Università di Pisa

Dipartimento di Informatica

Technical Report : TR-95-11

Testing Optimality for Quadratic 0-1 Problems

Paolo Carraresi, Fernanda Farinaccio, Federico Malucelli

December 03, 1996

ADDR: Corso Italia 40,56125 Pisa,Italy. TEL: +39-50-887111. FAX: +39-50-887226

Testing optimality for quadratic 0-1 problems[1]

by

Paolo Carraresi[†], Fernanda Farinaccio[2] and Federico Malucelli[3]

Abstract: *The issue tackled is testing whether a given solution of a quadratic 0-1 problem is optimal. The paper presents an algorithm based on the necessary and sufficient optimality condition introduced by Hirriart-Urruty for general convex problems. A measure of the quality of the solution is provided. Computational results show the applicability of the method. The method is extended to constrained quadratic 0-1 problems such as quadratic assignment and quadratic knapsack .*

Keywords: *Quadratic 0-1 programming, Optimality testing.*

1. Introduction

Quadratic 0-1 problems have a particular relevance in modeling many practical applications. A large number of combinatorial problems can be formulated as the unconstrained maximization of pseudo-Boolean functions (real valued functions in 0-1 variables) [6-8]. Moreover, the maximization of any pseudo-Boolean function can be reduced to the maximization of a quadratic pseudo-Boolean function, as noted in [19]. Other constrained 0-1 quadratic problems, such as Quadratic Knapsack, Quadratic Assignment and Quadratic Semi-Assignment, arise from an uncounted number of applications [4, 13, 16]. All these problems, in general, are very difficult to solve, even for relatively small instances; however many heuristic algorithms which attain very good solutions can be found in the literature. Usually, a crucial point is to certify the good quality of the solutions found by the heuristics, since, for most of the problems, the upper bounds are very loose. In a previous paper [2] it has been proved that testing the optimality of a given solution of any NP-hard combinatorial optimization problem cannot be easier than solving the problem. In that paper the authors also investigated an approximation scheme for testing the ε-optimality of a given solution for a fixed value of ε. Other complexity issues regarding local and global optima are discussed in [15, 17]. A review of most global conditions of optimality is contained in [9].The ε-optimality necessary and sufficient conditions used in this paper are based on the results obtained by Hirriart-Urruty in [10] for the maximization of a quadratic function on a convex set.

---

1 This work has been supported by Progetto Finalizzato Trasporti 2, Italian National Reseach Council, grant no. 93.01799.PF4.

† Professor Paolo Carraresi died suddenly and unexpectedly on March 5, 1994. At the time of his death this work was almost completed. The other two authors wrote the paper finding inspiration in the suggestions, the ideas and all they could learn working under the advise of Professor Carraresi.

2 Dipartimento di Informatica, Università di Pisa, Corso Italia 40, I-56125 Pisa, Italy. e_mail: farinacc@di.unipi.it.

3 Istituto di Elettronica, Università di Perugia, Via G. Duranti 1/A-1, I-06131 Perugia, Italy. e_mail: maluc@istel.ing.unipg.it.

In the present paper we introduce a new scheme for testing the optimality of a given solution for a quadratic 0-1 unconstrained problem. The proposed scheme expands on the results described in [2]. In particular, the approach presented here, instead of being an approximation scheme as the one in [2], is an exact algorithm, since it stops determining whether the tested solution is optimal or not. However, at any moment during the computation, it gives an upper bound to the absolute error approximation of the tested solution. The paper is organized as follows: in Section 2 the optimality and the ε-optimality conditions of a feasible solution are presented along with some properties. In Section 3 the algorithm for testing optimality is described in detail, while in Section 4 a method to evaluate the approximation of the solution is illustrated. The implementational aspects of the algorithm are discussed in Section 5. Section 6 contains some computational results. The last Section reports some concluding remarks and notes on the extension of the method to other quadratic constrained problems.

2. Notation and ε-optimality conditions

Consider the following quadratic 0-1 maximization problem:

$$\max \{\tfrac{1}{2}x^TQx + c^Tx: x \in \{0,1\}^n\}, \tag{2.1}$$

where, without loss of generality, we can assume that Q is a positive-definite symmetric matrix. We can always reduce to this case by suitably transforming Q and c as reported for example in [5]. Due to convexity, any optimal solution $\bar{x}$ of the continuous relaxation of (2.1) is an extreme point of $[0,1]^n$; hence any optimal solution of (2.1) is also optimal for

$$\max \{\tfrac{1}{2}x^TQx + c^Tx: x \in [0,1]^n\}, \tag{2.2}$$

and the two problems can be considered equivalently.

Given a convex polyhedron $S \subset \boldsymbol{R}^n$ and a convex function $g: \boldsymbol{R}^n \to \boldsymbol{R}$, a necessary and sufficient condition for $\bar{x} \in S$ to be a global maximum of g over S is that [10]

$$\partial_\varepsilon g(\bar{x}) \subset N_\varepsilon(S;\bar{x}), \ \forall\ \varepsilon > 0, \tag{2.3}$$

where $\partial_\varepsilon g(\bar{x})$ denotes the ε-subdifferential of g at $\bar{x}$, i.e.

$$\partial_\varepsilon g(\bar{x}) = \{u \in \boldsymbol{R}^n: g(x) - g(\bar{x}) \geq u^T(x - \bar{x}) - \varepsilon, \ \forall x \in \boldsymbol{R}^n\},$$

and $N_\varepsilon(S;\bar{x})$ the set of ε-normal directions to S at $\bar{x}$, i.e.,

$$N_\varepsilon(S;\bar{x}) = \{u \in \boldsymbol{R}^n: u^T(x - \bar{x}) \leq \varepsilon, \ \forall x \in S\}.$$

In our case, that is when $g(x) = \frac{1}{2}x^TQx + c^Tx$, it is:

$$\partial_\varepsilon g(\bar{x}) = \{u \in \boldsymbol{R}^n: (u-\bar{q})^TQ^{-1}(u-\bar{q}) \leq 2\varepsilon\}, \tag{2.4}$$

where $\bar{q} = Q\bar{x} + c$.

The set of ε-normal directions to $S=[0,1]^n$ at $\bar{x}$, when $\bar{x}$ is one of the vertices of S, that is $\bar{x} \in \{0,1\}^n$, can be given explicitly by applying the definition. Define $I_0(\bar{x}) = \{i: \bar{x}_i = 0\}$ and $I_1(\bar{x}) = \{i: \bar{x}_i = 1\}$. Since the set of extreme points of $[0,1]^n$ is $\{0,1\}^n$, then $u \in N_\varepsilon([0,1]^n; \bar{x})$ if and only if

$$\sum_{i \in I_0(\bar{x})} u_i x_i + \sum_{i \in I_1(\bar{x})} u_i (x_i - 1) \le \varepsilon, \quad \forall x \in \{0,1\}^n. \tag{2.5}$$

It can be seen that (2.5) reduces to only one inequality, that is:

$$\sum_{i \in I_0(\bar{x})} \max\{0, u_i\} + \sum_{i \in I_1(\bar{x})} \max\{0, -u_i\} \le \varepsilon$$

For the sake of simplicity, further on we will denote $\max\{0,u_i\}$ and $\max\{0,-u_i\}$ by u_i^+ and u_i^-, respectively.

