Università degli Studi di Pisa
Dipartimento di Statistica e Matematica Applicata all'Economia

Report n. 234

Decomposition methods for nonconvex quadratic programs

Riccardo Cambini and Claudio Sodini

Pisa, Novembre 2002

- Stampato in Proprio -

Via Cosimo Ridolfi, 10 – 56124 PISA – Tel. Segr. Amm. 050 945231 Segr. Stud. 050 945317 Fax 050 945375
Cod. Fisc. 80003670504 - P. IVA 00286820501 - Web http://statmat.ec.unipi.it/ - E-mail: dipstat@ec.unipi.it

Decomposition methods for nonconvex quadratic programs

Riccardo Cambini - Claudio Sodini *
Dept. of Statistics and Applied Mathematics, University of Pisa
Via Cosimo Ridolfi 10, 56124 Pisa, ITALY
E-mail: cambric@ec.unipi.it, csodini@ec.unipi.it

November 28, 2002

Abstract

The aim of this paper is to suggest branch-and-bound schemes, based on a relaxation of the objective function, to solve nonconvex quadratic programs over a compact feasible region.

The various schemes are based on different d.c. decomposition methods applied to the quadratic objective function.

To improve the tightness of the relaxations, we also suggest to solve the relaxed problems with an algorithm based on the so called "optimal level solutions" parametrical approach.

Keywords Quadratic programming, optimal level solutions, d.c. optimization.
AMS - 2000 Math. Subj. Class. 90C20, 90C26, 90C31.
JEL - 1999 Class. Syst. C61, C63.

1 Introduction

In this paper we propose various solution methods for quadratic indefinite programs and the way they can be solved by means of branch and bound algorithms based on the partition of the feasible region and the relaxation of the objective function.

These problems have been approached in the literature in several ways (see for example [2, 3, 4, 6, 11, 12, 15, 16]); in particular, the proposed algorithms are based on convex relaxations obtained by means of a transformation of the objective function in a d.c. form [1, 7, 10, 13, 14].

In particular, we study different decompositions of the objective function $f(x) = \frac{1}{2}x^TAx + c^Tx$ in d.c. forms, which result to be different from the ones proposed in [1, 7, 13, 14] and which provide relaxations more tight than

*This paper has been partially supported by M.I.U.R. and C.N.R.

the ones usually given in the literature. Note that the relaxations used in this paper are not necessarily convex.

In Section 2 we first study how to decompose matrix A in the form $A = Q - \sum_{i=1}^{n-\nu_+(A)} d_i d_i^T$, where $Q \in \Re^{n\times n}$ is positive definite and $d_i \in \Re^n$ $\forall i$; in order to decrease numerical errors and computational complexity, such a decomposition is obtained without the use of eigenvalues and eigenvectors. By using the previous decomposition the following d.c. form of the objective function follows:

$$f(x) = \frac{1}{2}x^T Ax - \frac{1}{2}\sum_{i=1}^{n-\nu_+(A)} (d_i^T x)^2 + c^T x$$

and a relaxation is obtained linearizing the quadratic form $\sum_{i=1}^{n-\nu_+(A)}(d_i^T x)^2$. This allows us to suggest a branch and bound scheme based on a bipartition of the current feasible region. Finally, it is shown that the algorithm proposed in [5] allow us to solve, in the branch and bound scheme, subproblems having a more tight nonconvex relaxation.

In Section 3 the particular case of the box constrained problems is studied. A branch and bound scheme based on the decomposition of matrix A in the form $A = Q - dd^T - \mathrm{diag}(w)$, where Q is positive definite and $\mathrm{diag}(w)$ is a positive semidefinite diagonal matrix with diagonal elements given by the components of vector w, is given by means of the linearization of the quadratic form $x^T\mathrm{diag}(w)x$. Some decomposition procedures are proposed and compared with respect of the tightness of the corresponding relaxations.

2 A general approach

Let us consider the following quadratic program over a compact polyhedron.

Definition 2.1 We define the following quadratic program:

$$P: \begin{cases} \min f(x) = \frac{1}{2}x^T Ax + c^T x \\ x \in X = \{x \in \Re^n : Bx \geq b\} \end{cases}$$

where X is a compact polyhedron, $B \in \Re^{m\times n}$, $b \in \Re^m$, $c \in \Re^n$ and $A \in \Re^{n\times n}$ is any symmetric matrix. From now on we will denote also with (ν_+, ν_-, ν_0) the inertia of A, with $\nu_+(A) + \nu_-(A) + \nu_0(A) = n$. In other words, $\nu_+(A)$ is the number of positive eigenvalues of A, $\nu_-(A)$ is the number of negative ones, $\nu_0(A)$ is the algebraic multiplicity of the zero eigenvalue.

If A is positive semidefinite then f is convex and hence problem P can be solved by means of any of the known solving algorithms for convex quadratic programs.

The aim of this section is to propose a branch and bound scheme to solve problem P when A is not positive definite.

2.1 Preliminaries

It is well known that the decomposition method of Lagrange (see for all [9]), based on the "Law of Inertia" ([1]), given a symmetric matrix A provides a decomposition of the kind $A = Q - \sum_{i=1}^{h} d_i d_i^T$ where Q is positive semidefinite with $\text{rank}(Q) = \nu_+(A)$, $h = \nu_-(A)$ and $d_1, \ldots, d_h$ are linearly independent.

Such a procedure can be slightly modified as described in procedure ModLagrange(A,Q,k,$d_1,\ldots,d_k$), in order to obtain a decomposition of the kind

$$A = Q - \sum_{i=1}^{k} d_i d_i^T$$

where Q is positive definite, $k = \nu_-(A) + \nu_0(A) = n - \nu_+(A)$ and $d_1, \ldots, d_k$ are linearly independent.

Procedure **ModLagrange**(*inputs*: A; *outputs*: Q, k, $d_1, \ldots, d_k$)
T:=A; Q:=0; k:=0; used:= $[1, 1, \ldots, 1, 1] \in \Re^n$;
while T$\neq 0$ *do*
  if $T[i,i] = 0\ \forall i \in \{1, \ldots, n\}$
  then select $r \in \{1, \ldots, n\}$ *such that* row[T,r]$\neq 0$;
    Q[r,r]:=Q[r,r]+1; T[r,r]:=-1;
  else select $r \in \{1, \ldots, n\}$ *such that* $T[r,r] \neq 0$;
  end if;
  v:=row[T,r]; α:=T[r,r]; T:=T-$\frac{1}{\alpha}vv^T$; Q:=Q+$\frac{1}{|\alpha|}vv^T$; used[r]:=0;
  if $\alpha < 0$ *then* k:=k+1; $d_k := \sqrt{\frac{-2}{\alpha}}v$ *end if*;
end do;
for i *from* 1 *to* n *do*
  if used[i]=1 *then* Q[i,i]:=Q[i,i]+1; k:=k+1; d_k:=0; d_k[i]:=1; *end if*;
end do;
end proc. □

