithm 1. Observe that the objective functions of these two problems differ from each other only on the linear term. Indeed, in passing from the rectangular K^k which to be bisected at iteration k to subrectangles $K^{k_i} := \prod_{j=1}^{n}[a_j^{k_i}, t_j^{k_i}](i = 1, 2)$ only the coordinates $t_{j_k}^{k_i}$ and $a_{j_k}^{k_i}$ are modified. Then using (5) and (6), after a simple calculation, we get

$$d_{j_k}^{k_1} = d_{j_k}^{k_2} = 4/(t_{j_k}^k - a_{j_k}^k) = -2d_{j_k}^k/c_{j_k}^k, \tag{33}$$

$$c_{j_k}^{k_1} = (3a_{j_k}^k + t_{j_k}^k)/(a_{j_k}^k + t_{j_k}^k) = (1 + 2c_{j_k}^k)/c_{j_k}^k, \tag{34}$$

$$c_{j_k}^{k_2} = (a_{j_k}^k + 3t_{j_k}^k)/(a_{j_k}^k - t_{j_k}^k) = -1 + 2c_{j_k}^k. \tag{35}$$

(As before d^{k_i} and $c^{k_i}(i = 1, 2)$ denote the vectors containing diagonal elements of the matrix D^{k_i} which define the ellipsoids E^{k_i} and the centres of E^{k_i} respectively). It means that the ellipsoids E^{k_1} and E^{k_2} are defined by the same matrix while their centres are different. This allows us to reduce the number of operations as well as the memory storage during the performing of the algorithm.

The most important in the computational implementation of a branch and bound algorithm is the construction of a convenient computer scheme for storing and updating the informaiton about partition elements. For each E corresponding to $K \in \mathcal{R}$ we need save the following information: vectors d and c, the optimal solution $\bar{x}$ of (QE) (for finding a starting point of Algorithm 2a or 2b and choosing the index of branching procedure) and the lower bound $\beta(K)$. Since a vector d may define two different ellipsoids, for storing these informations we use two linked lists: POINT and ELLIP. The first list contains the coordinates of vector defining an ellipsoid $E \in \mathcal{R}$, the second one contains the information about the ellipsoid E: centre c, $\bar{x}$, $\beta(K)$ and the index of the vector d used to link these lists. In passing from iteration k to iteration $k + 1$ at most one new element may be added to the list POINT while an element selected in the list ELLIP will be replaced by at most two new elements. An element in the list POINT will be deleted if all of its linked elements in the list ELLIP have been deleted.

However we do not need store any information about the rectangle K corresponding to E. As soon as an ellipsoid E^k is selected for branching, using (5) and (6) we can recover the edges of the rectangle K^k to be divided.

Let us provide now an illustrative example.

EXAMPLE: Consider the following problem

$$\min\{F(x) := \langle x, Ax\rangle : x \in K^0 := [-1, 1]^4\}, \tag{36}$$

$$\begin{pmatrix} -22 & -1 & 7 & -6 \\ -1 & -33 & -4 & 7 \\ 7 & -4 & -27 & 1 \\ -6 & 7 & 1 & -38 \end{pmatrix}$$

We find an ε-optimal solution of (36) with $\varepsilon = 0.02$.
Step Initialization: Solving problem (QE^0) where $E^0 = \{x \in \mathbb{R}^4 : \|x\| \le 2\}$, we obtain

$$\beta(K^0) = -180 \text{ and } \bar{x}^0 = (0.5165, -1.0325, -0.5163, 1.5494)^T.$$

Applying Algorithm 2b to (36) from the point $x_0^0 = (0.5165, -1, -0.5163, 1)^T$, we get the current best feasible point $x^0 = (1, -1, -1, 1)^T$ and upper bound $\gamma^0 = F(x^0) = -168$.
Then *Stop* := *false*.
Iteration 1 $(k = 0)$: K^0 is divided into two rectangles by Rule 2

$$K^{0_1} = \{x \in K^0 : x_4 \le 0\}; \quad K^{0_2} = \{x \in K^0 : x_4 \ge 0\}.$$

Solving (QE^{0_1}) and (QE^{0_2}) we obtain

$$\beta(K^{0_1}) = -177, 4677; \quad \bar{x}^{0_1} = (-0.5929, 1.5471, 0.9532, -0.7942)^T;$$
$$\beta(K^{0_2}) = -137, 3199; \quad \bar{x}^{0_2} = (-0.3221, 1.5467, 1.2219, 0.5524)^T.$$

Set $\mathcal{R} := \{E^{0_1}\}$.
Iteration 2 $(k = 1)$*:* K^{0_1} is divided into two rectangles

$$K^{1_1} = \{x \in K^{0_1} : x_2 \le 0\}; \quad K^{1_2} = \{x \in K^{0_1} : x_2 \ge 0\}.$$