Let us introduce the following notation to characterize the supporting hyperplanes of $N_\varepsilon([0,1]^n; \bar{x})$. Consider the diagonal matrix

$$A = \begin{bmatrix} a_1 & & 0 \\ & \circ & \\ 0 & & a_n \end{bmatrix}, \text{ where } a_i = \begin{cases} 1, & i \in I_0(\bar{x}), \\ -1, & i \in I_1(\bar{x}), \end{cases}$$

the 2^n-1 supporting hyperplanes of $N_\varepsilon([0,1]^n; \bar{x})$ are given by

$$\{u \in \boldsymbol{R}^n : y^T A u = \varepsilon\} \qquad y \in \{0,1\}^n, \ y \neq 0. \tag{2.6}$$

The vector y in (2.6) will be called the *characteristic vector* of the corresponding hyperplane.

Definition 2.1

The value $L(\bar{x})$ will be called the *lower threshold* of $\bar{x}$ if and only if

$$\partial_\varepsilon g(\bar{x}) \subset N_0([0,1]^n; \bar{x}), \ \forall \varepsilon \in [0, L(\bar{x})].$$

Definition 2.2

The value $U(\bar{x})$ will be called an *upper threshold* of $\bar{x}$ if and only if

$$(\partial_{U(\bar{x})} g(\bar{x}) \subset N_{U(\bar{x})}([0,1]^n; \bar{x})) \Rightarrow (\partial_\varepsilon g(\bar{x}) \subset N_\varepsilon([0,1]^n; \bar{x}) \ \forall \varepsilon \ge U(\bar{x}))$$

Lemma 2.1 [2]

For any solution $\bar{x}$, $L(\bar{x})$ and $U(\bar{x})$ exist finite and $\bar{x}$ is optimal if and only if

$$\partial_\varepsilon g(\bar{x}) \subset N_\varepsilon([0,1]^n; \bar{x}), \qquad \forall \ \varepsilon \in (L(\bar{x}), U(\bar{x}))$$

Definition 2.3

Given a real value $\bar{\varepsilon}>0$, a solution $\bar{x}$ is an $\bar{\varepsilon}$-optimal solution if and only if

$$\max \{\tfrac{1}{2}x^TQx + c^Tx: x \in \{0,1\}^n\} - \tfrac{1}{2}\bar{x}^TQ\bar{x} + c^T\bar{x} \leq \bar{\varepsilon}.$$

Lemma 2.2 [10]

Given $\bar{\varepsilon}>0$, and $\bar{x}\in\{0,1\}^n$, $\bar{x}$ is an $\bar{\varepsilon}$-optimal solution if and only if

$$\partial_\varepsilon g(\bar{x}) \subset N_{\varepsilon+\bar{\varepsilon}}([0,1]^n;\bar{x}),\ \forall\varepsilon>0.$$

As a consequence of Lemma 2.2 we have [2]:

Theorem 2.3

If

$$\partial_\varepsilon g(\bar{x}) \subset N_\varepsilon([0,1]^n;\bar{x}), \qquad \forall\varepsilon\in[0,\tau'] \text{ and } \forall\varepsilon\in[\tau'',+\infty),$$

then $\bar{x}$ is $(\tau''-\tau')$-optimal.

3. The testing algorithm

On the basis of properties stated in Section 2, we can devise a procedure to test the optimality of a given solution $\bar{x}$. The idea is:

- to approximate the parametric elliptic set $\partial_\varepsilon g(\bar{x})$ with a polyhedron $T(\varepsilon)$ parametric in ε, such that $\partial_\varepsilon g(\bar{x})\subset T(\varepsilon)$ for each $\varepsilon\in(L(\bar{x}),U(\bar{x}))$,
- to check whether $T(\varepsilon)$ is included in $N_\varepsilon([0,1]^n;\bar{x})$, for each value of ε, and, if not,
- to iteratively refine the approximation.

This method stops either when $T(\varepsilon)\subset N_\varepsilon([0,1]^n;\bar{x}))$, for each value of ε, or when it can be proved that there is an ε such that $\partial_\varepsilon g(\bar{x})\not\subset N_\varepsilon([0,1]^n;\bar{x})$.

Initially $T(\varepsilon)$ is given by any polyhedron which expands with ε, and has all the supporting hyperplanes always tangent to $\partial_\varepsilon g(\bar{x})$, for every value of ε. At each iteration we check whether:

i) all the parametric vertices $v(\varepsilon)$ of $T(\varepsilon)$ are included in $N_\varepsilon([0,1]^n;\bar{x})$, for every value of ε, or, on the contrary,

ii) if there are some $v(\varepsilon)\in T(\varepsilon)$ for which there exists a value ε' such that $v(\varepsilon')\notin N_{\varepsilon'}([0,1]^n;\bar{x})$.

In the first case, the solution $\bar{x}$ is optimal as $\partial_\varepsilon g(\bar{x})\subset N_\varepsilon([0,1]^n;\bar{x})$, for each $\varepsilon\in(L(\bar{x}),U(\bar{x}))$, while, in the second case, the approximation $T(\varepsilon)$ can be refined. The

refinement of $T(\varepsilon)$ is achieved by considering the intersection of $T(\varepsilon)$ with an additional parametric halfspace corresponding to one of the inequalities defining $N_\varepsilon([0,1]^n;\bar{x})$ which is violated by $v(\varepsilon')$. This halfspace is translated in such a way that it is always tangent to $\partial_\varepsilon g(\bar{x})$. If the parametric tangent point $\bar{u}(\varepsilon)$ between the new supporting hyperplane of $T(\varepsilon)$ and $\partial_\varepsilon g(\bar{x})$, does not belong to $N_\varepsilon([0,1]^n;\bar{x})$ for some value of ε, then $\bar{x}$ is not optimal as $\partial_\varepsilon g(\bar{x}) \not\subset T(\varepsilon)$. The procedure is summarized in Table 1 where $V(T(\varepsilon))$ denotes the set of vertices of polyhedron $T(\varepsilon)$.

```
Procedure Opt(x̄);
Begin
  Compute Upper and Lower Thresholds U(x̄), L(x̄);
  Let T(ε) be the initial parametric polyhedron;
  While ∃ v(ε)∈ V(T(ε)): ∃ε'∈ ( L(x̄),U(x̄)]), v(ε')∉ N_ε'([0,1]^n;x̄) Do
  Begin
     Select v(ε)∈ V(T(ε)): ∃ε'∈ (L(x̄),U(x̄)]),v(ε')∉ N_ε'([0,1]^n;x̄);
     Find y: y^T Au(ε')>ε';
     Determine function h(ε) such that the hyperplane y^T Au(ε)=h(ε) is tangent to ∂_ε g(x̄)
        ∀ ε∈ (L(x̄),U(x̄));
     Let ū(ε) be the set of tangent points between y^T Au(ε)=h(ε) and ∂_ε g(x̄) ∀ ε∈ (L(x̄),U(x̄));
     If ∃ ε': ū(ε')∉ N_ε'([0,1]^n;x̄) Then Return ("not optimal")
     Else T(ε):=T(ε)∩{y∈ R^n: y^T Au(ε)≤h(ε)}
  End
  Return ("optimal")
End.
```

Table 1

Let us analyze in detail the most important steps of the procedure.

