A Fast Branch-and-Bound Algorithm for Non-convex Quadratic Integer Optimization Subject To Linear Constraints Using Ellipsoidal Relaxations☆

Christoph Buchheim[a], Marianna De Santis[a], Laura Palagi[b]

[a] *Fakultät für Mathematik, Technische Universität Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany*
[b] *Dipartimento di Ingegneria informatica automatica e gestionale A. Ruberti, Sapienza Università di Roma, Via Ariosto, 25, 00185 Roma, Italy*

Abstract

We propose two exact approaches for non-convex quadratic integer minimization subject to linear constraints where lower bounds are computed by considering ellipsoidal relaxations of the feasible set. In the first approach, we intersect the ellipsoids with the feasible linear subspace. In the second approach we penalize exactly the linear constraints. We investigate the connection between both approaches theoretically. Experimental results show that the penalty approach significantly outperforms CPLEX on problems with small or medium size variable domains.

Keywords: integer programming, quadratic programming, global optimization
2010 MSC: 90C10, 90C20, 90C57

1. Introduction

We address quadratic integer optimization problems with box constraints and linear equality constraints,

$$\begin{array}{rl} \min & q(x) = x^\top Q x + c^\top x \\ \text{s.t.} & Ax = b \\ & l \le x \le u \\ & x \in \mathbb{Z}^n \ , \end{array} \tag{1}$$

where $Q \in \mathbb{R}^{n\times n}$ is assumed to be symmetric but not necessarily positive semidefinite, $c \in \mathbb{R}^n$, and w.l.o.g. $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$. Moreover, we may assume $l < u$ and $l, u \in \mathbb{Z}^n$. Problems of this type arise, e.g., in quadratic min cost flow problems, where the linear equations model flow conservation, $l = 0$ and u represents edge capacities. Note that we can also handle linear inequalities in (1) by simply introducing slack variables.

Problems of type (1) are very hard to solve in theory and in practice. In general, the problem is NP-hard both due to the integrality constraints and due to the non-convexity of the objective function. Few exact algorithms have been proposed in the literature so far, most of them based on either linearization or convexification [1, 10] or on SDP-relaxations [3].

For the variant of (1) containing only box constraints, but no other linear constraints, Buchheim et al. [2] recently proposed a branch-and-bound algorithm based on ellipsoidal relaxations of the feasible box. More precisely, a suitable ellipsoid E containing $[l, u]$ is determined and $q(x)$ is minimized over $x \in E$.

The latter problem is known as the trust region subproblem [4, 6, 8] and can be solved efficiently, thus yielding a lower bound in our context. Besides many other improvements, this branch-and-bound algorithm mostly relies on an intelligent preprocessing technique that allows to solve the dual of a trust region problem in each node of the enumeration tree in a very short time, making it possible to enumerate millions of nodes in less than a minute.

Our aim is to adapt this method to the presence of linear equations $Ax = b$. For this, we propose two different, but related approaches. In the first approach (see Section 2), we intersect the ellipsoid E with the subspace given by $Ax = b$. This leads, by considering an appropriate projection, to a trust region type problem that, in principle, can still be solved efficiently. In the second approach (see Section 3), we instead lift the constraints into the objective function by adding a penalty term $M\|Ax - b\|^2$ drawing inspiration from the augmented Lagrangians theory [7]. A finite and computable real value $\bar{M} > 0$ exists such that the resulting quadratic problem with only the simple constraints $[l, u] \cap \mathbb{Z}^n$ is equivalent to (1). Thus the branch-and-bound algorithm defined in [2] which uses an ellipsoidal relaxation E of the feasible set can be used in a straighforward way.

In Section 4, we show that the bound obtained from the penalty approach converges to the bound obtained in the projection approach when $M \to \infty$. We finally present the results of an experimental comparison of the penalty approach with CPLEX, see Section 5.

2. Projection approach

The first approach we propose for the computation of lower bounds for Problem (1) is based on the familiar partitioning of x into basic and non-basic variables x_B and x_N. Without loss of

☆The second author has been supported by the German Research Foundation (DFG) under grant BU 2313/4. The third author has been partially supported by the Italian project PLATINO (Grant Agreement n. PON01_01007).