Applying Algorithm 2b from the point $(-0.5929, 1, 0.9532, -0.7942)^T$, we get $x^1 = (-1, 1, 1, -1)^T$, $F(x^1) = -168$ which is also the current best feasible point. $\gamma^1 = -168$. Solving (QE^{1_1}) and (QE^{1_2}), we obtain

$$\beta(K^{1_1}) = -153.3060; \quad \bar{x}^{1_1} = (-1.3564, -0.5717, 1.3653 - 0.7624)^T;$$
$$\beta(K^{1_2}) = -179.6700; \quad \bar{x}^{1_2} = (-1.1067, 0.7522, 1.4550 - 0.8178)^T.$$

Set $\beta(K^{1_2}) := \beta(K^{0_1}) = -177.4677$ and $\mathcal{R} := \{E^{1_2}\}$.
Iteration 3 ($k = 2$)*:* K^{1_2} is divided into two rectangles

$$K^{2_1} = \{x \in K^{1_2} : x_3 \leq 0\}; \quad K^{2_2} = \{x \in K^{1_2} : x_3 \geq 0\}.$$

Applying Algorithm 2b from the point $(-1, 0.7522, 1, -0.8178)^T$ we get $x^2 = x^1$ then $\gamma^2 = -168$. Solving (QE^{2_1}) and (QE^{2_2}) we obtain

$$\beta(K^{2_1}) = -135.2690; \quad \bar{x}^{2_1} = (1.6955, 0.8368, -0.8227 - 0.7525)^T;$$
$$\beta(K^{2_2}) = -177.3723; \quad \bar{x}^{2_2} = (-1.5534, 0.8019, 0.8227 - 0.9489)^T.$$

Set $\mathcal{R} := \{E^{2_2}\}$.
Iteration 4 ($k = 3$)*:* K^{2_2} is divided into two rectangles

$$K^{3_1} = \{x \in K^{2_2} : x_1 \leq 0\}; \quad K^{3_2} = \{x \in K^{2_2} : x_1 \geq 0\}.$$

Applying algorithm 2b from the point $(-1, 0.8019, 0.8227, -0.9489)^T$ we get $x^3 = x^1$ and then $\gamma^3 = -168$. Solving (QE^{3_1}) and (QE^{3_2}) we obtain

$$\beta(K^{3_1}) = -171.0561; \quad \bar{x}^{2_1} = (-0.8612, 1.0110, 0.9145, -1.1608)^T;$$
$$\beta(K^{3_2}) = -138.0100; \quad \bar{x}^{3_2} = (0.4883, 1.1687, 0.8278, -1.1673)^T.$$

Hence $\mathcal{R} := \emptyset$ and therefore *stop* := *true*. Then both x^0 and x^1 are ε-optimal solutions of (36).

Note that if we choose $\varepsilon = 0.01$ the algorithm gives the same optimal solutions x^0 and x^1 after 9 iterations.

4. A variant of BDCE (BDCEU)

A branch and bound approach using a convex underestimation (BBU) has been successfully applied to indefinite quadratic programming in which the objective function is the difference of a nonseparable convex part and a separable convex part [1, 41]. For Problem (QB) we can introduce the following d.c. decomposition of the objective function

$$f(x) := g(x) - h(x) = \frac{1}{2}\langle (A + \rho I)x, x\rangle + \langle b, x\rangle - \frac{\rho}{2}\|x\|^2, \tag{37}$$

where ρ is a positive number such that the matrix $\rho I + A$ be positive semidefinite. The convex function $h(x) = (\rho/2)\sum_{i=1}^n x_i^2$ is clearly separable in its variables: $h(x) = \sum_{i=1}^n h_i(x_i)$ with $h_i(x_i) = (\rho/2)x_i^2$. The convex underestimation of $f(x)$ on the subdivided rectangle $K = \prod_{i=1}^n [a_i, t_i]$ is then ([1], [41]) $g(x) - \phi_{K^k}(x)$.

Here $\phi_K(x) = \sum_{i=1}^n \phi_{K_i}(x_i)$ is defined by: for $i = 1, \ldots, n$, $\phi_{K_i}(x_i)$ is the affine function that agrees with h_i at the endpoints of the segment $[a_i, t_i]$. In other words

$$\phi_{K_i}(x_i) = \frac{\rho}{2}(a_i + t_i)x_i - \frac{\rho}{2}a_i t_i.$$

BBU [1, 41] differs from BDCE in three points:

- The use of the convex underestimation $g(x) - \phi(x)$ for computing the lower bound of $f(x)$ over the subdivided rectangle K. We are faced with a box constrained convex quadratic minimization problem which can be globally solved by the DCA.
- The use of w-subdivision in the branching procedure: for the selected subrectangle K^k we have $\phi_{K^k}(x^k) - h(x^k) > 0$. Choose an index i_k satisfying

  $$i_k \in \operatorname{argmax}_i \{\phi_{K^{k_i}}(x_i^k) - h_i(x_i^k)\}$$

  and subdivide K^k into two subrectangles

  $$K^{k_1} = \{x \in K^k : x_{i_k} \le x_{i_k}^k\}; \quad K^{k_2} = \{x \in K^k : x_{i_k} \ge x_{i_k}^k\}.$$

- The starting point of the DCA in the solution of Problem (QB).

A comparison between the two algorithms through a lot of numerical examples will be presented in Section 5.