Remark 2.1 Note that, in procedure ModLagrange(A,Q,k,$d_1,\ldots,d_k$), at every iterative step it is $A = Q + T - \sum_{i=1}^{k} d_i d_i^T$, where Q is positive semidefinite and the vectors $d_1, \ldots, d_k$ are linearly independent. The procedure

[1] (The Law of Inertia for symmetric matrices [9]) Let $A \in \Re^{n\times n}$, $A \neq 0$, be a symmetric matrix and let $u_1, \ldots, u_r \in \Re^n \setminus \{0\}$ be $1 \leq r \leq n$ linearly independent vectors such that

$$A = \sum_{i=1}^{r} \alpha_i u_i u_i^T,$$

where $\alpha_i \in \{-1, 1\}\ \forall i = 1, \ldots, r$. Then the number of positive and the number of negative coefficients α_i are independent of the chosen set of linearly independent vectors $u_1, \ldots, u_r$.

stops when T becomes the null matrix (which happens, for the Law of Inertia, after $n - \nu_0(A)$ iterations of the while cycle) and the flags vector *used* is scanned (making $\nu_0(A)$ more vectors d_i). As a consequence, the output of the procedure is a decomposition of the kind $A = Q - \sum_{i=1}^{k} d_i d_i^T$, with $k = \nu_-(A) + \nu_0(A)$. Note finally that matrix Q is made using n linearly independent vectors, hence it is nonsingular. Since Q is positive semidefinite and nonsingular it results to be positive definite. □

Example 2.1 Applying procedure ModLagrange(A,Q,k,$d_1,\ldots,d_k$) to

$$A = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$$

we have:

- T:=A; Q:=0; k:=0; used:=[1,1];
- T[i,i]=0, i=1,2, hence: r:=1; Q[1,1]:=1; T[1,1]:=-1; v:=[-1,2]; α:=-1;

$$\begin{aligned} T &:= \begin{bmatrix} -1 & 2 \\ 2 & 0 \end{bmatrix} + \begin{bmatrix} 1 & -2 \\ -2 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix} \\ Q &:= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & -2 \\ -2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix} \end{aligned}$$

- $\alpha < 0$ hence: used[1]:=0; k:=1; $d_1^T := \sqrt{2}\,[-1, 2]$
- T[2,2]≠0 hence: r:=2; v:=[0,4]; α:=4; used[2]:=0;

$$\begin{aligned} T &:= \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix} - \frac{1}{4}\begin{bmatrix} 0 & 0 \\ 0 & 16 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ Q &:= \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix} + \frac{1}{4}\begin{bmatrix} 0 & 0 \\ 0 & 16 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ -2 & 8 \end{bmatrix} \end{aligned}$$

- since T=0 and used=0 we finally have:

$$A = Q - d_1 d_1^T = \begin{bmatrix} 2 & -2 \\ -2 & 8 \end{bmatrix} - \begin{bmatrix} 2 & -4 \\ -4 & 8 \end{bmatrix}$$

□

The previous procedure can be applied to matrix A of function f, so that function f can then be rewritten as:

$$f(x) = \frac{1}{2}x^T Q x - \frac{1}{2}\sum_{i=1}^{n-\nu_+(A)} (d_i^T x)^2 + c^T x$$

Note finally that, by means of $n-\nu_+(A)$ linear programs we can also compute the following values:

$$\underline{l}_i := \min_{x \in X} d_i^T x, \quad \overline{u}_i := \max_{x \in X} d_i^T x, \quad i = 1, \ldots, n - \nu_+(A) \tag{2.1}$$

so that:

$$\underline{l}_i \leq d_i^T x \leq \overline{u}_i \quad \forall x \in X, \ \forall i = 1, \ldots, n - \nu_+(A).$$

Remark 2.2 The problem of decomposing a symmetric matrix $A \in \Re^{n \times n}$ as the difference of two positive semidefinite matrices have been generally approached in the literature using the diagonal form of A (see for example [7, 13, 1, 14]), so that it is necessary to compute the n eigenvectors of A (hence to resolve n homogeneous linear systems). All the decompositions proposed in this paper are computed directly by means of at most n "pivoting-like" operations and without the need of computing eigenvalues and eigenvectors. Clearly, this chosen approach results to be less "expensive", with respect to both time-computing and numerical errors, than the one based on the diagonal form of A. □

2.2 Relaxations

In the branch and bound algorithm we are going to propose, the feasible region will be partitioned using the valued $d_i^T x$, $i = 1, \ldots, n - \nu_+(A)$. In particular, in the current step of the branch and bound algorithm the considered partition of the feasible region is of the kind $X \cap Y$ where:

$$\begin{aligned} \overline{Y} &= \{x \in \Re^n : \ \underline{l}_i \leq d_i^T x \leq \overline{u}_i \ \forall i = 1, \ldots, n - \nu_+(A)\}, \\ Y &= \{x \in \Re^n : \ l_i \leq d_i^T x \leq u_i \ \forall i = 1, \ldots, n - \nu_+(A)\} \subseteq \overline{Y}. \end{aligned}$$

Obviously, it is $X \cap \overline{Y} = X$. As a consequence function f can be relaxed over the current partition just linearizing the concave form $-\sum_{i=1}^{n-\nu_+(A)} (d_i^T x)^2$ over Y, thus obtaining the function:

$$\begin{aligned} f_Y(x) &= \frac{1}{2} x^T Q x - \frac{1}{2} \sum_{i=1}^{n-\nu_+(A)} [d_i^T x (l_i + u_i) - l_i u_i] + c^T x \\ &= \frac{1}{2} x^T Q x + \tilde{c}^T x + \tilde{c}_0 \end{aligned}$$

with:

$$\tilde{c} = c - \frac{1}{2} \sum_{i=1}^{n-\nu_+(A)} d_i (l_i + u_i) \quad \text{and} \quad \tilde{c}_0 = \frac{1}{2} \sum_{i=1}^{n-\nu_+(A)} l_i u_i$$

In particular, it is:

$$f(x) - f_Y(x) = -\frac{1}{8} \sum_{i=1}^{n-\nu_+(A)} (2 d_i^T x - (l_i + u_i))^2 + \frac{1}{8} \sum_{i=1}^{n-\nu_+(A)} (u_i - l_i)^2$$

so that the maximum error done linearizing over the current partitions Y the concave form $-\sum_{i=1}^{n-\nu_+(A)}(d_i^T x)^2$ is:

$$Err(f_Y) \;=\; \frac{1}{8}\sum_{i=1}^{n-\nu_+(A)}(u_i - l_i)^2 \tag{2.2}$$

In other words, for all $x \in X \cap Y$ it results:

$$0 \le f(x) - f_Y(x) \le Err(f_Y)$$

Remark 2.3 It is worth comparing the proposed approach with the ones given in the literature [7, 13, 1, 14]. First of all, in the literature both the objective function and the feasible region are usually transformed by means of a change of variables based on the eigenvectors of A, while in our approach just the objective function is decomposed leaving both the variables and the feasible region untouched. This implies that the original structure of the feasible region is maintained and that all numerical errors due to the computing of the eigenvectors are avoided.