Email addresses: christoph.buchheim@tu-dortmund.de (Christoph Buchheim), marianna.de.santis@math.tu-dortmund.de (Marianna De Santis), laura.palagi@uniroma1.it (Laura Palagi)

generality we assume that $\operatorname{rank}(A) = m$. Let B be a basis of A, then the equality constraint can be written as $Bx_B + Nx_N = b$; a similar idea is used, e.g., in [9]. This leads to a k-dimensional trust region type problem, where $k = n - m$ is the dimension of the kernel of the matrix A.

Let H be a positive definite matrix that, together with the center point x^0, defines an ellipsoid

$$E(H) = \{x \in \mathbb{R}^n \mid (x - x^0)^\top H(x - x^0) \leq 1\}$$

such that $[l, u] \subseteq E(H)$. Consider the following relaxation of our original problem (1)

$$\begin{array}{ll} \min & q(x) = x^\top Qx + c^\top x \\ \text{s.t.} & Ax = b \\ & x^\top Hx \leq 1 \end{array} \tag{2}$$

where w.l.o.g. we assumed $x^0 = 0$. We show that Problem (2) can be transformed into a trust-region type problem so that the branch-and-bound algorithm defined in [2] can be applied. Let us write vectors and matrices accordingly to the partition induced by B and N. Thus, the x_B variables can be eliminated via substituting

$$x_B = B^{-1}b - B^{-1}Nx_N$$

in (2). We obtain

$$\begin{array}{ll} \min & x_N^\top \widetilde{Q} x_N + \widehat{c}^\top x_N + d \\ \text{s.t.} & x_N^\top \widetilde{H} x_N + \widetilde{h}^\top x_N \leq 1 - b^\top B^{-\top} H_{BB} B^{-1} b \\ & x_N \in \mathbb{R}^k \ , \end{array} \tag{3}$$

where

$$\begin{aligned} &\widetilde{Q} = Q_{NN} + N^\top B^{-\top} Q_{BB} B^{-1} N - Q_{BN}^\top B^{-1} N - N^\top B^{-\top} Q_{BN} \\ &\widehat{c} = 2(-N^\top B^{-\top} Q_{BB} B^{-1} + Q_{NB} B^{-1})b - (B^{-1}N)^\top c_B + c_N \\ &d = b^\top B^{-\top} Q_{BB} B^{-1} b + c_B^\top B^{-1} b \\ &\widetilde{H} = H_{NN} + N^\top B^{-\top} H_{BB} B^{-1} N - H_{NB} B^{-1} N - N^\top B^{-\top} H_{BN} \\ &\widetilde{h} = 2\left(-N^\top B^{-\top} H_{BB} B^{-1} + H_{NB} B^{-1}\right) b \end{aligned}$$

Let

$$x_N^0 = -\frac{1}{2} \widetilde{H}^{-1} \widetilde{h}$$

be the center of the ellipsoidal constraint in problem (3). Then we can rewrite Problem (3) as

$$\begin{array}{ll} \min & x_N^\top \widetilde{Q} x_N + \widehat{c}^\top x_N + d \\ \text{s.t.} & (x_N - x_N^0)^\top \widetilde{H} (x_N - x_N^0) \leq \alpha \\ & x_N \in \mathbb{R}^k \ , \end{array}$$

where

$$\alpha = 1 - b^\top B^{-\top} H_{BB} B^{-1} b + \frac{1}{4} \widetilde{h}^\top \widetilde{H}^{-1} \widetilde{h}.$$

Next we consider the transformation $z = x_N - x_N^0$ and obtain the following problem which is exactly of the desired form

$$\begin{array}{ll} \min & z^\top \widetilde{Q} z + \widetilde{c}^\top z + \widetilde{d} \\ \text{s.t.} & z^\top \widetilde{H} z \leq \alpha \\ & z \in \mathbb{R}^k, \end{array} \tag{4}$$

where

$$\begin{array}{rcl} \widetilde{c} & = & 2\widetilde{Q} x_N^0 + \widehat{c}, \\ \widetilde{d} & = & \widehat{c}^\top x_N^0 + (x_N^0)^\top \widetilde{Q} x_N^0 + d. \end{array}$$