In Algorithm BDCE it may happen that $\beta(K^{k_i}) \le \beta(K^k)$ for $i = 1, 2$.

To speed up the convergence of BDCE, it seems to be interesting for such a case to perform an additional computation of lower bound by the bounding procedure of BBU: after solving Problems $(QE^{k_i})(i = 1, 2)$ if $\beta(K^{k_i}) \le \beta(K^k)$, we solve the problem

$$\tilde{\beta}(K^{k_i}) := \min \left\{ \frac{1}{2}\langle (A + \rho I)x, x\rangle + \langle b, x\rangle - \sum_{i=1}^n \left[\frac{\rho}{2}(a_i^k + t_i^k)x_i - \frac{\rho}{2}a_i^k t_i^k \right] : a_i^k \le x_i \le t_i^k \right\}$$

and then set $\beta(K^{k_i}) := \min\{\beta(K^{k_i})\}$. The modified algorithm will be denoted by BDCEU.

5. Computational results and conclusions

The algorithms were coded in Pascal under a Unix system. In this section we present some computational tests on the performance of our algorithm for different classes of test problems. All test problems are run in double precision on a SUN/SPARC-20 station.

In Algorithms 1 and 2a, 2b when $\|x^k\| > 1$ the stopping criterion $er := \|x^{k+1} - x^k\| \le \epsilon$ was replaced by $er : +\|x^{k+1} - x^k\|^2/\|x^k\|^2 \le \epsilon$. We took $\epsilon = 10^{-8}$. For

all test problems the initial point of Algorithm 1 was $x = e$ (it is a point in the boundary of E^0).

5.1. SPIN-GLASS {-1,1} INDEFINITE QUADRATIC MINIMIZATION PROBLEM

In statistical physics, to study magnetic order of the magnetic material, one is led to minimize the global energy of a complex physical system composed of magnetic moments in interaction, called spin glass. The discrete mathematical model of this problem is given by [5], [24]

$$\min\left\{\sum_{i=1}^{N} a_{ij}x_ix_j : x_i \in \{-1, 1\}\right\} \tag{38}$$

where $x = (x_1, \ldots, x_N)$ is a spin state vector, $A = (a_{ij})$ is an interaction matrix. It is an $N \times N$ integral symmetric matrix which is almost block tridiagonal of the form ($N = n^2$).

$\begin{matrix} & \pm1 & & \pm1 \\ \pm1 & & \pm1 & \\ & \pm1 & & \pm1 \\ \pm1 & & \pm1 & \end{matrix}$	$\begin{matrix} \pm1 & & & \\ & \pm1 & & \\ & & \pm1 & \\ & & & \pm1 \end{matrix}$		$\begin{matrix} \pm1 & & & \\ & \pm1 & & \\ & & \pm1 & \\ & & & \pm1 \end{matrix}$
$\begin{matrix} \pm1 & & & \\ & \pm1 & & \\ & & \pm1 & \\ & & & \pm1 \end{matrix}$	$\begin{matrix} & \pm1 & & \pm1 \\ \pm1 & & \pm1 & \\ & \pm1 & & \pm1 \\ \pm1 & & \pm1 & \end{matrix}$	$\begin{matrix} \pm1 & & & \\ & \pm1 & & \\ & & \pm1 & \\ & & & \pm1 \end{matrix}$	
	$\begin{matrix} \pm1 & & & \\ & \pm1 & & \\ & & \pm1 & \\ & & & \pm1 \end{matrix}$	$\begin{matrix} & \pm1 & & \pm1 \\ \pm1 & & \pm1 & \\ & \pm1 & & \pm1 \\ \pm1 & & \pm1 & \end{matrix}$	$\begin{matrix} \pm1 & & & \\ & \pm1 & & \\ & & \pm1 & \\ & & & \pm1 \end{matrix}$
$\begin{matrix} \pm1 & & & \\ & \pm1 & & \\ & & \pm1 & \\ & & & \pm1 \end{matrix}$		$\begin{matrix} \pm1 & & & \\ & \pm1 & & \\ & & \pm1 & \\ & & & \pm1 \end{matrix}$	$\begin{matrix} & \pm1 & & \pm1 \\ \pm1 & & \pm1 & \\ & \pm1 & & \pm1 \\ \pm1 & & \pm1 & \pm1 \end{matrix}$

Here is an example with $n = 4$, all nonzero a_{ij} are 1 or -1.

Furthermore, to study a change of phase of the magnetic material, physicists need to draw the curve of minimal energies in function of the density d^- of negative interactions ($a_{ij} = -1$) with respect to the total of interactions ($a_{ij} = 1$ or -1):

$$d^- = \frac{|\{(i, j) : a_{ij} = -1\}|}{|\{(i, j) : a_{ij} = 1 \text{ or } -1\}|} = \frac{|\{(i, j) : a_{ij} = -1\}|}{4n^2}$$

where $|\{(i, j) : a_{ij} = -1\}|$ denotes the cardinal of the set $\{(i, j) : a_{ij} = -1\}$.