Note finally that our approach refers to a general compact feasible region (not necessarily a box-constrained one) and that the relaxed problems are strictly convex ones (generally more easy to be solved than convex ones). □

2.3 Branch and bound

The results of the previous subsection allow us to relax Problem P over the current partitions Y with the following problem:

$$P_Y : \begin{cases} \min f_Y(x) \\ x \in X \cap Y \end{cases}$$

Note that P_Y is a strictly convex quadratic problem and can be solved with any of the algorithms known in the literature.

We are now able to suggest a branch and bound scheme based on the proposed relaxation. The main procedure just initialize the algorithm, call the recursive subprocedure "Explore$_1$()" and then provides the optimal solution.

Procedure **Solve$_1$**(P)
  determine a decomposition $A = Q - \sum_{i=1}^{n-\nu_+(A)} d_i d_i^T$;
  determine $\bar{l}_i$ and $\overline{u}_i$ $\forall i = 1, \ldots, n - \nu_+(A)$;
  $UB := +\infty$;
  Explore$_1(\overline{Y})$;
  x^* is the optimal solution and UB is the minimum value;
end proc.

The core of the algorithm is in the recursive procedure "Explore$_1$()" which is described below.

Procedure **Explore$_1$(Y)**
Let $\overline{x}$ be the optimal solution of P_Y;
if $f(\overline{x}) < UB$ *then* $UB := f(\overline{x})$; $x^* := \overline{x}$ *end if*;
if $f_Y(\overline{x}) < UB$ and $Err(f_Y) > \epsilon$ *then*
let $i \in \{1, \ldots, n - \nu_+(A)\}$ be such that $l_i < d_i^T \overline{x} < u_i$;
define $Y_1 = \{x \in Y : \ l_i \leq d_i^T x \leq d_i^T \overline{x}\}$;
define $Y_2 = \{x \in Y : \ d_i^T \overline{x} \leq d_i^T x \leq u_i\}$;
Explore$_1(Y_1)$;
Explore$_1(Y_2)$;
end if;
end proc.

Procedure "Explore$_1(Y)$" first look for the optimal solution $\overline{x}$ of the corresponding relaxed problem P_Y; if $\overline{x}$ allows to decrease the upper bound UB it is choosen as the new incumbent optimal solution. Then the value of the relaxed function $f_Y(\overline{x})$ is analyzed; if it is not lower than UB, that is:

$$f(\overline{x}) \geq \min_{x \in X \cap Y} f(x) \geq \min_{x \in X \cap Y} f_Y(x) = f_Y(\overline{x}) \geq UB$$

then it is impossible to improve the incumbent optimal solution exploring furthermore Y. In the case:

$$f_Y(\overline{x}) < UB \leq f(\overline{x})$$

then there exists an index $i \in \{1, \ldots, n - \nu_+(A)\}$ such that $l_i < d_i^T \overline{x} < u_i$ (if such an index does not exist it is $f_Y(\overline{x}) = f(\overline{x})$), hence it is possible to partition the set Y into two subsets Y_1 and Y_2 as defined before. The incumbent optimal solution can then be improved using the relaxed functions f_{Y_1} and f_{Y_2}, defined over the sets Y_1 and Y_2 respectively, since these functions provide a better approximation of f than f_Y. In order to guarantee the convergence of the algorithm, the partitioning of Y is stopped also when the maximum error $Err(f_Y)$ is not greater than a fixed value $\epsilon > 0$.

Remark 2.4 In the recursive procedure "Explore$_1$()" we have to select an index $i \in \{1, \ldots, n - \nu_+(A)\}$ such that $l_i < d_i^T \overline{x} < u_i$. This can be done in several ways; one of the possible choice is to choose the index j such that:

$$u_j - l_j = \max_{i \in \{1, \ldots, n - \nu_+(A)\}, l_i < d_i^T \overline{x} < u_i} \{u_i - l_i\}$$

which allows us to decrease as much as possible the maximum error $Err(f_Y)$ done linearizing the concave form over the current partition. □

Example 2.2 Let us solve with the described approach the problem

$$\begin{cases} \min f(x_1,x_2) = \frac{1}{2}[x_1,x_2]\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ (x_1,x_2) \in X = \{(x_1,x_2) \in \Re^2 : -1 \le x_1 \le 3,\ -2 \le x_2 \le 3\} \end{cases}$$

- The matrix in the quadratic form of the objective function is decomposed as described in Example 2.1, so that

$$f(x_1,x_2) = \frac{1}{2}[x_1,x_2]\begin{bmatrix} 2 & -2 \\ -2 & 8 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - (-x_1+2x_2)^2$$

- we get

$$\underline{l}_1 = \min_{(x_1,x_2)\in X}(-x_1+2x_2) = -7 \quad , \quad \overline{u}_1 = \max_{(x_1,x_2)\in X}(-x_1+2x_2) = 7$$

attained in $(x_1,x_2) = (3,-2)$ and $(x_1,x_2) = (-1,3)$, respectively

- $UB := +\infty$; $\overline{Y} = \{-7 \le -x_1+2x_2 \le 7\}$;

$(P_{\overline{Y}})$ $f_{\overline{Y}}(x_1,x_2) = x_1^2 + 4x_2^2 - 2x_1x_2 - 49$; the optimal solution is $(0,0)$; $UB := f(0,0) = 0$; $x^* := (0,0)$; since $f_{\overline{Y}}(0,0) = -49 < UB$ we get $Y_1 = \{-7 \le -x_1+2x_2 \le 0\}$ and $Y_2 = \{0 \le -x_1+2x_2 \le 7\}$

(P_{Y_1}) $f_{Y_1}(x_1,x_2) = x_1^2 + 4x_2^2 - 2x_1x_2 - 7x_1 + 14x_2$; the optimal solution is $\left(\frac{7}{3},-\frac{7}{6}\right)$; $UB := f\left(\frac{7}{3},-\frac{7}{6}\right) = -\frac{49}{9}$; $x^* := \left(\frac{7}{3},-\frac{7}{6}\right)$; since $f_{Y_1}\left(\frac{7}{3},-\frac{7}{6}\right) = -\frac{49}{3} < UB$ we get $Y_{1,1} = \{-7 \le -x_1+2x_2 \le -\frac{14}{3}\}$ and $Y_{1,2} = \{-\frac{14}{3} \le -x_1+2x_2 \le 0\}$

$(P_{Y_{1,1}})$ $f_{Y_{1,1}}(x_1,x_2) = x_1^2+4x_2^2-2x_1x_2-\frac{35}{3}x_1+\frac{70}{3}x_2+\frac{98}{3}$; the optimal solution is $(3,-2)$; $UB := f(3,-2) = -12$; $x^* := (3,-2)$; $f_{Y_{1,1}}(3,-2) = -12 \ge UB$