This approach can be embedded into the branch-and-bound procedure proposed in [2], where the enumeration strategy is depth-first and branching is done by fixing the value of the variables in a predetermined order. By the latter restriction, we ensure that the matrices $\widetilde{Q}$ and $\widetilde{H}$ only depend on the depth of the node in the branch-and-bound tree, i.e., on which variables have been fixed so far, but not on their values: First, a basis B^0 of A can be computed in the preprocessing phase. By always fixing non-basic variables, we get at each level $\ell = 0, \ldots, n - m$ that a basis of A^ℓ (the reduced matrix A indexed by only non-fixed variables) is $B^\ell = B^0$, whereas N^ℓ is obtained by simply removing columns from N^0. When all non-basic variables have been fixed to $\bar{x}_N$, then the corresponding node is a leaf and either $\bar{x}_B := (B^\ell)^{-1} b - B^{-1} N^\ell \bar{x}_N \in \mathbb{Z}^m$ or the node is infeasible.

Now the matrices $\widetilde{Q}$ and $\widetilde{H}$ in a given node of the branch-and-bound tree depend only on Q^ℓ, H^ℓ, B^ℓ, and N^ℓ, where Q^ℓ and H^ℓ denote the reduced matrices Q and H on level ℓ, which in turn depend only on the level ℓ and not on the specific variable fixings. In summary, $\widetilde{Q}$ and $\widetilde{H}$ only depend on the ordering of variables but not on the fixings of variables. This implies that only n different matrices $\widetilde{Q}^\ell$ and $\widetilde{H}^\ell$ appear in the entire branch-and-bound tree, so that, similarly to [2], all time-consuming calculations concerning $\widetilde{Q}^\ell$ and $\widetilde{H}^\ell$ can be performed in a preprocessing phase.

On the other hand, the construction of problem (4) at every node of the branch-and-bound tree requires the computation of $\widetilde{c}$, $\widetilde{d}$, $\widetilde{h}$, and α, which in turn depend on the values at which the variables have been fixed, as the right hand side term b is affected by the fixings. These vectors can however be updated quickly in an incremental fashion.

As a final remark, we want to point out that it is also possible to use the kernel representation of the equality constraints as $\{x \in \mathbb{R}^n \mid Ax = b\} = \{x \in \mathbb{R}^n \mid x = Vy + w, y \in \mathbb{R}^k\}$, where $V \in \mathbb{R}^{n \times k}$ is an orthonormal matrix defining a basis for $\ker(A)$ and $w \in \mathbb{R}^n$ is any vector satisfying $Aw = b$. By substituting x by $Vy + w$ in (2) and by further manipulations of the expressions, we get again a trust region type problem. The main difference between the two approaches consists in the computations needed to define the trust region relaxation at each node of the branch-and-bound tree, but the resulting bounds both agree with (2) and are hence the same.

3. Penalty approach

In the second approach we take inspiration from an old idea based on Lagrangian relaxation, in which the squared violation of linear constraints $\|Ax - b\|^2$ is lifted to the quadratic objective function (see [7] and references therein). Indeed Poljak et al. [7] prove the existence of a value $\bar{M}$ such that problem (1) is equivalent to

$$\min_{x \in X} \ q(x) + M\|Ax - b\|^2 \text{ for all } M \geq \bar{M}$$

whenever X is a finite set. In the following we explain how to obtain a finite value of $\bar{M}$ for which this equivalence holds.