For solving problem (38) we first write it in the form

$$\min\{\langle x, (A - 4I)x\rangle + 4 : x \in \{-1, 1\}^n\}.$$

Then solving the concave minimization program

$$\min\{\langle x, (A - 4I)x\rangle + 4 : x \in [-1, 1]^n\} \tag{39}$$

we obtain an optimal solution of (39) in $\{-1, 1\}^n$ following Algorithm 2b.

Tables 1 and 2 indicate the performance of three algorithms BDCE, BBU and BDCEU for two classes of spin-glass problems in which the sign of $a_{ij} \neq 0$ was chosen as follows:

Data for Table 1:

For $j = 0, \ldots, n-1$

S1 for $i = 1 + jn, \ldots, 1 + jn + n - 2$:

if $i > N/4$ then $a_{i,i+1} := 1$ else $a_{i,i+1} := -1$; $a_{i+1,i} := a_{i,i+1}$

S2 $a_{1+jn,1+jn+n-1} = a_{1+jn+n-1,1+jn} := 1$;

For $i := 1, \ldots, (n-1)n$:

S3 if $i > N/4$ then $a_{i,n+i} := 1$ else $a_{i,n+i} := -1$; $a_{n+i,i} = a_{i,n+i} := 1$.

For $i := 1, \ldots, n$:

S4 if odd(i) then $a_{i,(n-1)n+i} := 1$ else $a_{i,(n-1)n+i} := -1$; $a_{n-1)n+i,i} := a_{i,(n-1)n+i}$.

The data for Table 2 was generated by replacing in S1, S3, S4 $N/4$ with $N/3$ and 1 with -1, -1 with 1.

We find an ε-solution of these problems with $\varepsilon = 0.05$. The abbreviations in these Tables were the following: num – number of element $a_{ij} = -1$ in the data; iter – number of iterations; CPU – CPU time in seconds; NLB2 – number of how many times the bounding procedure of BBU has been used in BDCEU; NALB2 – number of how many times if the lower bounds obtained by this bounding procedure is better than lower bounds in BDCE (i.e. $\tilde{\beta}(K^{k_i}) < \beta(K^{k_i})$ the optimal value of (QE^{K_i})).

In the second class of problems, since BDCE converges after one iteration (except the case where $N = 36$), it is not interesting to present the performance of BDCEU in Table 2.

Computational results reported in these tables show that:

- BDCE is very efficient for Spin-glass problems, especially for the second class (except when $N = 36$).
- The use of second process of lower bounding (BDCEU) is interesting only for $N \leq 36$.
- BBU is efficient when N small ($N \leq 36$) but is not applicable to large-scale problems.

Table 1. The performance of BDCE, BDCEU and BBU for the first class of spin-glass problems

		BDCE		BDCEU				BBU	
N	num	iter	CPU	iter	CPU	NLB2	NALB2	iter	CPU
9	8	9	0.02	1	0.00	0	0	9	0.28
16	14	216	10.17	31	2.80	50	38	141	6.95
25	22	740	94.8	21	4.77	30	16	115	17.8
36	34	21750	6045	61	32.25	93	27	563	187.5
49	46	9	3.93	9	5.37	8	0	797	5883.78
64	60	17	15.48	17	22.08	26	10	1273	2020
81	76	10	12.22	10	19.33	11	1	3002	6097
100	96	30	89.92	24	109.83	40	14		
121	116	10	28.98	10	49.63	11	0		
144	138	26	251.57	26	331.18	44	12		
169	162	16	128.43	16	208.4	23	7		
196	190	19	228.35	19	373.37	30	3		
225	218	19	288.62	19	503.35	30	2		
256	248	24	531.55	24	897.37	40	4		
289	280	20	568.02	20	1016.05	32	0		
324	316	25	812.02	25	1510.88	42	4		
361	390	21	877.8	21	1746	20	8		

5.2. THE 'DIFFICULT CLASS OF TEST PROBLEMS' IN [30]

In this experiment we solved the following class of problems given by Pardalos et al. in [30]

$$\min \Bigg\{ f(x) := -n(n-1)\sum_{i=1}^{n} x_i - \sum_{i=1}^{n/2} x_i + n\sum_{i<j} x_i x_j + n\sum_{i>j} x_i x_j : x \in \{0,1\}^n \Bigg\}. \tag{40}$$

Pardalos et al. regarded that as a *difficult class of test problems* since (40) has an exponential number of discrete local minima. As indicated in [30], for n even, the global solution of (40) is the point

$$x^* = (1, \ldots, 1, 0, \ldots, 0)^T \tag{41}$$

with exactly $n/2$ ones followed by $n/2$ zeros and any point with $n/2$ ones is a discrete local minimizer. Fol solving (40) we write this in the form $(0, 1)$-quadratic

Table 2. The performance of BDCE and BBU for the second class of spin- glass problems