Tangent hyperplanes and vertices of $T(\varepsilon)$

Any supporting hyperplane of $N_\varepsilon([0,1]^n;\bar{x})$ with gradient Ay, in order to intersect $\partial_\varepsilon g(\bar{x})$ in at the most one point, must be of the form [2]:

$$y^T Au(\varepsilon) = a\sqrt{\varepsilon} + b, \tag{3.1}$$

where $a = \sqrt{2y^T AQAy}$ and $b=y^T A\bar{q}$.

The expression of the tangent point between the hyperplane and $\partial_\varepsilon g(\bar{x})$ is:

$$\bar{u}(\varepsilon)= \left(\frac{2\varepsilon}{y^T AQAy}\right)^{1/2} QAy + \bar{q}. \tag{3.2}$$

Given that all the tangent hyperplanes are of the form (3.1), we obtain that $T(\varepsilon)$ can be represented as the following parametric system of inequalities:

$$Du(\varepsilon) \le d\sqrt{\varepsilon} + D\bar{q},$$

where D is the matrix whose rows are given by the gradients of the hyperplanes defining $T(\varepsilon)$ and d is a suitable vector.

Consider the elliptic set $\partial'_\varepsilon g(\bar{x})$ obtained by translating $\partial_\varepsilon g(\bar{x})$ so that it is centered in the origin, and let $T'(\varepsilon)$ be the corresponding parametric polyhedron containing $\partial'_\varepsilon g(\bar{x})$. It is $\partial'_\varepsilon g(\bar{x})=\{u-\bar{q}:u\in\partial_\varepsilon g(\bar{x})\}$ and $T'(\varepsilon)=\{u-\bar{q}: u\in T(\varepsilon)\}$; hence $T'(\varepsilon)$ is defined by the system:

$$Du(\varepsilon) \le b\sqrt{\varepsilon}$$

All the vertices of $T'(\varepsilon)$ are defined by a feasible basis of D. Let B be a set of n row indices which identify a subset of linearly independent rows of D, and let D_B denote the corresponding basic matrix. The parametric vertex $v'_B(\varepsilon)$ associated with a feasible basis B is given by:

$$v'_B(\varepsilon) = D_B^{-1} d_B \sqrt{\varepsilon}$$

where $d_B \in \boldsymbol{R}^n$ is obtained by considering the elements of d corresponding to the indices in B.

Hence the vertices $v_B(\varepsilon)$ of $T(\varepsilon)$ are given by:

$$v_B(\varepsilon) = D_B^{-1} d_B \sqrt{\varepsilon} + \bar{q}, \qquad \text{for each feasible basis } B.$$

Upper and Lower thresholds

A value for $L(\bar{x})$ can be computed as the maximum ε for which the inclusion $\partial_\varepsilon g(\bar{x})\subset N_0([0,1]^n;\bar{x})$ holds true. It is assumed that $\bar{q}$ is strictly contained in $N_0([0,1]^n;\bar{x})$ otherwise $\bar{x}$ is not optimal [2], then the minimum ε such that $\partial_\varepsilon g(\bar{x})$ is tangent to a face of $N_0([0,1]^n;\bar{x})$ defines $L(\bar{x})$.

$$L(\bar{x}) = \min\{\frac{\bar{q}_i^2}{2q_{ii}} : i=1,\ldots,n\}.$$

A value for $U(\bar{x})$ is obtained by considering the expansion speed of $\partial_\varepsilon g(\bar{x})$ with respect to $N_\varepsilon([0,1]^n;\bar{x})$ when ε increases. Let λ_{max} denote the maximum eigenvalue of Q. The Hausdorff distance between $\partial_\varepsilon g(\bar{x})$ and $\partial_0 g(\bar{x})$ is given by $(2\lambda_{max}\varepsilon)^{1/2}$ [11], while the Hausdorff distance between $N_\varepsilon([0,1]^n;\bar{x})$ and $N_0([0,1]^n;\bar{x})$ is ε. Hence when $\varepsilon > \frac{\lambda_{max}}{2}$, $N_\varepsilon([0,1]^n;\bar{x})$ grows faster than $\partial_\varepsilon g(\bar{x})$. Then, if $\partial_\varepsilon g(\bar{x}) \subset N_\varepsilon([0,1]^n;\bar{x})$ for $\varepsilon = \frac{\lambda_{max}}{2}$, one the more reason the inclusion will hold for $\varepsilon > \frac{\lambda_{max}}{2}$.

Hence

$$U(\bar{x}) = \frac{\lambda_{max}}{2}.$$

Determination of $h(\varepsilon)$

Given a hyperplane with gradient Ay, our aim is to determine a function $h(\varepsilon)$ such that the parametric hyperplane $y^TAu(\varepsilon)=h(\varepsilon)$ is always tangent to $\partial_\varepsilon g(\bar{x})$, that is the set $\{u: y^TAu = h(\varepsilon)\}$ intersects $\partial_\varepsilon g(\bar{x})$ in exactly one point $\bar{u}(\varepsilon)$, for each value of ε. Function $h(\varepsilon)$ can be easily determined considering the expression of $\bar{u}(\varepsilon)$ defined by (3.2). Hence $h(\varepsilon) = y^TA\,\bar{u}(\varepsilon)$.

Initial parametric polyhedron $T(\varepsilon)$

A possible way to determine an initial parametric polyhedron $T(\varepsilon)$ containing $\partial_\varepsilon g(\bar{x})$, $\forall\varepsilon>0$, is described next. Consider n hyperplanes with gradients Ae_i, $i=1,\ldots,n$, where e_i is the i-th unit vector; for each hyperplane we have to determine the function $h_i(\varepsilon)$ such that $u^TAe_i = h_i(\varepsilon)$ is tangent to $\partial_\varepsilon g(\bar{x})$, $\forall\varepsilon>0$. The expression of the tangent point between the i-th hyperplane and $\partial_\varepsilon g(\bar{x})$ is given by the following vector of linear functions of $\sqrt{\varepsilon}$:

$$\bar{u}^i(\varepsilon) = \sqrt{\frac{2\varepsilon}{q_{ii}}}\,QAe_i + \bar{q}, \qquad i=1,\ldots,n,\ \forall\varepsilon>0.$$

The intersection between all the hyperplanes $u^TAe_i = h_i(\varepsilon)$ gives the vertex $v^0(\varepsilon)$ of the parametric cone whose single components are given by:

$$v_i^0(\varepsilon) = \bar{u}_i^i(\varepsilon), \qquad i=1,\ldots,n.$$

Theorem 3.1

$T(\varepsilon)$ has dimension n, for each $\varepsilon>0$.

Proof:

$\partial_\varepsilon g(\bar{x})$ has dimension n for $\varepsilon>0$ as Q is positive definite; as $\partial_\varepsilon g(\bar{x})\subseteq T(\varepsilon)\subset \boldsymbol{R}^n$, the thesis follows.