More generally, consider the problem

$$\begin{array}{rl} q(x^*) = \min & q(x) = x^\top Q x + c^\top x \\ \text{s.t.} & x \in \mathcal{F} \cap X \end{array} \tag{5}$$

where $X \subseteq \mathbb{R}^n$ is again a finite set and $\mathcal{F} \subseteq \mathbb{R}^n$ is arbitrary. Let $d_{\mathcal{F}}(x) : \mathbb{R}^n \to \mathbb{R}_0^+$ be a function such that

$$\begin{array}{ll} d_{\mathcal{F}}(x) = 0 & \text{if } x \in \mathcal{F}; \\ d_{\mathcal{F}}(x) > 0 & \text{otherwise.} \end{array}$$

Since X is a finite set we can define

$$d_{\mathcal{F}}^{\min} = \min_{x \in X, x \notin \mathcal{F}} d_{\mathcal{F}}(x).$$

We notice that $d_{\mathcal{F}}^{min} > 0$ by definition of $d_{\mathcal{F}}(x)$.

Let $M > 0$ and $f : \mathbb{R}^n \to \mathbb{R}$ be the following penalty function

$$f(x) = q(x) + M\left(\frac{d_{\mathcal{F}}(x)}{d_{\mathcal{F}}^{\min}}\right)^2.$$

Let us consider the problem

$$\begin{array}{rl} \min & f(x) \\ \text{s.t.} & x \in X. \end{array} \tag{6}$$

We first note that a solution of problem (6) provides a lower bound for problem (5) for every $M > 0$. Indeed we have

$$\min_{x \in \mathcal{F} \cap X} q(x) = \min_{x \in \mathcal{F} \cap X} f(x) \geq \min_{x \in X} f(x).$$

Now let lb be any lower bound for problem (5), e.g., obtained by minimizing $q(x)$ over any relaxation C of $\mathcal{F} \cap X$. Moreover, let ub be any upper bound for problem (5), e.g., obtained as $ub = q(\hat{x})$ for some $\hat{x} \in \mathcal{F} \cap X$. Then obviously

$$lb \leq q(x^*) \leq ub.$$

We show that for $M > ub - lb \geq 0$ problems (5) and (6) are equivalent.

Theorem 1. *Let $M > ub - lb \geq 0$. Then problems* (5) *and* (6) *have the same minimizers.*

Proof. First we observe that by definition of $d_{\mathcal{F}}^{\min}$ we have

$$\frac{d(x, \mathcal{F})}{d_{\mathcal{F}}^{\min}} \geq 1 \qquad \forall\, x \in X,\ x \notin \mathcal{F}$$

so that

$$f(x) \geq q(x) + M \qquad \forall\, x \in X,\ x \notin \mathcal{F}.$$

Further, by definition, $d(x, \mathcal{F}) = 0$ for all $x \in \mathcal{F}$ so that

$$f(x) = q(x) \qquad \forall\, x \in \mathcal{F}.$$

We first prove that if x^* is a solution of (5) then x^* is a solution of (6). Let x^* be a solution of (5), then we have

$$q(x^*) \leq q(x) \quad \forall x \in \mathcal{F} \cap X. \tag{7}$$

Suppose by contradiction that there exists a point $\bar{x} \in X$ such that

$$f(\bar{x}) < f(x^*) = q(x^*). \tag{8}$$

We consider two cases:

i. Suppose that $\bar{x} \in \mathcal{F}$, that is $d(\bar{x}, \mathcal{F}) = 0$. Then, from (7)

$$f(\bar{x}) = q(\bar{x}) \geq q(x^*)$$

which contradicts (8).

ii. Now assume that $\bar{x} \notin \mathcal{F}$, that is $d(\bar{x}, \mathcal{F}) > 0$.
Since $q(\bar{x}) \geq lb$ and $\frac{d(\bar{x}, \mathcal{F})}{d_{\mathcal{F}}^{\min}} \geq 1$, we have

$$\begin{array}{rl} f(\bar{x}) & = q(\bar{x}) + M\left(\frac{d(\bar{x}, \mathcal{F})}{d_{\mathcal{F}}^{\min}}\right)^2 \geq \\ & \geq lb + M > lb + ub - lb = ub \\ & \geq q(x^*) \end{array} \tag{9}$$

which again contradicts (8).