		BDCE		BBU	
N	num	iter	CPU	iter	CPU
9	18	1	0.00	3	0.08
16	34	1	0.02	9	0.4
25	56	1	0.07	13	1.67
36	82	26509	10355	86	17.27
49	116	1	0.4	23	14.43
81	196	1	1.03	9	26.15
100	392	1	1.97	498	1337.68
121	298	1	3.33	53	360.32
144	356	1	5.28		
169	422	1	815		
196	490	1	13.4		
225	566	1	15.82		
256	646	1	32.58		
289	732	1	32.58		
324	822	1	45.38		
361	920	1	59.95		

problem (1) where A and b are defined by

$$A = -2n^2 I + 2nee^T,$$
$$b_i = -1, \quad i = 1, \ldots, n/2; \qquad b_i = 0, \quad i = (n/2) + 1, \ldots, n.$$

Here I denotes the $(n \times n)$-identity matrix. We can easily see that the largest eigenvalue of A is zero. So A is an negative semi- definite matrix and we are faced with a concave minimization program. Then applying Algorithm BDCE to solve the problem

$$\min \left\{ \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle : x \in [0,1]^n \right\}, \tag{42}$$

and following Algorithm 2b, we obtain a solution in $\{0,1\}^n$.

For all test problems this algorithm stopped at the initialization step. The solution $\bar{x}^0$ of problem (QE^0) is near to the optimal solution of (QB) and Algorithm 2b provided this solution after one iteration. So the efficiency of the branchy and bound algorithm only depends upon the performance of BDCE for 21 problems of 'the difficult class of test problems' (40). To demonstrate the efficiency of Algorithm 1 we employed this and Safeguarding algorithym proposed by Moré &

Table 3. A 'difficult class of test problems' [30]

n	BDCE-ALG1	BDCE-SFG	n	BDCE-ALG1	BDCE-SFG
50	0.02	0.25	600	3.18	545.6
100	0.03	1.96	650	3.68	787.2
150	0.2	6.37	700	4.32	994.0
200	0.35	14.85	750	4.92	1233.5
250	0.53	32.77	800	5.57	1506.8
300	0.79	56.17	850	6.34	1822.6
350	1.05	101.43	900	7.49	2163.4
400	1.41	134.69	950	7.84	2633.4
450	1.66	220.01	1000	8.43	3022.7
500	2.17	269.3	1500	18.93	
550	2.71	416.4			

Abbreviations: BDCE-ALG1 – time in seconds of Algorithm BDCE with subprocedure being Algorithm 1; BDCE-SFG – time in seconds of Algorithm BDCE with subprocedure being Safeguarding algorithm.

Sorensen [29] for solving Problem (QE^0). In columns 4 and 5 of Table 3 we reported CPU time (in seconds) of Algorithm BDCE using Algorithm 1 and Safeguarding subprocedure, respectively. In Safeguarding algorithm we took $\sigma_1 = 10^{-5}$.

It is interesting to note that in [30] Pardalos et al. reported that for solving problems of this class the initial incumbent (obtained by heuristic algorithm) of their branch and bound algorithm was the solution but it took excessively long to prove the optimality.

5.3. A CLASS OF CONCAVE MINIMIZATION PROBLEM WHERE a IS A DENSE RANDOM MATRIX

In the third experiment we solved 40 problems of the form

$$\min\left\{\frac{1}{2}\langle x, Ax\rangle : x \in \{0,1\}^n\right\}. \tag{43}$$

We set $A = PDP^T$ for some orthogonal matrix P and a diagonal matrix D. $P = Q_1Q_2Q_3$, wheere $Q_j = I - 2\,w_j w_j^T/\|w_j\|^2$, $j = 1, 2, 3$ and the components w_j are random numbers in $(-2, 2)$. The diagonal elements of the matrix D are random numbers in $(-50, 0)$.

We took $\varepsilon = 0.01$ when $n < 40$ and $\varepsilon = 0.05$ when $n \geq 40$.

Table 4 indicates the performance (number of iteration and CPU times in seconds) of three algorithms BDCE, BDCEU and BBU. As in Tables 1 and 2, we present also the quantities NLB2 and NALB2 of BDCEU. We solved 5 problems of each dimension and took the average result.

Observe that for solving this class of problems BBU is more efficient than BDCE. Thus, the use of BDCEU is interesting for BDCE.

Table 4. Comparison between BDCE, BDCEU, BBU in the third experiment

	BDCE		BDCEU				BBU	
n	iter	CPU	iter	CPU	NLB2	NALB2	iter	CPU
10	1	0.02	1	0.02	0	0	4.6	0.04
20	30	1.02	13.6	0.6	15	5	3	0.07
30	95.6	5.96	69	7.2	79.6	22.3	19.6	1.28
40	15	1.03	1.6	0.33	2	2	1	0.08
50	239	43.8	52.2	18.7	56	24.6	59	12.1
60	111.6	18.5	5	1.62	3	3	1	0.15
80	1229	425.2	25.6	20.3	39	17	3.6	1.5
100	1939	1139.22	105	107.6	150	56	16	10.1

In Table 5 we reported in more detail the performance of BDCE for this class. The column labeled *itA2b* indicates the iteration for which Algorithm 2b provides an ε-global solution while the column labeled γ^0 (resp. OPT) indicates the upper bound at step initialization (resp. ε-optimal value).