❑

Testing $u(\varepsilon)\in N_\varepsilon([0,1]^n;\bar{x})$

Consider any parametric point $u(\varepsilon)$. According to the definition of $N_\varepsilon([0,1]^n;\bar{x})$, to determine whether $u(\varepsilon)\in N_\varepsilon([0,1]^n;\bar{x})$ for each value of $\varepsilon\in(L(\bar{x}),U(\bar{x}))$, we have to study for which values of ε the following function is non positive:

$$f(u(\varepsilon)) = \sum_{i\in I_0(\bar{x})} u_i^+(\varepsilon) + \sum_{i\in I_1(\bar{x})} u_i^-(\varepsilon) - \varepsilon.$$

Note that all the parametric points (either tangent points or vertices of $T(\varepsilon)$), that have to be tested during the algorithm, are of the form:

$$u(\varepsilon) = t\sqrt{\varepsilon} + \bar{q}. \tag{3.3}$$

This means that $f(u(\varepsilon))$ is a continuous piecewise quadratic function in $\sqrt{\varepsilon}$ with at the most n points where the first derivative is not continuous (*breakpoints*). The breakpoints correspond to changes of sign of functions $u_i(\varepsilon)$, $i=1,\ldots,n$. As $u_i(\varepsilon)$ are linear functions of $\sqrt{\varepsilon}$, when ε grows they have at the most one change of sign, hence the maximum operation which gives $u_i^+(\varepsilon)$ and $u_i^-(\varepsilon)$ induces a discontinuity in the first derivative of $f(u(\varepsilon))$.

Let us call $\varepsilon_1 = L(\bar{x}) \le \varepsilon_2 \le \ldots \le_k \varepsilon = U(\bar{x})$ ($k \le n+1$) the values of ε which induce a change of sign in $u_i(\varepsilon)$, $i=1,\ldots,n$. When ε varies inside interval $[\varepsilon_j, \varepsilon_{j+1}]$ ($j=1,\ldots,k$-1), $f(u(\varepsilon))$ reduces to a parabola of the form:

$$\pi_j(\varepsilon) = -\varepsilon + r_j\sqrt{\varepsilon} + s_j.$$

As a consequence, given a parametric point $u(\varepsilon)$ of the form (3.3), to verify that $u(\varepsilon) \in N_\varepsilon([0,1]^n; \bar{x})$ for each value of $\varepsilon \in (L(\bar{x}), U(\bar{x}))$, it suffices to see if $\pi_j(\varepsilon) \le 0$ for $\varepsilon \in [\varepsilon_j, \varepsilon_{j+1}]$ ($j=1,\ldots,k$-1), hence it suffices to find the zeros of $\pi_j(\varepsilon)$ solving a second degree equation. This proves the following theorem.

Theorem 3.2

Given a parametric point $u(\varepsilon) = t\sqrt{\varepsilon} + \bar{q}$, testing $u(\varepsilon) \in N_\varepsilon([0,1]^n; \bar{x})$ for each $\varepsilon \in (L(\bar{x}), U(\bar{x}))$ can be done in $O(n)$ time.

Finding the characteristic vector of a violated inequality

Assume that $u(\varepsilon) \notin N_\varepsilon([0,1]^n; \bar{x})$ for some values of ε. When we fix the value of ε, testing whether $u(\varepsilon) \in N_\varepsilon([0,1]^n; \bar{x})$ reduces to verifying the inequality:

$$\sum_{i \in I_0(\bar{x})} u_i^+(\varepsilon) + \sum_{i \in I_1(\bar{x})} u_i^-(\varepsilon) \le \varepsilon. \tag{3.4}$$

Inequality (3.4) can also be represented as $y^T A u(\varepsilon) \le \varepsilon$ where the characteristic vector y is given by:

$$y_i = \begin{cases} 1, & (i \in I_0(\bar{x}) \text{ and } u_i(\varepsilon) > 0) \text{ or } (i \in I_1(\bar{x}) \text{ and } u_i(\varepsilon) < 0), \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

Hence the determination of the characteristic vector of a violated inequality can be done according to (3.5) once a value ε such that $u(\varepsilon) \notin N_\varepsilon([0,1]^n; \bar{x})$ has been found. Note that

different values of ε can give different characteristic vectors, since the sign of $u(\varepsilon)$ can be affected.

4. Error estimate

In the previous section we described an algorithm to verify the inclusion of $\partial_\varepsilon g(\bar{x})$ in $N_\varepsilon([0,1]^n;\bar{x})$ for each ε≥0. In the worst case the algorithm could consider all the faces of $N_\varepsilon([0,1]^n;\bar{x})$ which are 2^n-1. Hence, in principle, the algorithm converges finitely, even though it can do an exponential number of iterations. However, at any moment during the execution of the algorithm we are able to give a measure of the approximation of the solution under consideration.

Consider a parametric polyhedron $T(\varepsilon)$ obtained during the computation of **Opt**$(\bar{x})$, and let $W(T(\varepsilon)) = \{v(\varepsilon) \in V(T(\varepsilon)): \exists\, \varepsilon' \in (L(\bar{x}),U(\bar{x}))\ v(\varepsilon') \notin N_\varepsilon([0,1]^n;\bar{x})\}$.

In general, for any $v(\varepsilon) \in W(T(\varepsilon))$, due to the form of $f(v(\varepsilon))$ we have that $v(\varepsilon) \notin N_\varepsilon([0,1]^n;\bar{x})$ (i.e. $f(v(\varepsilon))>0$) for ε in the intervals $(\varepsilon'_1,\varepsilon''_1)$, $(\varepsilon'_2,\varepsilon''_2),\ldots,(\varepsilon'_k,\varepsilon''_k)$,where the number of intervals k is limited by n. Hereafter, such intervals will be called the *violation intervals* of $v(\varepsilon)$.

Lemma 4.1

Let $v(\varepsilon) = t\sqrt{\varepsilon} + \bar{q}$ be a vertex of $W(T(\varepsilon))$. Function $f'(v(\varepsilon)) = f(v(\varepsilon))+\varepsilon$ is monotonically non decreasing with respect to ε.

Proof:

We can assume that $\bar{q}_i < 0$ for each $i \in I_0(\bar{x})$ and $\bar{q}_i > 0$ for each $i \in I_1(\bar{x})$, otherwise we could immediately state that $\bar{x}$ is not optimal (see Section 2). Observe that

$$f'(v(\varepsilon)) = \sum_{i \in I_0(\bar{x})} v_i^+(\varepsilon) + \sum_{i \in I_1(\bar{x})} v_i^-(\varepsilon)$$

is a piecewise linear function of $\sqrt{\varepsilon}$, and, from the definition of $v_i^+(\varepsilon)$ and $v_i^-(\varepsilon)$, the coefficient of $\sqrt{\varepsilon}$ is non negative for each value of ε. In fact, if $i \in I_0(\bar{x})$, $[t_i\sqrt{\varepsilon} + \bar{q}_i\,]^+ = \max\{0, t_i\sqrt{\varepsilon} + \bar{q}_i\}$ is equal to zero for each $\varepsilon \geq 0$ when $t_i \leq 0$. On the other end, if $i \in I_1(\bar{x})$, $[t_i\sqrt{\varepsilon} + \bar{q}_i\,]^- = \max\{0, -t_i\sqrt{\varepsilon} - \bar{q}_i\}$ is equal to zero for each ε≥0 when $t_i \geq 0$.