We now prove that if $\bar{x}$ is a solution of (6) then it is a solution of (5). As a solution of (6) we have that $\bar{x} \in X$. We first prove that $\bar{x} \in \mathcal{F} \cap X$. By contradiction, we suppose that $\bar{x} \notin \mathcal{F}$, then, as in (9), we have

$$f(\bar{x}) = q(\bar{x}) + M\left(\frac{d(\bar{x}, \mathcal{F})}{d_{\mathcal{F}}^{\min}}\right)^2 > ub = q(\hat{x}),$$

where $\hat{x} \in \mathcal{F} \cap X$. This contradicts the fact that $\bar{x}$ is a solution for (6).

Now, suppose by contradiction that there exists a solution $x^* \in \mathcal{F} \cap X$ of problem (5) such that

$$q(x^*) < f(\bar{x}) = q(\bar{x}). \tag{10}$$

Since $x^* \in \mathcal{F} \cap X$ we have that $f(x^*) = q(x^*)$, then (10) implies that $f(x^*) < f(\bar{x})$, which contradicts the fact that $\bar{x}$ is an minimizer for problem (6). □

As an immediate consequence of the preceding theorem, we get the following equivalence result.

Corollary 1. *Let $M > ub - lb \geq 0$ and $p : \mathbb{R}^n \to \mathbb{R}$ be a function such that*

$$p(x) \geq \left(\frac{d(x, \mathcal{F})}{d_{\mathcal{F}}^{\min}}\right)^2$$

for all $x \in \mathbb{R}^n$. Then, problem (5) *has the same minimizers as*

$$\begin{array}{rl} \min & q(x) + Mp(x) \\ s.t. & x \in X. \end{array}$$

Going back to Problem (1), we have that

$$X = [l, u] \cap \mathbb{Z}^n, \ \ \mathcal{F} = \{x \in \mathbb{R}^n : Ax = b\}$$

and A, b have integer entries. Our set $\mathcal{F}$ is closed and convex, therefore it is possible to choose $d_{\mathcal{F}}(x)$ as

$$d_{\mathcal{F}}(x) = \min_{y \in \mathcal{F}} d(x, y),$$

with $d(x, y)$ being defined as $d(x, y) = \|A(x - y)\|$ so that

$$d_{\mathcal{F}}(x) = \|Ax - b\|.$$

With this choice we have that

$$m := d_{\mathcal{F}}^{min} = \min_{x \in X, Ax \neq b} \|Ax - b\| \tag{11}$$

and we can consider

$$\left(\frac{d(x, \mathcal{F})}{d_{\mathcal{F}}^{\min}}\right)^2 = \frac{\|Ax - b\|^2}{m^2}.$$

Computing m by (11) is NP-hard as it is equivalent to solving a shortest vector problem. However, from the integrality of A and b, we derive $m \geq 1$. This means that

$$p(x) = \|Ax - b\|^2 \geq \frac{\|Ax - b\|^2}{m^2} = \left(\frac{d(x, \mathcal{F})}{d_{\mathcal{F}}^{\min}}\right)^2$$

is a function that satisfies the conditions of Corollary 1, so that solving (1) is equivalent to solving

$$\begin{array}{ll} \min & q(x) + M\|Ax - b\|^2 \\ \text{s.t.} & x \in X \end{array} \tag{12}$$

with $M > ub - lb \geq 0$. Problem (12) is a quadratic integer optimization problem with only box constraints and the branch-and-bound algorithm proposed in Buchheim et al. [2] can be used in a straightforward way. We recall that this leads, at each node, to the solution of the dual of the following trust region problem

$$\begin{array}{ll} \min & x^\top Qx + c^\top x + M\|Ax - b\|^2 \\ \text{s.t.} & x^\top Hx \leq 1, \end{array}$$

for some $H \succ 0$ such that $E(H) = \{x \in \mathbb{R}^n \mid x^\top Hx \leq 1\} \supseteq [l, u]$.