Observing the quantities γ^0 in column 5 of Table 5 we see that Algorithm 2b provides either an ε-solution or a very good approximation for the optimal value at the very initialization step.

In the last experiment we study tyhe performance of BDCE for different tolerances ε and different branching procedures. In Table 6 we provide examples of the form (43) with the same data as in Table 5 indicating the performance of Algorithm BDCE for different tolerances ε. Table 6 indicates that in this experiment one can obtain the same ε-optimal solution with different values of ε.

It is interesting to investigate the influence of the branching strategy used in the algorithm. Table 7 gives the performance of Algorithm BDCE (number of iterations and CPU time in seconds) using different branching procedures (Rule 1 and the combined Rule 2 – Rule 1) for 6 problems of the form

$$\min\{F(x) := (1/2)\langle x, Ax\rangle : x \in [-1, 1]^n\}, \tag{44}$$

where A is negative semi-definite matrix.

The data was randomly generated in a similar way as described in the third experiment. Here the components w_j are random numbers in $(-50, 50)$ and the diagonal elements of the matrix D are random numbers in $(-100, 0)$. Of course we took the same tolerance ($\varepsilon = 0.05$) in two cases in order to obtain the same

Table 5. More details about the performance of BDCE

n	iter	CPU	itA2b	γ^0	OPT	n	iter	CPU	itA2b	γ^0	OPT
10	1	0.03	0	−97.398	−97.398	50	15	1.37	2	−845.89	−848.25
10	1	0.01	0	−147.684	−147.684	50	696	129.27	294	−565.08	−571.8771
10	1	0.02	9	−116.82	−118.36	50	9	0.78	0	−786.91	−786.91
20	81	2.9	2	−277.82	−282.40	60	186	31.85	34	−819.38	−821.5455
20	7	0.14	0	−310.36	−310.36	60	99	16.15	0	−805.85	−805.85
20	3	0.08	0	−301.08	−301.08	60	50	7.62	0	−868.14	−868.14
30	120	7.8	7	−421.61	−424.65	80	1123	276.17	56	−1049.13	−1050.2282
30	10	0.92	0	−339.78	−339.78	80	1092	569.43	61	−987.76	−990.0716
30	157	10.51	124	−371.33	−371.77	80	1473	430.00	131	−986.89	−994.67
40	43	2.88	0	−553.89	−553.89	100	1938	1138.32	6	−1197.47	−1203.22
40	1	0.10	0	−630.20	−630.20	100	2002	1142.22	2000	−1266.20	−1268.41
40	1	0.13	0	−575.09	−575.09						

Table 6. Behavior of BDCE for different tolerances ε

	$\varepsilon = 0.01$			$\varepsilon = 0.05$			$\varepsilon = 0.06$		
n	iter	CPU	OPT	iter	CPU	OPT	iter	CPU	OPT
30	120	7.8	−424.65	18	0.72	−424.32	2	0.07	−421.6
30	10	0.92	−339.78	9	0.63	−339.78	9	0.63	−339.78
30	157	10.51	−371.32	9	0.35	−371.32	9	0.35	−371.32
40	944	74.2	−553.89	43	2.88	−553.89	30	2.40	−553.89
40	36	3.1	−630.20	1	0.10	−630.20	2	0.10	−630.20
40	48	4.6	−575.09	1	0.13	−575.09	2	0.13	−575.09

Table 7. Behavior of BDCE for different branching procedures

	Rule 1		Rule 2 – Rule 1			Rule 1		Rule 2 – Rule 1	
n	iter	CPU	iter	CPU	n	iter	CPU	iter	CPU
10	235	3.36	118	1.96	15	3197	60.92	93	4.85
10	542	3.92	24	0.49	15	18444	400.20	3	0.31
10	389	5.72	20	0.31	15	1642	29.41	7	0.58

optimal solution. The rectangular bisection of a longest edge (Rule 1) is bad. The main disadvantage of this is that the generated ellipsoids do not provide a good lower bound: it is often happened that $\beta(K^{k_i}) \leq \beta(K^k)(i = 1, 2))$. So the lower bound is improved very very slowly.

Conclusion

We have proposed a new branch and bound algorithm based on d.c. optimization algorithms and ellipsoidal technique for nonconvex quadratic programming problems with box constraints. The main contribution of this paper is to give a good combination between local and global approaches: first we have provided a simple and efficient algorithm using the information of bounding procedure for finding a good local minima of box constrained quadratic problem; secondly, we have used the elippsoidal technique for bounding based on an efficient algorithm (the DCA) for the ball constrained quadratic problem we have studied also the new branching procedure in rectangular partition in order to well adapt to the new ellipsoidal bounding technique.

Preliminary experiments showed the efficiency as well as the limit of BDCE. The advantages are that either an ε-optimal solution or a good feasible point is obtained rapidly by Algorithm 2a (or 2b) and thereby a lot of number of generated ellipsoids are deleted from further consideration. Therefore, in general the algo-

rithm is efficient for a proper choice of tolerance ε (nevertheless this choice does not present always exactly the quality of the solution obtained, see e.g. Table 6). Moreover, the proposed algorithm is very efficient for some classes of problems (Tables 1, 2, 3). Again, both Algorithm 1 and Algorithm 2a or 2b (the DCA) for subproblems in bounding procedures are simple and not expensive. The limit is that, as often happened in a branch and bound algorithm, the lower bound is improved slowly. That is why we introduced the second process of lower bounding (BDCEU) in such a case. In the study of the performance of BDCE, an open question is that which ellipsoid can be chosen to warrant the best initial lower bound? Meanwhile our ellipsoid technique seems to be quite suitable.

References

1. Le Thi Hoai An and Pham Dinh Tao (1997), Solving a class of linearly constrained indefinite quadratic problems by d.c. algorithms, *Journal of Global Optimization* 11: 253–285.
2. Le Thi Hoai An, Pham Dinh Tao and Le Dung Muu (1996), Numerical solution for optimization over the efficient set by d.c. optimization algorithm, *Operations Research Letters* 19: 117–128.
3. F. Barahona, M. Junger and R.G. Reinelt (1989), Experiments in quadratic 0-1 programming, *Mathematical Programming* 44: 127– 137.
4. H.P. Benson (1990), Separable concave minimization via partial outer approximation and branch and bound, *Operations Research Letters* 9: 389–394.
5. I. Bieche, R. Maynard, R. Rammal and J.P. Uhry (1980), On the ground states of the frustration model of a spin glass by a matching method of graph theory, *J. Phys. A: Math. Gen.* 13: 2553–2576.
6. I.M. Bomze and G. Danninger (1993), A global optimization algorithm for concave quadratic problems, *SIAM Journal on Optimization* 3(4): 826–842.
7. I.M. Bomze and G. Danninger (1994), A finite algorithm for solving general quadratic problems, *Journal of Global Optimization* 4: 1– 16.
8. T.F. Coleman and L.A. Hulbert (1989), A direct active set algorithm for large sparse quadratic programs with simple bounds, *Mathematical Programming* 45: 373–406.
9. R.S. Dembo and U. Tulowitzki (1984), On the minimization of quadratic functions subject to box constraints, *SIAM J. Sci. Comp.*
10. R. Fletcher (1991), *Practical Methods of Optimization*, 2nd edn., Wiley-Interscience.
11. C.A. Floudas and V. Visweswaran (1994), Quadratic optimization, in: *Handbook of Global Optimization* (pp. 217–269), edited by R. Horst and P.M. Pardalos, Kluwer Academic Publishers, Dordrecht.
12. G.E. Forsythe and G.H. Golub (1965), On the stationary values of a second-degree polynomial on the unit sphere, *J. Soc. Indust. Appl. Math.* 13(4).
13. D.M. Gay (1981), Computing optimal locally constrained steps, *SIAM J. Sci. Stat. Comput.* 2: 186–197.
14. M. Grotschel, L. Laszlo and A. Schrijver (1993), *Geometric Algorithms and Combinatorial Optimization*, 2nd rev. edn., Springer Verlag, Berlin.
15. C.G. Han, P.M. Pardalos and Y. Ye (1990), Computational aspects of an interior point algorithm for quadratic programming problems with box constraints, in: *Large-Scale Numerical Optimization*, SIAM.
16. P. Hansen, B. Jaumard, M. Ruiz and J. Xiong (1991), Global minimization of indefinite quadratic functions subject to box constraints, *Technical Report G-91-54, GERAD*, École Polytechnique, Université McGill, Montréal, Canada.