As $f'(v(\varepsilon))$ is continuous, the thesis follows. ❑

Consider $v(\varepsilon) \in W(T(\varepsilon))$, and its violation intervals $(\varepsilon'_1,\varepsilon''_1)$, $(\varepsilon'_2,\varepsilon''_2),\ldots,(\varepsilon'_k,\varepsilon''_k)$. Let us introduce the following notation:

$\alpha(v) = \max\{(\varepsilon''_j - \varepsilon'_j), j=1,\ldots,k\}$

$\beta(v) = \max\{f'(v(\varepsilon)): \varepsilon \in (L(\bar{x}),U(\bar{x}))\}$

$\gamma(v) = \min\{\alpha(v), \beta(v)\}$.

Lemma 4.2

Let $v(\varepsilon)$ be a vertex of $W(T(\varepsilon))$, then $v(\varepsilon) \in N_{\varepsilon+\alpha(v)}([0,1]^n; \bar{x})$ for each $\varepsilon>0$.

Proof:

For any violation interval $(\varepsilon',\varepsilon'')$ of $v(\varepsilon)$, we know that $v(\varepsilon') \in N_{\varepsilon'}([0,1]^n; \bar{x})$ and $v(\varepsilon'') \in N_{\varepsilon''}([0,1]^n; \bar{x})$, hence $f(v(\varepsilon')) \le 0$ and $f(v(\varepsilon'')) \le 0$. Consider $\varepsilon \in (\varepsilon',\varepsilon'')$, we have:

$$f(v(\varepsilon)) + \varepsilon \le f(v(\varepsilon'')) + \varepsilon'' \le \varepsilon'' = \varepsilon + \varepsilon'' - \varepsilon \le \varepsilon + \varepsilon'' - \varepsilon'. \qquad (4.1)$$

The first inequality is due to the fact that $f'(v(\varepsilon))$ is monotone and the second from the non positivity of $f(v(\varepsilon''))$.

(4.1) implies that $v(\varepsilon) \in N_{\varepsilon+\varepsilon''-\varepsilon'}([0,1]^n; \bar{x})$ for each $\varepsilon \in (\varepsilon',\varepsilon'')$. This result can be extended to all the violation intervals of $v(\varepsilon)$, and from the definition of $\alpha(v)$ the thesis follows. ❑

Theorem 4.3

For any $T(\varepsilon)$ obtained during the computation of **Opt**$(\bar{x})$, $\bar{x}$ is a δ-optimal solution, where $\delta=\max\{\gamma(v): v(\varepsilon) \in W(T(\varepsilon))\}$.

Proof:

Consider $v(\varepsilon) \in W(T(\varepsilon))$, note that $v(\varepsilon) \in N_{\varepsilon+\alpha(v)}([0,1]^n; \bar{x})$ for each $\varepsilon \in (L(\bar{x}),U(\bar{x}))$ according to Lemma 4.2, hence, if α' denotes $\max\{\alpha(v): v(\varepsilon) \in W(T(\varepsilon))\}$, we have

$$\partial_\varepsilon g(\bar{x}) \subset N_{\varepsilon+\alpha'}([0,1]^n; \bar{x}) \qquad \forall \varepsilon>0.$$

On the other hand, $f'(v(\varepsilon)) \le \beta(v)$ for each $\varepsilon \in (L(\bar{x}),U(\bar{x}))$, which means that $v(\varepsilon) \in N_{\varepsilon+\beta(v)}([0,1]^n; \bar{x})$ for each $\varepsilon \in (L(\bar{x}),U(\bar{x}))$. Let us define $\beta'=\max\{\beta(v): v(\varepsilon) \in W(T(\varepsilon))\}$; β' is an upper bound to the Hausdorff distance between $\partial_\varepsilon g(\bar{x})$ and $N_\varepsilon([0,1]^n; \bar{x})$ when $\partial_\varepsilon g(\bar{x}) \not\subset N_\varepsilon([0,1]^n; \bar{x})$, as $T(\varepsilon)$ is an outer approximation of $\partial_\varepsilon g(\bar{x})$. Since $\partial_\varepsilon g(\bar{x}) \subset N_\varepsilon([0,1]^n; \bar{x})$ for $\varepsilon \le L(\bar{x})$ and $N_\varepsilon([0,1]^n; \bar{x})$ expands faster than $\partial_\varepsilon g(\bar{x})$ when $\varepsilon > U(\bar{x})$ we have that

$$\partial_\varepsilon g(\bar{x}) \subset N_{\varepsilon+\beta'}([0,1]^n; \bar{x}) \qquad \forall \varepsilon>0.$$

Hence the thesis follows. ❑

5. Implementational aspects

Lemma 4.1 suggests some strategies to implement procedure **Opt**($\bar{x}$) in order to improve the convergence of the approximation measure of the solution at each iteration. In particular, the selection of $v(\varepsilon) \notin N_\varepsilon([0,1]^n; \bar{x})$ for some ε, can be done by considering

$$\operatorname{argmax}\{\gamma(v(\varepsilon)): v(\varepsilon) \in W(T(\varepsilon))\}.$$

Moreover, suppose that $v(\varepsilon)$ has been chosen, and that $f(v(\varepsilon)) > 0$ for ε having value in the intervals $(\varepsilon'_1,\varepsilon''_1)$, $(\varepsilon'_2,\varepsilon''_2),\ldots,(\varepsilon'_k,\varepsilon''_k)$, $1 \le k \le n$; the selection of the halfspace to intersect with $T(\varepsilon)$ can be done by considering the largest interval where $f(v(\varepsilon)) > 0$, or the interval where $f(v(\varepsilon))$ attains the maximum. This strategy ensures the fact that the error estimate decreases at each iteration. It can be efficiently implemented by storing the vertices in $W(T(\varepsilon))$ into a heap sorted by $\gamma(v(\varepsilon))$.

There are at least two ways to represent the parametric polyhedron $T(\varepsilon)$. The first representation maintains a list of the vertices $v(\varepsilon) \in V(T(\varepsilon))$; in addition, for each vertex, a list of pointers to the adjacent vertices has to be saved and updated during the algorithm. This approach, which can be very efficient for fixed polyhedra, in our case can present some difficulties as $T(\varepsilon)$ is parametric, and is also modified during the algorithm when new faces are added, thus the adjacency lists of the vertices are modified.

Another way to represent the polyhedron consists of considering the system of parametric inequalities that defines it, that is:

$$Du(\varepsilon) \le d\sqrt{\varepsilon},$$

and to obtain the vertices $v(\varepsilon) \in V(T(\varepsilon))$ by enumerating the set of feasible basis of D (see Section 4). In addition we explicitly maintain the list of vertices $v(\varepsilon) \in W(T(\varepsilon))$ which do not belong to $N_\varepsilon([0,1]^n; \bar{x})$ for some value of ε, and the corresponding sets of intervals where $v(\varepsilon) \notin N_\varepsilon([0,1]^n; \bar{x})$.