Finally, we want to mention how we determine the values lb and ub needed to compute M. In order to compute a valid upper bound ub, a feasibility problem needs to be solved. A possibility is that of solving the following integer linear programming problem

$$\begin{array}{ll} \min & 0 \\ \text{s.t.} & Ax = b \\ & x \in X. \end{array} \tag{13}$$

For what concerns the computation of a valid lower bound lb, a continuous relaxation of problem (1) needs to be solved. A possibility, in order to deal with an easily solvable continuous relaxation, is to consider the following trust region problem

$$\begin{array}{ll} \min & x^\top Qx + c^\top x \\ \text{s.t.} & x^\top Hx \leq 1, \end{array} \tag{14}$$

with $H \succ 0$ such that $E(H) = \{x \in \mathbb{R}^n \mid x^\top Hx \leq 1\} \supseteq [l, u]$.

4. Theoretical comparison of the two approaches

From standard results in continuous optimization we can prove the following result, that helps us in interpreting how the penalty and the projection approaches compare to each other. For any given $H \succ 0$ let

$$q_H^* := \min\{q(x) \mid Ax = b;\ x^\top Hx \leq 1\}$$

and

$$f_H^*(M) := \min\{q(x) + M\|Ax - b\|^2 \mid x^\top Hx \leq 1\}\ .$$

Theorem 2. *Let $H \succ 0$ such that*

$$[l, u] \cap \mathbb{Z}^n \subseteq \{x \in \mathbb{R}^n \mid x^\top Hx \leq 1\}.$$

Then

$$q_H^* = \lim_{M \to \infty} f_H^*(M)\ .$$

Proof. Let x^* be a solution of the problem

$$\begin{array}{ll} \min & q(x) \\ \text{s.t.} & Ax = b \\ & x^\top Hx \leq 1. \end{array}$$

For $k \in \mathbb{N}$, consider the problem

$$\begin{array}{ll} \min & F^k(x) \\ \text{s.t.} & x^\top Hx \leq 1 \end{array} \tag{15}$$

where

$$F^k(x) = q(x) + k\|Ax - b\|^2.$$

Since $F^k(x)$ is continuous and $\{x \in \mathbb{R}^n \mid x^\top Hx \leq 1\}$ is compact, Problem (15) satisfies the assumptions of the Weierstrass Theorem for every k.

Let $\{x^k\}$ be the sequence of optimal solutions of problem (15). We prove that $q(\bar{x}) = q(x^*)$ for every limit point $\bar{x}$ of $\{x^k\}$. For every k, since x^k is the optimal solution of Problem (15), we have

$$F^k(x^k) = q(x^k) + k\|Ax^k - b\|^2 \leq F^k(x^*) = q(x^*). \tag{16}$$

Since $q(x^k)$ is bounded over $\{x \in \mathbb{R}^n \mid x^\top Hx \leq 1\}$ we obtain

$$\lim_{k \to +\infty} \|Ax^k - b\| = 0\ .$$

Therefore every limit point $\bar{x}$ of $\{x^k\}$ satisfies $A\bar{x} = b$.

Since the term $k\|Ax^k - b\|^2$ is nonnegative for every k, from (16) we get

$$q(x^k) \leq F^k(x^k) \leq q(x^*). \tag{17}$$

Taking the limit of (17) we have $q(\bar{x}) \leq q(x^*)$, but since

$$\bar{x} \in \{x \in \mathbb{R}^n \mid Ax = b,\ x^\top Hx \leq 1\}$$

this implies $q(\bar{x}) = q(x^*)$ by the optimality of x^*. □

Equation (16) in the proof of Theorem 2 states that for $M > 0$ and $H \succ 0$ the lower bound computed by the penalty approach is less or equal (i.e. not stronger) than the one computed by the projection approach.

5. Numerical experience

In this section we report computational results on a set of random integer quadratic instances with only one linear equality constraint. We consider an exact method based on the penalty formulation because, although in principle the bound is not better than in the projection approach, the implementation effort is milder.

Given an instance of Problem (1), we first compute a valid upper bound ub and a valid lower bound lb by solving problem (13) and problem (14), respectively. We then use $M = ub - lb + 0.01$ and consider problem (12). According to the results in Section 3, solving problem (12) with such a setting for the penalty parameter M is equivalent to solving our original instance.