$(P_{Y_{1,2}})$ $f_{Y_{1,2}}(x_1,x_2) = x_1^2+4x_2^2-2x_1x_2+\frac{14}{3}x_1-\frac{28}{3}x_2$; the optimal solution is $(0,0)$; $f(0,0) = 0$; $f_{Y_{1,2}}(0,0) = 0 > UB$

(P_{Y_2}) $f_{Y_2}(x_1,x_2) = x_1^2 + 4x_2^2 - 2x_1x_2 + 7x_1 - 14x_2$; the optimal solution is $\left(-1,\frac{3}{2}\right)$; $f\left(-1,\frac{3}{2}\right) = -3$; since $f_{Y_2}\left(-1,\frac{3}{2}\right) = -15 < UB$ we get $Y_{2,1} = \{0 \le -x_1+2x_2 \le 4\}$ and $Y_{2,2} = \{4 \le -x_1+2x_2 \le 7\}$

$(P_{Y_{2,1}})$ $f_{Y_{2,1}}(x_1,x_2) = x_1^2 + 4x_2^2 - 2x_1x_2 + 4x_1 - 8x_2$; the optimal solution is $\left(-1,\frac{3}{4}\right)$; $f\left(-1,\frac{3}{4}\right) = -\frac{3}{2}$; $f_{Y_{2,1}}\left(-1,\frac{3}{4}\right) = -\frac{21}{4} > UB$

$(P_{Y_{2,2}})$ $f_{Y_{2,2}}(x_1,x_2) = x_1^2 + 4x_2^2 - 2x_1x_2 + 11x_1 - 22x_2 + 28$; the opt. solution is $\left(-1,\frac{5}{2}\right)$; $f\left(-1,\frac{5}{2}\right) = -5$; $f_{Y_{2,2}}\left(-1,\frac{5}{2}\right) = -7 > UB$

- the opt. solution of the problem is $x^* := (3,-2)$ with $f(3,-2) = -12$.

□

2.4 Further enhancements

In [5] a finite algorithm to solve problems of the kind

$$\begin{cases} \min g(x) = \frac{1}{2}x^T Q x - \frac{1}{2}(d^T x)^2 + c^T x \\ x \in X = \{x \in \Re^n : Bx \geq b\} \end{cases} \tag{2.3}$$

has been proposed, using the so called "optimal level solutions" approach [5, 8]. This algorithm can be used to improve the tightness of the relaxations used in the branch and bound schemes described in the previous subsections.

Let us first recall that in the previous subsections function f has been decomposed as:

$$f(x) = \frac{1}{2}x^T Q x - \frac{1}{2}\sum_{i=1}^{k}(d_i^T x)^2 + c^T x \quad \text{where } k = n - \nu_+(A)$$

with an error due to the initial relaxation given by:

$$Err(f_{\overline{Y}}) = \frac{1}{8}\sum_{i=1}^{k}(\overline{u}_i - \overline{l}_i)^2$$

In order to decrease as much as possible the error of the relaxations, let us denote with $j \in \{1, \ldots, k\}$ an index such that:

$$\overline{u}_j - \overline{l}_j = \max_{i=1,\ldots,k}\{\overline{u}_i - \overline{l}_i\} \tag{2.4}$$

where $\overline{u}_i$ and $\overline{l}_i$ are the bounds defined in (2.1). The partition of the feasible region used in the current step of the branch and bound algorithm is now of the kind $X \cap Y_j$ where:

$$\begin{aligned} \overline{Y}_j &= \{x \in \Re^n : \overline{l}_i \leq d_i^T x \leq \overline{u}_i \ \forall i = 1, \ldots, k, \ i \neq j\}, \\ Y_j &= \{x \in \Re^n : l_i \leq d_i^T x \leq u_i \ \forall i = 1, \ldots, k, \ i \neq j\} \subseteq \overline{Y}_j. \end{aligned}$$

Function f can then be relaxed over the current partition just linearizing the concave form $-\sum_{i=1, i\neq j}^{k}(d_i^T x)^2$ over Y_j with the function:

$$\begin{aligned} g_{Y_j}(x) &= \frac{1}{2}x^T Q x - \frac{1}{2}(d_j^T x)^2 - \frac{1}{2}\sum_{i=1,\ i\neq j}^{k}[d_i^T x(l_i + u_i) - l_i u_i] + c^T x \\ &= \frac{1}{2}x^T Q x - \frac{1}{2}(d_j^T x)^2 + \overline{c}^T x + \overline{c}_0 \end{aligned}$$

with:

$$\overline{c} = c - \frac{1}{2}\sum_{i=1,\ i\neq j}^{k} d_i(l_i + u_i) \quad \text{and} \quad \overline{c}_0 = \frac{1}{2}\sum_{i=1,\ i\neq j}^{k} l_i u_i$$

Note that function g_{Y_j} is of the same kind of the objective function of problem (2.3). It then results:

$$f(x) - g_{Y_j}(x) = -\frac{1}{8}\sum_{i=1,\ i\neq j}^{k}(2d_i^T x - (l_i + u_i))^2 + \frac{1}{8}\sum_{i=1,\ i\neq j}^{k}(u_i - l_i)^2$$

so that the maximum error done linearizing over the current partition Y_j the concave form $-\sum_{i=1,i\neq j}^{k}(d_i^T x)^2$ is:

$$Err(g_{Y_j}) \ = \ \frac{1}{8}\sum_{i=1,\ i\neq j}^{k}(u_i - l_i)^2 = Err(f_{Y_j}) - \frac{1}{8}(\overline{u}_j - \overline{l}_j)^2 \qquad (2.5)$$

In other words, for all $x \in X \cap Y_j$ it is:

$$0 \leq f(x) - g_{Y_j}(x) \leq Err(g_{Y_j}) < Err(f_{Y_j})$$

hence g_{Y_j} is more tight than f_{Y_j}.

Problem P can then be relaxed over the current partition Y_j with the problem:

$$G_{Y_j} : \begin{cases} \min g_{Y_j}(x) \\ x \in X \cap Y_j \end{cases}$$

which can be solved by means of the algorithm proposed in [5]. Note that this relaxation is more tight than P_{Y_j} but, from a computational point of view, G_{Y_j} (which is not a positive definite quadratic program) is more time-expensive to be solved than P_{Y_j} (which is a strictly convex program). Obviously, if $\nu_+(A) = n-1$, that is to say that A has one nonpositive eigenvalue, the problem can be solved directly by the algorithm proposed in [5], without the need of any branch and bound step.

The branch and bound scheme based on the relaxation G_{Y_j} is analogous to the one described in the previous section. The main procedure becomes:

Procedure **Solve$_2$**(P)
  determine a decomposition $A = Q - \sum_{i=1}^{n-\nu_+(A)} d_i d_i^T$;
  determine $\overline{l}_i$ and $\overline{u}_i$ $\forall i = 1, \ldots, n - \nu_+(A)$;
  determine j as in (2.4); $UB := +\infty$;
  Explore$_2(\overline{Y}_j)$;
  x^* is the optimal solution and UB is the minimum value;
end proc.

while the recursive procedure "Explore$_3$()" is:

Procedure **Explore$_2$**(Y_j)
  Let $\overline{x}$ be the optimal solution of G_{Y_j};
  if $f(\overline{x}) < UB$ *then* $UB := f(\overline{x})$; $x^* := \overline{x}$ *end if*;
  if $g_{Y_j}(\overline{x}) < UB$ *and* $Err(g_{Y_j}) > \epsilon$ *then*

let $i \in \{1, \ldots, n-\nu_+(A),\ i \neq j\}$ be such that $l_i < d_i^T \overline{x} < u_i$;
define $Y_j^1 = \{x \in Y_j :\ l_i \leq d_i^T x \leq d_i^T \overline{x}\}$;
define $Y_j^2 = \{x \in Y_j :\ d_i^T \overline{x} \leq d_i^T x \leq u_i\}$;
Explore$_2(Y_j^1)$;
Explore$_2(Y_j^2)$;
end if;
end proc.

3 Box constrained case

Let us consider now the following particular case of problem P:

$$P_B : \begin{cases} \min f(x) = \frac{1}{2} x^T A x + c^T x \\ x \in X_B = \{x \in \Re^n : \tilde{l}_i \leq x_i \leq \tilde{u}_i,\ i = 1, \ldots, n\} \end{cases}$$

that is the case where P is a box constrained problem.

Obviously, problem P_B can be solved as described in the previous section; recall that these approaches require that $n - \nu_+(A)$ linear programs must be solved in the initialization of the branch and bound.

In this section we aim to show that different decompositions of matrix A allow us to avoid the computing of the solutions of such linear programs.

3.1 Relaxations and branch and bound

Our approach is based on the decomposition of matrix A in the following form ([2]):

$$A = Q - dd^T - \mathrm{diag}(w) \tag{3.1}$$

where $Q \in \Re^{n \times n}$ is symmetric and positive definite, $d, w \in \Re^n$, $w \geq 0$ and $\mathrm{diag}(w)$ is the positive semidefinite diagonal matrix with diagonal elements given by the components of vector w. Note that the vector d can be equal to the zero vector.

Such a decomposition allows us to rewrite function f as follows:

$$f(x) = \frac{1}{2} x^T Q x - \frac{1}{2} \left(d^T x\right)^2 - \frac{1}{2} \sum_{i=1}^{n} w_i x_i^2 + c^T x$$

The feasible region can now be partitioned using the variables x_i, $i = 1, \ldots, n$, such that $w_i > 0$. In particular, in the current step of the branch and bound algorithm the considered partition of the feasible region is of the kind $X_B \cap W$ where:

$$\begin{aligned} \overline{W} &= \{\tilde{l}_i \leq x_i \leq \tilde{u}_i\ \forall i \text{ such that } w_i > 0\} \supseteq X_B, \\ W &= \{l_i \leq x_i \leq u_i\ \forall i \text{ such that } w_i > 0\} \subseteq \overline{W}. \end{aligned}$$

[2]Note that in [10, 13] a decomposition of this kind (with $d = 0$ and $\mathrm{diag}(w) = kI_n$, $k > 0$ large enough) was studied. Another decomposition of this kind with $d = 0$, based on diagonal dominance, has been proposed in [13].

As a consequence, problem P_B can be relaxed over the current partition just linearizing the concave form $-\frac{1}{2}\sum_{i=1}^{n} w_i x_i^2$ over W, thus obtaining the problem:

$$P_{BW}: \begin{cases} \min f_W(x) \\ x \in X_B \cap W \end{cases}$$

where:

$$\begin{aligned} f_W(x) &= \frac{1}{2}x^T Q x - \frac{1}{2}\left(d^T x\right)^2 - \frac{1}{2}\sum_{i=1}^{n} w_i[x_i(l_i+u_i) - l_i u_i] + c^T x \\ &= \frac{1}{2}x^T Q x - \frac{1}{2}\left(d^T x\right)^2 + \overline{c}^T x + \overline{c}_0 \end{aligned}$$

with:

$$\overline{c}_0 = \frac{1}{2}\sum_{i=1}^{n} w_i l_i u_i \quad \text{and} \quad \overline{c} = (\overline{c}_i) \text{ where } \overline{c}_i = c_i - \frac{1}{2}w_i(l_i+u_i)\ \forall i = 1,\ldots,n.$$

Note that:

- in the case $d = 0$ problem P_{BW} is a strictly convex quadratic problem and problem P_B can be solved with a branch and bound scheme analogous to the one described in Subsection 2.3;
- in the case $d \neq 0$ problem P_{BW} is of the kind (2.3) and can be solved by means of the algorithm proposed in [5], so that problem P_B can be approached with a branch and bound scheme analogous to the one described in Subsection 2.4.

Since:

$$f(x) - f_W(x) = -\frac{1}{8}\sum_{i=1}^{n} w_i(2x_i - (l_i+u_i))^2 + \frac{1}{8}\sum_{i=1}^{n} w_i(u_i-l_i)^2$$

the maximum error done linearizing the concave form $-\frac{1}{2}\sum_{i=1}^{n} w_i x_i^2$ over the current partition W is:

$$Err(f_W) = \frac{1}{8}\sum_{i=1}^{n} w_i(u_i-l_i)^2 \tag{3.2}$$

reached at points $\tilde{x} \in X_B \cap W$ such that:

$$\tilde{x}_i = \frac{1}{2}(l_i+u_i) \quad \forall i \in \{1,\ldots,n\} \text{ such that } w_i > 0;$$

in particular, for all $x \in X_B \cap W$ it results:

$$0 \leq f(x) - f_W(x) \leq Err(f_W).$$

It is now clear that in order to decrease the maximum error $Err(f_W)$ in the various branch and bound steps we have to use decompositions of the kind (3.1) with a vector w having many zero components and positive ones with a small value.

The branch and bound scheme which solves problem P_B is analogous to the ones described in the previous sections. The main procedure is:

Procedure **Solve$_3$**(P_B)
  determine a decomposition $A = Q - dd^T - \mathrm{diag}(w)$;
  $UB := +\infty$;
  Explore$_3(\overline{W})$;
  x^* is the optimal solution and UB is the minimum value;
end proc.

The core of the algorithm is in the following recursive procedure "Explore$_3$()":

Procedure **Explore$_3$**(W)
  Let $\overline{x}$ be the optimal solution of P_{BW} ([3]);
  if $f(\overline{x}) < UB$ *then* $UB := f(\overline{x})$; $x^* := \overline{x}$ *end if*;
  if $f_W(\overline{x}) < UB$ *and* $Err(f_W) > \epsilon$ *then*
    let $i \in \{1, \ldots, n\}$ be such that $w_i > 0$ and $l_i < \overline{x}_i < u_i$;
    define $W_1 = \{x \in W : \ l_i \leq x_i \leq \overline{x}_i\}$;
    define $W_2 = \{x \in W : \ \overline{x}_i \leq x_i \leq u_i\}$;
    Explore$_3(W_1)$;
    Explore$_3(W_2)$;
  end if;
end proc.

Remark 3.1 In the recursive procedure "Explore$_3$()" we have to select an index $i \in \{1, \ldots, n\}$ such that $w_i > 0$ and $l_i < \overline{x}_i < u_i$. In order to decrease as much as possible the maximum error $Err(f_W)$ done linearizing the concave form we can choose the index j such that:

$$w_j(u_j - l_j)^2 = \max_{i \in \{1,\ldots,n\}, w_i > 0, l_i < \overline{x}_i < u_i} \{w_i(u_i - l_i)^2\}$$

□

3.2 Decomposition procedures

Let us now provide some procedures which decompose matrix A in the form:

$$A = Q - dd^T - \mathrm{diag}(w)$$

[3] If $d = 0$ the optimal solution $\overline{x}$ of P_{BW} is determined by means of any of the known algorithm for positive definite quadratic problems; in the case $d \neq 0$ it is determined by means of the algorithm proposed in [5]

as described in the previous subsection. Note that, in order to obtain relaxations as tight as possible, the proposed procedures are aimed to determine a vector w with as few positive components as possible.

The first procedure, that is `Minor`(A,Q,w), provides a decomposition of the kind (3.1) with $d = 0$ and is based on the well known characterization of positive definite matrices based on the positiveness of all their NW minors. In this procedure the following notations are used:

- Q_k is the $k \times k$ NW submatrix of Q,
- $Q_{k\setminus i}$ is the $k \times k$ NW submatrix of Q without its i-th row and column.

Procedure **Minor**(*inputs*: A; *outputs*: Q, w)
Q:=A; w:=0;
if Q[1,1]$\leq$ 0 *then* w[1]:=1$-$Q[1,1]; Q[1,1]:=1 *end if*;
for k *from* 2 *to* n *do*
  if det(Q_k) $\leq$ 0
  then if Q[k,k]$\leq$ 0 *then* h:=k; den:=det(Q_{k-1});
    else found:=*false*;
      for i *from* 1 *to* k-1 *do*
        if not(found) *and* w[i]$>$ 0 *and* det($Q_{k\setminus i}$) $>$ 0
        then h:=i; den:=det($Q_{k\setminus i}$); found:=*true*;
        end if;
      end do;
      if not(found) *then* h:=k; den:=det(Q_{k-1}) *end if*;
    end if;
    val:=1 $-$ $\frac{\det(Q_k)}{\text{den}}$; w[h]:=w[h]+val; Q[h,h]:=Q[h,h]+val;
  end if;
end do;
end proc.

□

Note that, in order to obtain a vector w with as few positive components as possible, if the k-th NW minor of Q is nonpositive the procedure `Minor`(A,Q,w) first try to make it positive by increasing an already positive component of w, while a new positive component of w is created just if this is not possible.

The limit of this approach is that the diagonal elements are analyzed in a fixed order, that is from the first to the last one. Better results may be obtained with algorithms which consider the diagonal elements not in a fixed order, such as the procedure `Decomp1`(A,Q,w) which provides a decomposition of the kind (3.1) with $d = 0$ and which is based on pivoting operations similar to the ones used in the Lagrange's method. Note that in this procedure we denote with row[T,r] the r-th row of matrix T.

Procedure **Decomp1**(*inputs*: A; *outputs*: Q, w)
T:=A; Q:=0; w:=0; used:= $[1, 1, \ldots, 1, 1] \in \Re^n$;
while T$\neq$ 0 *do*
  if $T[i, i] \leq 0\ \forall i \in \{1, \ldots, n\}$
  then select $r \in \{1, \ldots, n\}$ *such that* row[T,r]$\neq$ 0;
    w[r]:=1-T[r,r]; T[r,r]:=1;
  else select $r \in \{1, \ldots, n\}$ *such that* $T[r, r] > 0$ and $T - \frac{1}{T[r,r]} vv^T$ has
    as much positive diagonal elements as possible, where v=row[T,r];
  end if;
  v:=row[T,r]; α:=T[r,r]; T:=T-$\frac{1}{\alpha}$vv^T; Q:=Q+$\frac{1}{|\alpha|}$vv^T; used[r]:=0;
end do;
w:=w+used; Q:=Q+diag(used);
end proc. □

Note that in order to obtain a vector w with as few positive components as possible, the procedure Decomp1(A,Q,w) first uses the positive diagonal elements of the temporary matrix T, then it analyzes the nonpositive ones.

In the next example the procedures Minor(A,Q,w) and Decomp1(A,Q,w) are applied to a given matrix A, showing that Decomp1(A,Q,w) may provide a decomposition with a matrix diag(w) having smaller rank.

Example 3.1 Let us apply procedures Minor(A,Q,w) and Decomp1(A,Q,w) to the following matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}$$

We obtain the following decompositions:

$$\text{minor:} \quad \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 0 \\ 3 & 0 & 46 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 45 \end{bmatrix}$$

$$\text{decomp1:} \quad \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 14 & 2 & 3 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 13 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

hence, in this case, Decomp1(A,Q,w) provides a vector w with a number of positive components smaller than Minor(A,Q,w).

A vector w with a smaller number of positive components can be obtained with the procedure Decomp2(A,Q,d,w) which represent an improvement of the procedure Decomp1(A,Q,w), note that this procedure provides a vector d which may be different from zero.

Finally, it is worth studying how many positive components the vector w in (3.1) may have.

***Procedure* Decomp2**(*inputs*: A; *outputs*: Q, d, w)
T:=A; Q:=0; w:=0; used:= $[1,1,\ldots,1,1] \in \Re^n$; d:=0; found:=*false*;
while T$\neq$ 0 *do*
if T$[i,i] \leq 0$ $\forall i \in \{1,\ldots,n\}$
then if found
then select $r \in \{1,\ldots,n\}$ *such that* row[T,r]$\neq$ 0;
w[r]:=1-T[r,r]; T[r,r]:=1;
else select $r \in \{1,\ldots,n\}$ *such that* v=row[T,r]$\neq$ 0 and
$T - \frac{1}{\alpha}vv^T$ has as much positive diagonal elements
as possible, where α is given by:

$$\alpha = \begin{cases} T[r,r] & \text{if } T[r,r] < 0 \\ -1 & \text{if } T[r,r] = 0 \end{cases};$$

if $T[r,r] = 0$ *then* Q[r,r]:=Q[r,r]+1; T[r,r]:=-1 *end if*;
end if;
else select $r \in \{1,\ldots,n\}$ *such that* $T[r,r] > 0$ and $T - \frac{1}{T[r,r]}vv^T$ has
as much positive diagonal elements as possible, where v=row[T,r];
end if;
v:=row[T,r]; α:=T[r,r]; T:=T-$\frac{1}{\alpha}$vv^T; Q:=Q+$\frac{1}{|\alpha|}$vv^T; used[r]:=0;
if $\alpha < 0$ *then* d:=$\sqrt{\frac{-2}{\alpha}}$v; found:=*true*; *end if*;
end do;
w:=w+used; Q:=Q+diag(used);
end proc.

□

Theorem 3.1 *Let $A \in \Re^{n\times n}$ be a symmetric matrix and denote with $n_+(A)$ the number of positive diagonal elements of A and with A_+ the submatrix of A obtained deleting the rows and the columns of A corresponding to its nonpositive diagonal elements. Consider also the following decompositions:*

$$\begin{aligned} A &= Q_1 - \text{diag}(w_1) \\ A &= Q_2 - dd^T - \text{diag}(w_2),\ d \neq 0 \end{aligned}$$

where Q_1 and Q_2 are positive definite, $d \in \Re^n$, $d \neq 0$, and $w_1, w_2 \in \Re^n$, $w_1, w_2 \geq 0$. Then:

i) $\text{rank}(\text{diag}(w_1)) \geq n - \nu_+(A_+) \geq n - \min\{n_+(A), \nu_+(A)\}$,

ii) $\text{rank}(\text{diag}(w_2)) \geq \nu_-(A) + \nu_0(A) - 1 = n - \nu_+(A) - 1$.

Proof *i)* Matrix A, by means of a permutations of its rows and columns, can be rewritten as:

$$\Pi A \Pi^T = \begin{bmatrix} A_+ & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

Since $Q = A + \mathrm{diag}(w)$ is a positive definite matrix we have that

$$\Pi Q \Pi^T = \begin{bmatrix} A_+ + \mathrm{diag}(w_{11}) & A_{12} \\ A_{21} & A_{22} + \mathrm{diag}(w_{22}) \end{bmatrix}$$

is positive definite too. As a consequence $A_+ + \mathrm{diag}(w_{11})$ is positive definite so that, for the "Law of Inertia", w_{11} has at least $n_+(A) - \nu_+(A_+)$ positive components; it results also that the diagonal elements of $A_{22} + \mathrm{diag}(w_{22})$ are all positive and hence, being the diagonal elements of A_{22} all nonpositive, all the $n - n_+(A)$ components of w_{22} are positive. The obtained conditions imply that vector w has at least $n - \nu_+(A_+)$ positive components. The whole result then follows noticing that both $n_+(A) \geq \nu_+(A_+)$ and $\nu_+(A) \geq \nu_+(A_+)$.

ii) Follows directly from the "Law of Inertia". □

The next property follows directly from procedures `Decomp1(A,Q,w)` and `Decomp2(A,Q,d,w)` and from the "Law of Inertia".

Property 3.1 *Let $A \in \Re^{n \times n}$ be a symmetric matrix which is not positive definite (hence $\nu_+(A) \leq n-1$) and let $A = Q - dd^T - \mathrm{diag}(w)$ be a decomposition obtained with procedure "decomp2". Then:*

i) $d \neq 0$,

ii) *if* $\nu_-(A) \leq 1$ *then* $\mathrm{rank}(\mathrm{diag}(w)) = n - \nu_+(A) - 1$.

Let finally $A = Q_1 - \mathrm{diag}(w_1)$ be a decomposition obtained with procedure "decomp1"; then:

iii) $\mathrm{rank}(\mathrm{diag}(w)) \leq \mathrm{rank}(\mathrm{diag}(w_1)) - 1$.

Example 3.2 Let us apply procedures "minor", "decompl" and "decomp2" to the following matrix:

$$A = \begin{bmatrix} -2 & -2 & -2 \\ -2 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix}$$

We obtain the following decompositions:

minor:
decompl:
$$\begin{bmatrix} -2 & -2 & -2 \\ -2 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -2 & -2 \\ -2 & 5 & 1 \\ -2 & 1 & 14 \end{bmatrix} - \begin{bmatrix} 3 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 14 \end{bmatrix}$$

decomp2:
$$\begin{bmatrix} -2 & -2 & -2 \\ -2 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 4 & 5 \\ 2 & 5 & 8 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} [2,2,2] - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

hence, the use of vector $d \neq 0$ in the decomposition allow us to decrease the rank of $\mathrm{diag}(w)$ from 3 to 1.

3.3 Solution schemes

In the previous subsection we have already pointed out that the maximum error done using the relaxed function f_W over the current partition W is:

$$Err(f_W) = \frac{1}{8}\sum_{i=1}^{n} w_i(u_i - l_i)^2$$

As a consequence, in order to decrease the maximum error in the various branch and bound steps we have to use decompositions of the kind

$$A = Q - dd^T - \mathrm{diag}(w)$$

with a vector w having many zero components and positive ones with a small value. In this light several schemes, corresponding to different ways of decomposing matrix A, can be suggested to solve problem P_B:

(1) A is decomposed with procedure `Minor`(A,Q,w) and $d = 0$.

(2) A is decomposed with procedure `Decomp1`(A,Q,w) and $d = 0$. It has been shown in Example 3.1 that this decomposition may provide smaller errors than the ones given by (1).

(3) A is decomposed with procedure `Decomp1`(A,Q,w), a positive component of w, say w_j, is chosed, the vector $d = \sqrt{w_j}e_j \neq 0$ is defined ([4]) and the j-th component of w is then set to zero, that is $w_j := 0$. For example, in order to decrease as much as possible the error we can choose the index $j \in \{1, \ldots, n\}$ such that:

$$w_j(\tilde{u}_j - \tilde{l}_j)^2 = \max_{i=1,\ldots,n}\{w_i(\tilde{u}_i - \tilde{l}_i)^2\} \tag{3.3}$$

This is clearly an improvement over the scheme suggested in (2).

(4) A is decomposed with procedure `Decomp2`(A,Q,d,w). It has been shown in Example 3.2 that this decomposition may provide smaller errors than the ones given by (1), (2) and (3).

(5) Given a predefined vector $d \neq 0$, $A+dd^T$ is decomposed with procedure `Decomp1`($A+dd^T$,Q,w). This scheme could be useful when the vector d is chosed a priori, for example in order to avoid initialization problems in the solving algorithm.

Remark 3.2 In [10, 13] decompositions of the kind $Q = A - \mathrm{diag}(w)$ (that is of the kind (3.1) with $d = 0$) are suggested with $\mathrm{diag}(w) = kI_n$ or with $\mathrm{diag}(w)$ computed using diagonal dominance properties; these decompositions provide a vector w having many (or all) positive components with large

[4] We denote with e_j the j-th column of the $n \times n$ identity matrix.

values. As a consequence, the obtained relaxations are affected by a large error $Err(f_W)$.

The procedures `Minor(A,Q,w)`, `Decomp1(A,Q,w)` and `Decomp1(A,Q,d,w)` described in the previous subsection try to reduce as much as possible the maximum error $Err(f_W)$ by determining a vector w with many zero components and positive ones with a small value. In this way, we may obtain convex relaxations more tight than the ones given in [10, 13]. □

Example 3.3 Let us solve the problem described in Example 2.2 with the approach (1). With procedure `Minor(A,Q,w)` we get:

$$\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}$$

so that $f(x_1, x_2) = \left(\frac{1}{2}x_1^2 + \frac{5}{2}x_2^2 + 2x_1x_2\right) - \frac{1}{2}x_1^2 - \frac{5}{2}x_2^2$.

- $UB := +\infty$; $\overline{W} = X$;

($P_{\overline{W}}$) $f_{\overline{W}}(x_1, x_2) = \left(\frac{1}{2}x_1^2 + \frac{5}{2}x_2^2 + 2x_1x_2\right) - x_1 - \frac{5}{2}x_2 - \frac{33}{2}$; the optimal solution is $(0, \frac{1}{2})$; $UB := f(0, \frac{1}{2}) = 0$; $x^* := (0, \frac{1}{2})$; since $f_{\overline{W}}(0, \frac{1}{2}) = -\frac{137}{8} < UB$ we get $W_1 = \{-2 \leq x_2 \leq \frac{1}{2}\}$ and $W_2 = \{\frac{1}{2} \leq x_2 \leq 3\}$

(P_{W_1}) $f_{W_1}(x_1, x_2) = \left(\frac{1}{2}x_1^2 + \frac{5}{2}x_2^2 + 2x_1x_2\right) - x_1 + \frac{15}{4}x_2 - 4$; the optimal solution is $(3, -\frac{39}{20})$; $UB := f(3, -\frac{39}{20}) = -\frac{117}{10}$; $x^* := (3, -\frac{39}{20})$; since $f_{W_1}(3, -\frac{39}{20}) = -\frac{1921}{160} < UB$ we get $W_{1,1} = \{-2 \leq x_2 \leq -\frac{39}{20}\}$, $W_{1,2} = \{-\frac{39}{20} \leq x_2 \leq \frac{1}{2}\}$

($P_{W_{1,1}}$) $f_{W_{1,1}}(x_1, x_2) = \left(\frac{1}{2}x_1^2 + \frac{5}{2}x_2^2 + 2x_1x_2\right) - x_1 + \frac{79}{8}x_2 + \frac{33}{4}$; the optimal solution is $(3, -2)$; $UB := f(3, -2) = -12$; $x^* := (3, -2)$; $f_{W_{1,1}}(3, -2) = -12 \geq UB$

($P_{W_{1,2}}$) $f_{W_{1,2}}(x_1, x_2) = \left(\frac{1}{2}x_1^2 + \frac{5}{2}x_2^2 + 2x_1x_2\right) - x_1 + \frac{29}{8}x_2 - \frac{63}{16}$; the opt. solution is $(3, -\frac{77}{40})$; $f(3, -\frac{77}{40}) = -\frac{231}{20}$; $f_{W_{1,2}}(3, -\frac{77}{40}) = -\frac{7489}{640} > UB$;

(P_{W_2}) $f_{W_2}(x_1, x_2) = \left(\frac{1}{2}x_1^2 + \frac{5}{2}x_2^2 + 2x_1x_2\right) - x_1 - \frac{35}{4}x_2 + \frac{9}{4}$; the optimal solution is $\left(-1, \frac{43}{20}\right)$; $f\left(-1, \frac{43}{20}\right) = -\frac{43}{10}$; $f_{W_2}\left(-1, \frac{43}{20}\right) = -\frac{1249}{160} > UB$

- the opt. solution of the problem is $x^* := (3, -2)$ with $f(3, -2) = -12$.

□

References

[1] Barrientos O. and R. Correa, "An algorithm for global minimization of linearly constrained quadratic functions", *Journal of Global Optimization*, vol.16, pp.77-93, 2000.

[2] Best M.J. and B. Ding, "Global and local quadratic minimization", *Journal of Global Optimization*, vol.10, pp.77-90, 1997.

[3] Best M.J. and B. Ding, "A decomposition method for global and local quadratic minimization", *Journal of Global Optimization*, vol.16, pp.133-151, 2000.

[4] Bomze I.M. and G. Danninger, "A finite algorithm for solving general quadratic problems", *Journal of Global Optimization*, vol.4, pp.1-16, 1994.

[5] Cambini R. and C. Sodini, "A finite algorithm for particular d.c. quadratic programming problem", *Annals of Operations Research*, vol.117, pp.33-49, 2002.

[6] Churilov L. and M. Sniedovich, "A concave composite programming perspective on D.C. programming", in *Progress in Optimization*, edited by A. Eberhard, R. Hill, D. Ralph and B.M. Glover, *Applied Optimization*, vol.30, Kluwer Academic Publishers, Dordrecht, 1999.

[7] De Angelis P.L., Pardalos P.M. and G. Toraldo, "Quadratic Programming with Box Constraints", in *Developments in Global Optimization*, edited by I.M. Bomze et. al., Kluwer Academic Publishers, Dordrecht, pp.73-93, 1997.

[8] Ellero A., "The optimal level solutions method", *Journal of Information & Optimization Sciences*, vol.17, n.2, pp.355-372, 1996.

[9] Gantmacher F.R., "The theory of matrices", vol.1, Chelsea Publishing Company, New York, 1960.

[10] Hiriart-Urruty J.B., "Generalized differentiability, duality and optimization for problems dealing with differences of convex functions", in *Convexity and Duality in Optimization*, Lecture Notes in Economics and Mathematical Systems, vol.256, Springer-Verlag, 1985.

[11] Horst R. and H. Tuy, "Global Optimization", Springer-Verlag, Berlin, 1990.

[12] Horst R., Pardalos P.M. and N.V. Thoai, "Introduction to Global Optimization", *Nonconvex Optimization and Its Applications*, vol.3, Kluwer Academic Publishers, Dordrecht, 1995.

[13] Le Thi Hoai An and Pham Dinh Tao, "Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms", *Journal of Global Optimization*, vol.11, pp.253-285, 1997.

[14] Rosen J.B. and P.M. Pardalos, "Global minimization of large scale constrained concave quadratic problems by separable programming", *Mathematical Programming*, vol.34, pp.163-174, 1986.

[15] Tuy H., "D.C. Optimization: theory, methods and algorithms", in *Handbook of Global Optimization*, edited by R. Horst and P.M. Pardalos, *Nonconvex Optimization and Its Applications*, vol.2, Kluwer Academic Publishers, Dordrecht, pp.149-216, 1995.

[16] Tuy H., "Convex Analysis and Global Optimization", *Nonconvex Optimization and Its Applications*, vol.22, Kluwer Academic Publishers, Dordrecht, 1998.