Given this kind of representation, at each step of the algorithm we have to enumerate all the vertices of $T(\varepsilon)$ lying on the hyperplane that we add. This can be accomplished by means of the algorithm proposed by Avis and Fukuda [1], which enumerates the vertices of a given polyhedron in linear space in the number of constraints describing the polyhedron. Given a starting basis B, which corresponds to a vertex of $T(\varepsilon)$ on the hyperplane we are interested in, the algorithm sets up an objective function such that B is the unique optimal basis. The main idea of the enumerating algorithm consists of considering a pivot selection rule for the simplex algorithm and enumerate in reverse order all the sequences of feasible basis which lead to the optimal solution. This enumeration is done in a depth first manner, to

use as little storage as possible. The enumerating algorithm is summarized in Table 2 by the recursive procedure **Vertex_enumeration**; N denotes the set of non basic indices and **Pivot_rule**(B) give the basis B' that is obtained starting from B. The first time the procedure is called, B is a basis corresponding to a vertex of the considered face and **Set_of_basis** is empty. For further details refer to [1, 20].

```
Procedure Vertex_enumeration(B,N,Set_of_basis);
Begin
  Set_of_basis:=Set_of_basis∪{B};
  For each j∈N do
  Begin
    Let h∈B such that Rank(B∪{j}\{h}) = N;
    B'=B∪{j}\{h};
    If Pivot_rule(B')=B Then Vertex_enumeration(B',{1,…,m}\B',Set_of_basis);
  End
End.
```

Table 2

6. Computational results

This section presents some computational results of the testing algorithm. In a first set of experiments we generated relatively small test problems (n=15) in order to be able to enumerate and test all the solutions in a reasonable time. We applied the algorithm to all 32,768 0-1 vectors.

Elements of matrix Q, were generated with a uniform distribution in the range $[-\alpha, \alpha]$, while elements of vector c were generated with a uniform distribution in the range $[-\beta,\beta]$. Quadratic problems with these characteristics have the property that the number of local optima grows exponentially with n [12]. Thus these problems are usually considered to be difficult. The range of the linear coefficients is chosen in order to have different percentage of "diagonal dominance" [3]. We have p% diagonal dominance if $\beta=(p/100)\alpha(n-1)$. It should be clear that when diagonal dominance is high the problem is easier, as stated also in [14].

We generated two sets of ten problems with α=50 and β=280 (p=40%) and α=50, β=140 (p=20%), which are reported in Table 3 and Table 4, respectively. After being generated, the test problems are scaled in order to make the quadratic matrix strictly positive definite (see Section 2). For practical reasons, we had to put an upper bound on the size of set $W(T(\varepsilon))$ used during the execution of the algorithm; in our experiments $W(T(\varepsilon))$ is allowed to contain at most 5,000 elements. In the tables we distinguished the cases when the algorithm recognized the optimality or the non optimality of the tested solutions, and the cases when the algorithm exceeded the maximum number of vertices. In the case of optimal solutions, we indicated the number of times an optimal solution is recognized and the number of vertices that have been generated to verify the optimality. For the non optimal solutions we

have two cases: in the first case (case 1) the non optimality is recognized before entering the while of procedure **Opt**($\bar{x}$), simply by checking the initial necessary conditions, that is $L(\bar{x})$ is not greater than zero, and not all the tangent points of the initial $T(\varepsilon)$ belong to $N_\varepsilon([0,1]^n; \bar{x})$; in the second case (case 2) the non optimality is recognized during the while of procedure **Opt**($\bar{x}$), in this case for each tested solution the number of vertices generated is reported. In the case of an undefined answer, that is when the algorithm was not able to say whether a solution is optimal or not, we reported the number of times it occurred, and for each occurrence we reported the number of generated hyperplanes and the relative error of the solution which was estimated when the algorithm stopped. The relative error is computed as $\max\{\gamma(v)\colon v(\varepsilon) \in W(T(\varepsilon))\}$ divided by the range of the objective function (i.e. the difference between the maximum and the minimum value). We evaluated the relative error with this method to avoid the influence of optimal solutions with zero value or almost zero value. The number of hyperplanes generated includes the initial ones (i.e. n) used to define the initial $T(\varepsilon)$.

	optimal		**non optimal**			**undefined**		
problem	no. sol.	no. vertices	case 1	case 2	no. vertices	no. sol.	no. hyperplanes	error %
P1.1	1	722	32766	1	1	0	-	-
P1.2	-	-	32766	1	17	1	56	19.91
P1.3	1	281	32767	0	0	0	-	-
P1.4	1	419	32767	0	0	0	-	-
P1.5	-	-	32766	1	4	1	46	15.47
P1.6	-	-	32767	0	0	1	42	8.08
P1.7	-	-	32767	0	0	1	56	0.94
P1.8	1	2135	32767	0	0	0	-	-
P1.9	-	-	32767	0	0	1	52	1.39
P1.10	-	-	32767	0	0	1	44	2.57

Table 3

	optimal		**non optimal**			**undefined**		
problem	no. sol.	no. vertices	case 1	case 2	no. vertices	no. sol.	no. hyperplanes	error %
P2.1	-	-	32766	1	3	1	50	28.58
P2.2	-	-	32766	1	1	1	53	31.28
P2.3	-	-	32764	2	2 1	2	45 39	55.18 49.87
P2.4	-	-	32767	0	-	1	43	16.98
P2.5	-	-	32766	0	-	2	49 51	14.81 24.75
P2.6	-	-	32765	2	2 186	1	43	20.08
P2.7	-	-	32767	0	0	1	44	18.31

P2.8	-	-	32764	3	613	1	47	21.12
P2.9	-	-	32767	0	0	1	49	25.76
P2.10	-	-	32767	0	0	1	48	11.71

Table 4

From the results in Tables 3 and 4, the testing algorithm turns out to be extremely efficient in recognizing non optimal solutions. For 40% diagonal dominance problems (Table 3) all the non optimal solutions are detected either during the initialization phase (in almost all cases) or after a few iterations of the algorithm. The optimal solution is recognized in four cases, and except for one problem (P1.8) the number of tested vertices is not excessively high. For the undefined cases, the estimate of the error is variable; in some cases it is very significant (P1.7, P1.9 and P1.10). For 20% diagonal dominance problems (Table 4), in nine cases the algorithm recognizes all the non optimal solutions very efficiently. However the algorithm is never able to certify the optimality, and the estimate of the error is rather high.

We carried out a second set of experiments on larger problems. We intended to analyze the efficiency of the algorithm and see how the estimate of the relative error decreases with the number of iterations, testing the optimal solution of quadratic problems with size equal to 100. In order to know the optimal solution, we generated the problems with negative linear coefficients and positive quadratic coefficients. This way [18], we can find the solution efficiently by means of a simple min-cut max-flow algorithm. Elements of matrix Q were generated with a uniform distribution in the range [1, 31] and q_{ii}=0, while elements of vector c were generated with a uniform distribution in the range [-1501,-1]. Table 5 presents a set of 10 problems. For each problem we report the value of the optimal solution (opt), the initial estimate of the relative error (100 Hp.), and the estimate of the relative error after the addition of 10, 20 and 30 hyperplanes (110 Hp., 120 Hp. 130 Hp.). Since we could not evaluate all the solutions, we had to give a lower approximation of the range, to obtain an estimate of the relative error. A simple local search has been used to produce a heuristic solution of min $\{1/2x^TQx + c^Tx: x \in \{0,1\}^n\}$.

problem	**opt.**	**100 Hp.**	**110 Hp.**	**120 Hp.**	**130 Hp.**
P3.1	489	24.85	22.03	21.01	20.74
P3.2	596	18.88	15.84	14.78	13.89
P3.3	0	7.98	5.84	5.18	4.96
P3.4	10191	6.29	4.76	4.37	4.22
P3.5	11072	9.53	7.10	6.69	6.02
P3.6	7846	11.74	9.81	9.05	8.50
P3.7	180	24.71	22.55	21.54	20.81

P3.8	1167	19.32	16.47	16.13	15.17
P3.9	4677	25.64	23.67	22.27	21.86
P3.10	115	27.49	25.16	24.05	22.99

Table 5

After the addition of 130 hyperplanes the number of generated vertices varies from a minimum of 2607 to more than 40000, and on average it is about 18700.

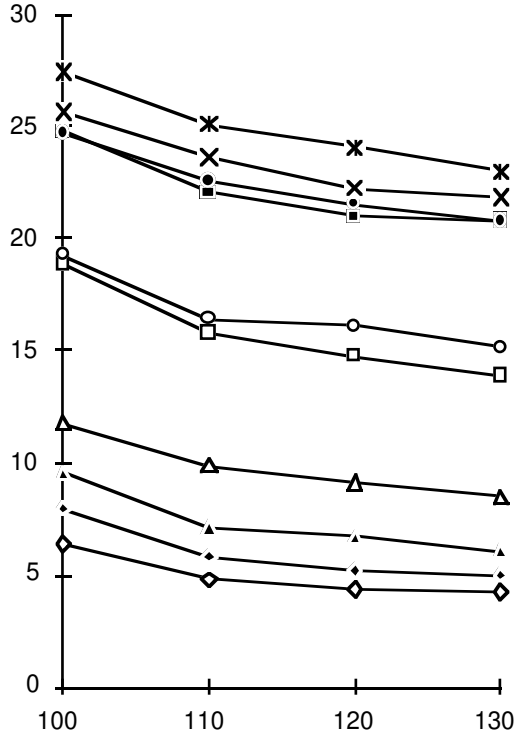

Figure 1

In Figure 1 we report the decrease of the relative error when the number of hyperplanes increases, for all the test problems of Table 5. In general we can observe that the relative error does not decrease very significantly with the number of iterations, and the decreease is independent of the initial error. In some cases (problems P3.3, P3.4 and P3.5) the initial error is relatively small and at the end of the algorithm we obtain a reasonable estimate.

To give an idea of the computational time we report the results for some of the above instances. The code has been implemented in C on a workstation HP 9000/712, and uses CPLEX routines for the vertex enumeration. In all the cases of recognized non optimal solution, the answer is almost immediate and the time is of the order of few hundreds of a second both in the case when we enter the inclusion routine or we stop before entering it. The optimal solution for instances P1.1, P1.3, P1.4 and P1.8 is recognized in 3.54, 1.37, 1.95 and 13.19 seconds, respectively. We also report the computational times in seconds for some of the larger instances when the number of hyperplanes increases (see Table 6).

problem	100 Hp.	110 Hp.	120 Hp.	130 Hp.
P3.1	0.08	2.73	136.87	1174.87
P3.4	0.05	15.97	386.44	1222.10
P3.10	0.07	2.56	31.85	67.66

Table 6

Table 6 places the exponential behavior of the algorithm in evidence.

7. Extension to constrained problems

Consider a general constrained quadratic 0-1 problem:

$$\max \{\tfrac{1}{2}x^TQx + c^Tx: x \in F \cap \{0,1\}^n\}, \qquad (7.1)$$

where Q is an $n \times n$ positive definite symmetric matrix, $c \in \boldsymbol{R}^n$ and $F \subseteq \boldsymbol{R}^n$ is a convex polyhedron. Classical examples of this kind of problem are Quadratic Assignment, Quadratic Semi-Assignment and Quadratic Knapsack.

As we did in the unconstrained case, also problem (7.1) can be transformed into an equivalent continuous problem. One possible transformation is that defined in [5], and in practice consists of scaling of the objective function coefficients; the equivalent problem is

$$\max \{\tfrac{1}{2}x^T(2\mu I_n+Q)x + (c+\mu e)^Tx: x \in F \cap [0,1]^n\},$$

where μ is a suitable non negative value, and I_n denotes the identity matrix of order n. The optimal solution will be on an integer extreme point of $F \cap [0,1]^n$.

Now let us briefly see the definitions of $\partial_\varepsilon g(\bar{x})$ and $N_\varepsilon(S; \bar{x})$ for $g(x)=\frac{1}{2}x^T(2\mu I^n+Q)x + (c+\mu e)^Tx$ and $S=F \cap [0,1]^n$. Since $g(x)$ is still a convex quadratic function, $\partial_\varepsilon g(\bar{x})$ is an elliptic set whose expression is similar to that of the unconstrained case; conversely, the definition of $N_\varepsilon(S; \bar{x})$ has to take into account the additional constraints, hence we have:

$$N_\varepsilon(S; \bar{x}) = \{u \in \boldsymbol{R}^n: u^T(x - \bar{x}) \leq \varepsilon, \forall x \in S\}.$$

Consider V_S the set of all the extreme points of S, $u^T(x-\bar{x}) \leq \varepsilon$, $\forall x \in S$ if and only if $u^T(x-\bar{x}) \leq \varepsilon \ \forall x \in V_S$. Thus the facets of $N_\varepsilon(S;\bar{x})$ can be described as $\{u: u^T(x-\bar{x}) = \varepsilon, \forall x \in V_S$. We can construct a parametric polyhedron $T(\varepsilon)$ approximating $\partial_\varepsilon g(\bar{x})$ as we did in the unconstrained case, where all the vertices of $T(\varepsilon)$ and all the tangent points between $T(\varepsilon)$ and $\partial_\varepsilon g(\bar{x})$ for each value of ε are of the form $u(\varepsilon) = t\sqrt{\varepsilon} + \bar{q}$. To verify whether $u(\varepsilon) \in N_\varepsilon(S; \bar{x})$ means that $u(\varepsilon)^T(x-\bar{x}) \leq \varepsilon$, $\forall x \in S$, hence we have to check if the function

$$\Phi(\varepsilon) = \max \{u(\varepsilon)^T(x-\bar{x}), \forall x \in S\} - \varepsilon \qquad (7.2)$$

assumes only non positive values, for each $\varepsilon>0$. This requires obtaining the set of optimal solutions of the parametric maximization problem in (7.2) when ε varies. Obviously this approach is reasonable only if it can be done efficiently, as for example, in the case of the Quadratic Knapsack, where $S=\{x\in[0,1]^n: a^Tx\leq b\}$, whilst probably this approach is not very efficient for more complex problems such as the Quadratic Assignment. In fact in this case all the solutions of a parametric linear assignment problem have to be found each time we have to check whether a parametric point $u(\varepsilon)\in N_\varepsilon(S;\bar{x})$.

Another interesting extension of the proposed testing approach is, when a non optimal solution is recognized, to find some helpful indications to modify it and move towards a better solution.

Acknowledgment

The two latter authors are indebted to Massimo Pappalardo for having introduced them to the study of global optimality and for helpful discussions. They are also grateful to the anonymous referees for the promptness and the efficacy of their comments and suggestions.

References

[1] Avis, D. and K. Fukuda, *A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra.* **Discrete Applied Mathematics**, 1991. to appear.

[2] Carraresi, P., F. Malucelli and M. Pappalardo, *Testing optimality for quadratic 0-1 unconstrained problems.* **ZOR - Mathematical Methods of Operations Research**, 1995. 42: p. 295-311.

[3] Carter, M. W., *The indefinite zero-one quadratic problem.* **Discrete Applied Mathematics**, 1984. 7: p. 23-44.

[4] Gallo, G., P. Hammer and B. Simeone, *Quadratic knapsack problems.* **Mathematical Programming**, 1980. 12: p. 132-149.

[5] Giannessi, F. and F. Nicolucci, *Connections between nonlinear and integer programming problems,* in *Symposia Mathematica*. 1976, Academic Press.

[6] Hammer, P. L., *Boolean Elements in Combinatorial Optimization.* **Annals of Discrete Mathematics**, 1979. 4: p. 51-71.

[7] Hammer, P. L., P. Hansen and B. Simeone, *Roof Duality, Complementation and Persistency in Quadratic 0-1 Optimization.* **Mathematical Programming**, 1984. (28): p. 121-155.

[8] Hansen, P., *Methods of nonlinear 0-1 programming.* **Annals of Discrete Mathematics**, 1979. 5: p. 53-69.

[9] Hirriart-Urruty, J.-B., *Conditions for Global Optimality,* in *Handbook of Global Optimization,* R. Horst and P.M. Pardalos, Editors. 1995, Kluwer Academic Publishers: p. 1-26.

[10] Hirriart-Urruty, J. B., *From convex optimization to nonconvex optimization. Part I: Necessary and sufficient conditions for global optimality,* in *Nonsmooth Optimization and Related Topics*. 1989, Plenum Press: p. 219-239.

[11] Hirriart-Urruty, J. B. and C. Lemarechal, *Testing necessary and sufficient conditions for global optimality in the problem of maximizing a convex quadratic function over a convex polyhedron.*, (1990) University Paul Sabatier, Toulouse.

[12] Long, J. and A. C. Williams, *On the number of local maxima in quadratic 0-1 programs.* **Operations Research Letters**, 1993. 13: p. 75-78.

[13] Malucelli, F., *Quadratic Assignment Problems: Solution Methods and Applications*, TD-9 (1993) University of Pisa - Dept. of Computer Science.

[14] Pardalos, P. M., *Construction of test problems in quadratic bivalent programming.* **ACM Transactions on Mathematical Software**, 1991. 17(1): p. 74-87.

[15] Pardalos, P. M. and S. Jha, *Complexity of uniqueness and local search in quadratic 0-1 programming.* **Operations Research Letters**, 1992. 11: p. 119-123.

[16] Pardalos, P. M., F. Rendl and H. Wolkowicz, *The quadratic assignment problem: a survey and recent developments,* in *Quadratic Assignment and Related Problems,* P.M. Pardalos and H. Wolkowicz, Editors. 1994, American Mathematical Society: p. 1-42.

[17] Pardalos, P. M. and G. P. Rodgers, *Computational aspects of a branch and bound algorithm for quadratic zero-one programming.* **Computing**, 1990. 45: p. 131-144.

[18] Picard, J. C. and H. D. Ratliff, *Minimum cuts and related problems.* **Networks**, 1974. 5: p. 357-370.

[19] Rosenberg, I. G., *Reduction of Bivalent Maximization to the Quadratic Case.* **Cahiers du Centre d'Etudes de Recherche Opérationelle**, 1975. (17): p. 71-79.

[20] Rote, G., *Degenerate convex hulls in high dimensions without extra storage*, (1993) Technische Universität Graz - Institut für Mathematik.

Nota da scartare eventualmente

The effectiveness of the algorithm can be improved by making some additional considerations. Assume to intersect $T(\varepsilon)$ with the halfspace $Ayu \leq h(\varepsilon)$ tangent to $\partial_\varepsilon g(\bar{x})$, $\forall \varepsilon \in (L(\bar{x}), U(\bar{x}))$. Consider now all the vertices $v^1(\varepsilon)$, $v^2(\varepsilon),\ldots,v^k(\varepsilon)$ of $T(\varepsilon)$ which layes on $Ayu=h(\varepsilon)$; if all these vertices belong to $N_\varepsilon([0,1]^n;\bar{x})$ then also the new face $Ayu=h(\varepsilon)$ belongs entirely to $N_\varepsilon([0,1]^n;\bar{x})$. This suggests to improve the effectiveness of the testing algorithm by considering a new $Ayu=h'(\varepsilon)$ which is no more tangent with $\partial_\varepsilon g(\bar{x})$, but actually intersects it. The linear function of $\sqrt{\varepsilon}h'(\varepsilon)$ is defined in such a way to guarantee that all the vertices of $T(\varepsilon)\cap\{Ayu=h'(\varepsilon)\}$ are in $N_\varepsilon([0,1]^n;\bar{x})$. Notice that $T(\varepsilon)\cap\{Ayu\leq h'(\varepsilon)\}$ is no more an outer approximation of $\partial_\varepsilon g(\bar{x})$, but we know that the portion of $\partial_\varepsilon g(\bar{x})$ whcih does not belong to $T(\varepsilon)\cap\{Ayu\leq h'(\varepsilon)\}$ is surely contained in $N_\varepsilon([0,1]^n;\bar{x})$. The computation of the vertices of $N_\varepsilon([0,1]^n;\bar{x})$ in general can be difficult as the intersections between $T(\varepsilon)$ and $\{Ayu\leq h'(\varepsilon)\}$ are not easy to find when $h'(\varepsilon)$ changes. Hence we can think about a simplified approach, where instead of finding the vertices of $T(\varepsilon)\cap\{Ayu=h'(\varepsilon)\}$, we consider the vertices laying on the edges of $T(\varepsilon)\cap\{Ayu\leq h(\varepsilon)\}$ adjacent to $v^1(\varepsilon)$, $v^2(\varepsilon),\ldots,v^k(\varepsilon)$. The new vertices might not be the vertices of $T(\varepsilon)\cap\{Ayu=h'(\varepsilon)\}$, but they constitute an valid approximation.