Hence, we can use the branch-and-bound scheme `GQIP` defined in [2] to solve the box constrained quadratic integer programming problem (12). In the following, we compare the performance of `GQIP` and the MIQP solver of `CPLEX 12.6` [5]. In running `GQIP` we take into account the time needed to compute M, i.e., the time needed for solving problem (13) and problem (14). All experiments were carried out on Intel Xeon processors running at 2.60 GHz. We considered the constrained integer quadratic instances in `http://cedric.cnam.fr/~lamberta/Library/eiqp_iiqp.html`. These instances are of dimension $n = 20, 30, 40$ and have only one constraint each, consisting in a linear equation with all positive entries. For each dimension 15 instances are available. As already pointed out in [2] the performance of `GQIP` strongly depends on how large is the integer domain. For this reason we decided to deal with different domains of variables $[0, u]$, with $u = 1, 2, 10$. We also considered three different right hand sides for the linear constraint $a^\top x = b$, for a total of 405 instances. Namely we chose

$$b = \frac{u}{2}\sum_{i=1}^{n} a_i, \quad b = \frac{u}{10}\sum_{i=1}^{n} a_i, \quad b = \frac{u}{n}\sum_{i=1}^{n} a_i \, .$$

We state average results respectively in Table 1, in Table 2 and in Table 3. The results are grouped by dimension n. In the columns of the tables we list the number of instances solved to proven optimality within the limit of 3 hours, out of the 15 instance (Sol), the average over the successfully solved instances of the time spent to solve the instance to optimality (Time) and the average number of nodes (♯ nodes). The running times are given in CPU-seconds. By taking a look at the tables, we can notice that the instances seem to be harder the bigger is the right hand side of the linear constraint.

For the $[0, 1]$ instances, both `CPLEX` and `GQIP` are able to solve all the instances within the time limit and are very fast. On average, `GQIP` is faster than `CPLEX` on all instances but the ones with biggest right hand side, where `CPLEX` is slightly faster than `GQIP`. For the $[0, 2]$ instances, we see that `GQIP` and `CPLEX` are able to solve 124 and 100 instances out of 135, respectively. We can notice that `GQIP` is much faster than `CPLEX`, especially for the instances with $b = \frac{u}{2}\sum_{i=1}^{n} a_i$ and $b = \frac{u}{10}\sum_{i=1}^{n} a_i$. For the $[0, 10]$ instances, we see that `GQIP` and `CPLEX` are able to solve 8 and 128 instances out of 135, respectively. It is evident that `GQIP` suffers by the enlargement of the feasible domain. Indeed, the branch-and-bound framework of `GQIP` requires to solve a huge number of subproblems, which depends both on n and on the number of integer values in $[l, u]$.

We can conclude that the penalty formulation embedded into a branch-and-bound scheme as the one defined in [2] seems to be quite promising for solving integer quadratic problems with small integer domains $\{l, \ldots, u\}$. We also notice that the operations required at each node of the branch-and-bound scheme do not depend on the number of constraints. Indeed the matrix A plays a role only for the hessian of the quadratic function, and the solution of problem (13) and all calculations needed can be performed in the preprocessing phase.

[0,1]				
n	Alg	Sol	Time	♯ nodes
20	`GQIP`	15	0.01	3501.40
	`CPLEX`	15	0.51	123.47
30	`GQIP`	15	0.09	17938.47
	`CPLEX`	15	1.98	314.07
40	`GQIP`	15	1.15	169471.40
	`CPLEX`	15	7.50	640.20
[0,2]				
n	Alg	Sol	Time	♯ nodes
20	`GQIP`	15	0.05	34303.47
	`CPLEX`	15	131.42	191111.20
30	`GQIP`	15	3.18	1066773.87
	`CPLEX`	2	5957.31	2847726.00
40	`GQIP`	11	63.28	10085272.82
	`CPLEX`	1	1554.17	239087.00
[0,10]				
n	Alg	Sol	Time	♯ nodes
20	`GQIP`	1	32.98	18428173.00
	`CPLEX`	15	18.77	9339.00
30	`GQIP`	0	–	–
	`CPLEX`	14	1203.30	130090.64
40	`GQIP`	0	–	–
	`CPLEX`	6	2864.64	151549.00

Table 1: Results for instances with $b = \frac{u}{2}\sum_{i=1}^{n} a_i$.