17. R. Horst and P.T. Thach (1992), A decomposition method for quadratic minimization problems with integer variables, in P.M. Pardalos (ed.), Advances in *Optimization and Parallel Computing* (pp. 143–163), Elsevier Science Publishers, Amsterdam.
18. R. Horst and N.V. Thoai (1996), A new algorithm for solving the general quadratic programming problem, *Computational Optimization and Applications* 5: 39–48.
19. R. Horst and H. Tuy (1993), *Global Optimization: Deterministic Approaches*, 2nd rev. edn., Springer Verlag, Berlin.
20. B. Kalantari and J.B. Rosen (1987), Algorithm for global minimization of linearly constrained concave quadratic functions, *Mathematics of Operations Research* 12: 544–561.
21. A. Kamath and N. Karmarkar (1992), A continuous method for computing bounds in integer quadratic optimization problems, *Journal of Global Optimization*, 1992: 229–241.
22. N. Karmarkar, K.G. Ramakrishnan and G.C. Mauricio (1991), An interior point algorithm to solve computationally difficult set covering problems, *Mathematical Programming* 52(2): 597–618.
23. A. Kamath, N. Karmarkar, K.G. Ramakrishnan and G.C. Mauricio (1992), A continuous approach to inductive inference, *Mathematical Programming* 57B(2): 215–238.
24. S. Kaufman (1993), *The origins of Order: Sell-Organization and Selection in Evolution*, Oxford University Press.
25. K. Levenberg (1963), A method for the solution of certain non-linear problems in least squares, *Quarterly Appl. Math.* 2: 164–168.
26. S. Lucidi, L. Palagi and M. Roma (1998), On some properties of quadratic programs with a convex quadratic constraint, *SIAM Journal on Optimization* 8(1).
27. J. Martinez (1994), Local minimizers of quadratic functions on Euclidean balls and spheres, *SIAM Journal on Optimization* 4(1): 159– 176.
28. D.W. Marquardt (1963), An algorithm for least-squares estimation of nonlinear parameters, *J. SIAM* 11: 431–441.
29. J.J. Moré and D.C. Sorensen (1983), Computing a trust region step, *SIAM J. Sci. Stat. Comput.* 4: 553–572.
30. P.M. Pardalos and G. Rodgers (1990), Computational aspects of a branch and bound algorithm for quadratic zero-one programming, *Computing* 45: 131–144.
31. P.M. Pardalos and G. Rodgers (1990), Parallel branch and bound algorithms for unconstrained quadratic zero-one programming, in: *Impacts of Recent Computer Advances on Operations Research* (pp. 131–143) edited by R. Sharda et al., North-Holland, Amsterdam.
32. P.M. Pardalos, J.H. Glick and J.B. Rosen (1987), Global minimization of indefinite quadratic problems, *Computing* 39: 281– 291.
33. Pham Dinh Tao and E.B. Souad (1988), Duality in d.c. (difference of convex functions) optimization: Subgradient methods, Trends in *Mathematical Optimization, International Series of Numer. Math.* 84 (Birkhauser): 276–294.
34. Phan Dinh Tao (1989), *Méthodes numériques pour la minimisation globale d'une forme quadratique (convexe ou non convexe) sur une boule et une sphère euclidiennes*, Rapport de Recherche, Université Joseph-Fourier, Grenoble.
35. Phan Dinh Tao and S. Wang (1990), Training multi-layered neural network with a Trust region based algorithm, *Math. Modell. Numer. Anal.* 24(4): 523–553.
36. Pham Dinh Tao and Le Thi Hoai An (1995), Lagrangian stability and global optimality on nonconvex quadratic minimization over Euclidiean balls and spheres, *Journal of Convex Analysis* 2: 263– 276.
37. Pham Dinh Tao and Le Thi Hoai An (1998), D.c. optimization algorithm for solving the trust region sub-problem, *SIAM Journal on Optimization* 8(2): 1–30.
38. A.T. Phillips and J.B. Rosen (1988), A parallel algorithm for constrained concave quadratic global minimization, *Mathematical Programming* 42: 412–448.

39. A.T. Phillips and J.B. Rosen (1990), A parallel algorithm for partially separable non-convex global minimization: Linear constraints, *Annals of Operations Research* 25: 101–118.
40. S. Poljak, R. Rendl and H. Wolkowicz (1995), A recipe for semidefinite relaxation for (0, 1)-quadratic programming, *Journal of Global Optimization* 7: 51–73.
41. Thai Quynh Phong, Le Thi Hoai An and Pham Dinh Tao (1996), On the global solution of linearly constrained indefinite quadratic minimization problems by decomposition branch and bound method. *RAIRO, Recherche Opérationnelle* 30(1): 31–49.
42. N.V. Thoai (1994), A decomposition method in nonconvex mixed-integer programming, *Forschungsbericht Nr. 94-10*.
43. J.F. Toland (1979), On subdifferential calculus and duality in nonconvex optimization, *Bull. Soc. Math. France, Mémoire* 60: 177–183.
44. H. Tuy (1992), On nonconvex optimization problems with separated nonconvex variables, *Journal of Global Optimization* 1(3): 229– 244.
45. F. Rendl and H. Wolkowicz (1994), A semidefinite framework to trust region subproblems with application to large scale minimization, *CORR Report* 94–32, Department of Combinatorics and Optimization, University of Waterloo.
46. J.B. Rosen and P.M. Pardalos (1986), Global minimization of large scale constrained quadratic problem by separable programming. *Mathematical Programming* 34(2): 163–174.
47. S.A. Santos and D.C. Sorensen, A New Matrix-Free Algorithm for the Large-Scale Trust-region Sub-problem, *SIAM Journal on Optimization* (in prep.).
48. D.C. Sorensen (1982), Newton's method with a model trust region modification, *SIAM J. Numer. Anal*. 19(2): 409–426.
49. D.C. Sorensen (1992), Implicit application of Polynomial Filters in a K-step Arnoldi Method, *SIAM Journal on Matrix Analysis and Applications* 13: 357–385.
50. D.C. Sorensen (1997), Minimization of a large scale quadratic function subject to a spherical constraint, *SIAM Journal on Opimization* 7(1): 141–161.
51. S.A. Vavasis and R. Zippel (1990), Proving polynomial time for sphere-constrained quadratic programming, *Technical Report 90-1182*, Department of Computer Science, Cornell University, Ithaca, NY.
52. S.A. Vavasis (1991), *Nonlinear Optimization: Complexity Issues*, Oxford University Press.
53. S.A. Vavasis (1992), Approximation algorithms for indefinite quadratic programming, *Mathematical Programming* 57: 279–311.
54. Y. Ye (1992), On affine scaling algorithms for nonconvex quadratic programming, *Mathematical Programming* 56: 285–300.