[1] Billionnet, A., Elloumi, S., Lambert, A., 2014. A branch and bound algorithm for general mixed-integer quadratic programs based on quadratic convex relaxation. Journal of Combinatorial Optimization 28 (2), 376–399.

[2] Buchheim, C., De Santis, M., Palagi, L., Piacentini, M., 2013. An exact algorithm for nonconvex quadratic integer minimization using ellipsoidal relaxations. SIAM Journal on Optimization 23, 1867–1889.

[3] Buchheim, C., Wiegele, A., 2013. Semidefinite relaxations for nonconvex quadratic mixed-integer programming. Mathematical Programming 141 (1-2), 435–452.

[4] Conn, A. R., Gould, N. I. M., Toint, P. L., 2000. Trust-region methods. SIAM/MPS Series on Optimization, SIAM.

[5] IBM ILOG CPLEX Optimizer, 2014. Www.ibm.com/software/commerce/optimization/cplex-optimizer.

[6] Moré, J., 1993. Generalization of the trust region problem. Optimization Methods and Software 2, 189–209.

[7] Poljak, S., Rendl, F., Wolkowicz, H., 1995. A recipe for semidefinite relaxation for (0,1)-quadratic programming. Journal of Global Optimization 7 (1), 51–73.

[8] Pong, T., Wolkowicz, H., 2014. The generalized trust region subproblem. Computational Optimization and Applications 58 (2), 273–322.

[9] Sherali, H., Dalkiran, E., Liberti, L., 2012. Reduced RLT representations for nonconvex polynomial programming problems. Journal of Global Optimization 52, 447–469.

[10] Sherali, H. D., Adams, W. P., 1999. A reformulation-linearization technique for solving discrete and continuous nonconvex problems. Vol. 31

[0,1]				
n	Alg	Sol	Time	♯ nodes
20	GQIP	15	0.01	1364.87
	CPLEX	15	0.15	12.80
30	GQIP	15	0.08	12505.00
	CPLEX	15	0.70	161.93
40	GQIP	15	0.68	96701.13
	CPLEX	15	1.30	209.27
[0,2]				
n	Alg	Sol	Time	♯ nodes
20	GQIP	15	0.04	27114.33
	CPLEX	15	3.61	4517.67
30	GQIP	15	3.83	1336027.73
	CPLEX	15	454.07	561628.53
40	GQIP	9	54.90	9353147.11
	CPLEX	7	6363.40	3775002.57
[0,10]				
n	Alg	Sol	Time	♯ nodes
20	GQIP	3	49.87	31859582.67
	CPLEX	15	1.70	1590.40
30	GQIP	0	–	–
	CPLEX	15	48.95	18069.73
40	GQIP	0	–	–
	CPLEX	15	202.32	33115.07

Table 2: Results for instances with $b = \frac{u}{10} \sum_{i=1}^{n} a_i$.

[0,1]				
n	Alg	Sol	Time	♯ nodes
20	GQIP	15	0.01	882.08
	CPLEX	15	0.01	0.00
30	GQIP	15	0.04	4596.07
	CPLEX	15	0.02	0.00
40	GQIP	15	0.29	31308.00
	CPLEX	15	0.16	6.60
[0,2]				
n	Alg	Sol	Time	♯ nodes
20	GQIP	15	0.03	15923.00
	CPLEX	15	0.48	328.40
30	GQIP	15	1.09	361899.73
	CPLEX	15	5.75	1105.93
40	GQIP	14	31.49	6236915.86
	CPLEX	15	38.50	2981.00
[0,10]				
n	Alg	Sol	Time	♯ nodes
20	GQIP	4	31.98	18194087.50
	CPLEX	15	1.94	2075.73
30	GQIP	0	–	–
	CPLEX	15	4.28	1868.20
40	GQIP	0	–	–
	CPLEX	15	6.58	1269.00

Table 3: Results for instances with $b = \frac{u}{n} \sum_{i=1}^{n} a_i$.

of Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht.