An Inexact Quadratic Simplex Algorithm Tailored to Zero-One Polyhedra

Sven Mallach
Universität Bonn
Friedrich-Hirzebruch-Allee 8
53115 Bonn, Germany
sven.mallach@cs.uni-bonn.de

Abstract
A simplex algorithm is presented that is tailored to quadratic optimization over the vertices of zero-one polyhedra. While the algorithm can guarantee only local optima in general, computational experiments show a good performance on standard benchmark libraries.

Keywords Simplex Algorithm, Binary Quadratic Programming, Quadratic Assignment, Maximum Cut.

1 Introduction

We consider linearly constrained binary quadratic programs (BQPs) over bounded polyhedral sets, i.e., mathematical optimization problems of the form

$$\begin{aligned} \min\ & \frac{1}{2}\dot{x}^\top \dot{Q}\dot{x} + \dot{c}^\top \dot{x} \\ \text{s.t.}\ & \dot{A}\dot{x} \leq b \\ & \dot{x} \leq 1 \\ & \dot{x} \geq 0 \\ & \dot{x} \in \mathbb{Z}^n \end{aligned}$$

where $\dot{Q} \in \mathbb{R}^{n\times n}$, $\dot{c} \in \mathbb{R}^n$, $\dot{A} \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$. The only further assumption that we make is that $\dot{Q}$ be symmetric, i.e., $\dot{Q} \in \mathbb{S}^n$.

We strive at exploiting that this problem, in contrast to general quadratic integer programs, retains a property heavily relied on in linear programming. Namely, if it has a solution then it has an optimum *vertex* solution – simply because $P_I := \operatorname{conv}\left(\left\{\dot{x} \in \mathbb{Z}^n : \dot{A}\dot{x} \leq b,\ 0 \leq \dot{x} \leq 1\right\}\right)$ has no strictly interior integral points. Even more, any vertex of P_I *is* for sure among the vertices of $P := \left\{\dot{x} \in \mathbb{R}^n : \dot{A}\dot{x} \leq b,\ 0 \leq \dot{x} \leq 1\right\}$.

Ideally, we would thus be interested in solving the problem

$$\min \frac{1}{2}\dot{x}^\top \dot{Q}\dot{x} + \dot{c}^\top \dot{x} \text{ s.t. } \dot{x} \in \operatorname{ext}(P) \tag{1}$$

where $\operatorname{ext}(P)$ is the set of extreme points (vertices) of P, as the former observation implies that such a solution delivers a valid lower bound on the optimal value of the corresponding BQP. Clearly, this bound can be expected stronger than the one obtained by solving $\min \frac{1}{2}\dot{x}^\top \dot{Q}\dot{x} + \dot{c}^\top \dot{x}$ s.t. $\dot{x} \in P$. Moreover, if the objective is convex, i.e. if $\dot{Q}$ is positive semidefinite, and if P is a 0-1-polytope (i.e. $P = P_I$, for instance because $\dot{A}$ is totally unimodular and $b \in \mathbb{Z}^m$), then a solution to (1) is even optimal for the original problem. Solving (1) remains difficult though, especially since, even if f is convex, a vertex whose neighbors all have strictly worse objective values may be a *locally* optimal extreme point only, and thus the corresponding objective value need not be a valid lower bound for the corresponding BQP[1].

On the other hand, we can still take advantage by solving (1) inexactly if P is a 0-1-polytope (i.e. $P = P_I$). Because, even irrespective of the objective's curvature, a sequence of improving vertices then directly corresponds

Draft paper as of March 4, 2021

[1] As a simple example, consider the unconstrained BQP with $\dot{Q} = \begin{bmatrix} 4 & -3 \\ -3 & 4 \end{bmatrix}$, and $c = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. $\dot{Q}$ is positive definite, the global optimum is attained at $(x_1, x_2)^\top = (0,0)^\top$ with objective value 0 while $(1,1)^\top$ has objective value 1 but is adjacent only to the vertices $(0,1)^\top$, and $(1,0)^\top$ with objective value 2, i.e., $(1,1)^\top$ is a local optimum in terms of neighbor vertices.

©0000 (copyright holder)

to a sequence of improving upper (or more generally, primal) bounds and incumbent solutions. Finding these can be of value on its own, or e.g. be used to speed up branch and bound computations. Moreover, this is particularly attractive as any vertex (as readily obtained e.g. by a phase-I linear simplex algorithm or from any iteration of a lower-bounding framework such as the Gilmore-Lawler approach [9, 12], see also [4] for a perspective on more general BQPs) may serve as a "start vertex" for improvements.

Accordingly motivated, we present a simplex algorithm tailored to this special case of (inexact) quadratic optimization over zero-one polyhedra that resides in between the usual linear and quadratic simplex algorithms. More precisely, it incorporates ingredients to optimize over a quadratic objective typical to quadratic simplex algorithms, but it always stays at vertices of the polyhedron like the linear simplex algorithm does. Pivoting operations take place as long as a corresponding strictly improving direction is at hand (with acceptable effort). Due to that, and since the algorithm is supposed to perform pivots between 0-1-vertices only, numerical as well as cycling issues may be broadly circumvented. In fact, the proposed method is a decent extension to any existing (linear) simplex implementation.

The potentials of the proposed inexact simplex algorithm are demonstrated by applying it to the quadratic assignment problem (QAP), to the unconstrained binary quadratic optimization problem (UBQP, sometimes also referred to as Quadratic Unconstrained Binary Optimization, QUBO), and to the maximum cut problem. Especially for the latter two problems, the solutions sustained when starting the algorithm at a feasible solution obtained by computing the Gilmore-Lawler bound are frequently near-optimal, even though this starting solution is typically far away from that. Also for the quadratic assignment problem, the results on some QAPLIB instances are of unexpected quality. Finally, typically only a few iterations are necessary to obtain these results while further options to improve them exist, and a repeated application within a branch-and-bound search is promising.

The remainder of this paper is organized as follows: Sect. 2 provides references to related work and sets the common notation for the presentation of the proposed tailored simplex algorithm that is described in Sect. 3. The indicated computational study is the subject of Sect. 4, and the paper closes with a conclusion in Sect. 5.

2 Related Work and Preliminaries

This work is an immediate adaption of the simplex algorithm for linear programming [6] and its extensions to quadratic programming which have been described, for instance, in [1, 18, 2, 5, 16, 17, 7, 15, 11], and which all provide some means of describing non-extreme points as (almost) basic solutions to an extended system. Certain relations also exist to the quadratic programming algorithm by Frank and Wolfe [8]. For the sake of self-containedness, the most central employed concepts are briefly summarized in the following two subsections.

2.1 Extreme Point Representation in the Simplex Algorithm

Let us first briefly recall the representation of vertices of polyhedra in the simplex algorithm (for more details, we refer to the various textbooks, e.g. [13]).

For ease of presentation, we assume w.l.o.g. that any upper bounds on the variables $\dot{x}$ are either implied by or part of the system $\dot{A}\dot{x} \leq b$. Then, for $P := \left\{\dot{x} \in \mathbb{R}^n : \dot{A}\dot{x} \leq b,\ \dot{x} \geq 0\right\}$, the equation system $Ax = b$ with $A := \begin{bmatrix}\dot{A} & I_m\end{bmatrix}$, and $x := (\dot{x}, s)^\mathsf{T}$, where $s \in \mathbb{R}^m$, $s \geq 0$, are additional *slack variables*, will be referred to as the *augmented system* of $\dot{A}\dot{x} \leq b$. Since thus $s = b - \dot{A}\dot{x}$, we can equivalently write $P = \{x \in \mathbb{R}^{n+m} : Ax = b,\ x \geq 0\}$.

Let $B \subset \{1, \dots, n+m\}$ with $|B| = m$, and $N = \{1, \dots, n+m\} \setminus B$. Moreover let A_B (A_N) be the matrix consisting of the columns of A indexed by B (N), and let x_B (x_N) be the vector consisting of the x-components indexed by B (N). If A_B is regular then it is called (its columns form) a *basis* and a unique solution of the augmented system is given by $A_B x_B + A_N x_N = b$ where $x_B = A_B^{-1} b$ and $x_N = 0$. Similarly, the partition $(x_B, x_N)^\mathsf{T}$ of x is then called a *basic solution* to the system $Ax = b$. If $x \geq 0$ holds in addition, then $(x_B, x_N)^\mathsf{T}$ is called a *basic feasible solution* to the system $Ax = b$ and it corresponds to a vertex of P (cf. [13]). Finally, if $x_i \neq 0$ for all $i \in B$, then (x_B, x_N) is called *non-degenerate*, otherwise (x_B, x_N) is called *degenerate*.

In the degenerate case, it might be that one could replace some $i \in B$ with $x_i = 0$ by some $i' \in N$ such that $(B \cup \{i'\}) \setminus \{i\}$ again gives a basis leading to the same solution. Therefore, while each basic feasible solution of the augmented system $Ax = b$ uniquely corresponds to a vertex of the polyhedron P, the reverse need not be true in the presence of degeneracy.

2.2 Quadratic Objective Functions in the Simplex Algorithm

The following presentation of the quadratic ingredients is inspired by the descriptions in [13] and [15]. In accordance with an augmented system as described in the previous subsection, we also define a new symmetric matrix $Q = \begin{bmatrix} \dot{Q} & 0 \\ 0 & 0 \end{bmatrix}$ and a new vector $c = \begin{bmatrix} \dot{c} \\ 0 \end{bmatrix}$ for the augmented objective function $f(x) \coloneqq \frac{1}{2} x^\mathsf{T} Q x + c^\mathsf{T} x$. Then, for a vector $\lambda \in \mathbb{R}^m$ of dual multiplier variables, let us define the Lagrangian for our augmented problem as

$$L(x, \lambda) \coloneqq f(x) + \lambda(Ax - b) = \frac{1}{2} x^\mathsf{T} Q x + c^\mathsf{T} x + \lambda(Ax - b).$$

The *reduced cost* vector $\delta \in \mathbb{R}^{m+n}$ is the partial derivative of the Lagrangian w.r.t. x, i.e.

$$\delta \coloneqq \delta(x, \lambda) \coloneqq x^\mathsf{T} Q + c + \lambda^\mathsf{T} A.$$

Although, in general, we cannot hope to arrive at an optimum solution with the algorithm proposed even if the objective function is convex, it will use to some extent the well-known Karush-Kuhn-Tucker optimality conditions for a convex quadratic program with equality constraints. These state that a pair (x, λ) is optimal for a convex $f(x)$ and $P = \{x \in \mathbb{R}^{n+m} : Ax = b,\ x \geq 0\}$ if and only if

$$Ax = b, \quad x \geq 0, \quad \delta \geq 0, \text{ and } (\delta I)x = 0.$$

Given a basic feasible solution, we will refer to the square submatrix of Q with rows and columns indexed by B as $Q_B \in \mathbb{R}^{m \times m}$. Moreover, for some $j \in N$, we denote by q_j the components of the j-th column of Q that belong to the rows indexed by B. Finally, we assume that B is always ordered ascendingly, and we refer to the i-th index in B by $B(i)$.

3 An Inexact Quadratic Simplex Algorithm tailored to Zero-One Polyhedra

Although the description that follows is kept more general wherever possible, let us assume that the polyhedron $P = \{\dot{x} \in \mathbb{R}^n : \dot{A}\dot{x} \leq b,\ \dot{x} \geq 0\}$ under consideration is a subset of the unit hypercube (i.e., $\dot{x} \leq 1$ is either implied or enforced by the system $\dot{A}\dot{x} \leq b$ as indicated in Sect. 2.1) that is integral. Moreover, let us assume that the augmented system for the bounded polyhedron P is given as input along with a basic feasible pair B, N. Clearly, such a pair can be found or infeasibility (emptiness of P) detected e.g. with a phase-I linear simplex algorithm.

1 Solve the system $A_B x_B = b$;
2 **repeat**
3 Solve the system $A_B^\mathsf{T} \lambda = -Q_B x_B - c_B$; `// only if` f `is convex`
4 **forall** $j \in N$ **do**
5 $\quad w_j \leftarrow x_B^\mathsf{T} q_j$;
6 $\quad \delta_j \leftarrow c_j + w_j + \lambda^\mathsf{T} A_{\cdot j}$; `// only if` f `is convex`
7 $\quad$ **if** $\delta_j < 0$ *or* f *is non-convex* **then**
8 $\quad\quad$ Solve the system $A_B d = A_{\cdot j}$;
9 $\quad\quad \phi \leftarrow \min \left\{ \frac{x_{B(i)}}{d_i} : d_i > 0,\ i \in \{1, \ldots, m\} \right\}$;
10 $\quad\quad$ **if** $\phi > 0$ **then**
11 $\quad\quad\quad v \leftarrow Q_B d$;
12 $\quad\quad\quad \Delta \leftarrow w_j + \phi \left(c_j - c_B^\mathsf{T} d - x_B^\mathsf{T} v - q_j^\mathsf{T} d \right) + \frac{\phi^2}{2} \left(v^\mathsf{T} d + q_{jj} \right)$;
13 $\quad\quad\quad$ **if** $\Delta < 0$ **then**
14 $\quad\quad\quad\quad x_B \leftarrow x_B - \phi d$;
15 $\quad\quad\quad\quad x_j \leftarrow \phi$;
16 $\quad\quad\quad\quad$ Choose $i^* \in \left\{ i \in \{1, \ldots, m\} : d_i > 0, \frac{x_{B(i)}}{d_i} = \phi \right\}$;
17 $\quad\quad\quad\quad B \leftarrow (B \setminus \{i^*\}) \cup \{j\}$;
18 $\quad\quad\quad\quad N \leftarrow (N \setminus \{j\}) \cup \{i^*\}$;
19 $\quad\quad\quad\quad$ Go to line 2;
20 **return** x;

Algorithm 1: A Tailored Quadratic Simplex Algorithm

Theorem 1. *Let B, N be a basic feasible pair such that $x = (x_B, x_N)^\mathsf{T}$ is the corresponding basic feasible solution. Then Algorithm 1, started with this pair, terminates in a finite number of steps with a basic feasible solution $x^* = (x^*_{B^*}, x^*_{N^*})^\mathsf{T}$ such that $f(x^*) \leq f(x)$.*

Proof. Given B and N, solving the system in step 1 of Algorithm 1 determines the unique $x_B \in \mathbb{R}^m$ such that $A_B x_B = b$ while $x_N = 0$ as discussed in Sect. 2.1.

For this first and any potential further current solution $x = (x_B, x_N)^\intercal$ the main loop (lines 2 and following) of the algorithm proceeds as follows:

If f is convex (i.e., Q is positive semidefinite), in step 3 the system

$$A_B^\intercal \lambda = -Q_B x_B - c_B$$

is solved for λ. Therefore, the pair (x_B, λ) satisfies the Karush-Kuhn-Tucker conditions for the *basic* variables, i.e., we have in particular

$$0 = \delta_B = x_B^\intercal Q_B + c_B + \lambda^\intercal A_B.$$

The task is now to decide whether a strictly improving vertex $x' = (x'_{B'}, x'_{N'})^\intercal \in \text{ext}(P)$ exists such that B and B' differ in exactly one index. Thus, exactly as in the simplex algorithm for linear programming, we compute the direction d of change in the variables x_B when increasing a candidate x_j, $j \in N$, to enter the basis. That is, we consider $A_B x'_B + A_{\cdot j} x'_j = b$, i.e., $x'_B = A_B^{-1} b - A_B^{-1} A_{\cdot j} x'_j = x_B - d x'_j$ with d as computed in step 8. The increase of x_j is limited by the non-negativity restrictions on x_B, and the corresponding maximum increase ϕ of x_j is calculated in step 9. In particular, because P is bounded, there is always at least one $i \in \{1, \ldots, m\}$ such that $d_i > 0$, so ϕ is well defined (in our case of a zero-one polytope, of course $d_i \in \{-1, 0, 1\}$). Observe also that ϕ does not depend on the choice of the variable to leave the basis, so one may select an arbitrary one that takes on the value zero when increasing x_j to ϕ (which is always equal to zero or one for a zero-one polytope).

In the convex case, increasing x_j, $j \in N$, can only decrease the objective if its partial derivative δ_j is strictly negative at (x, λ). So in this case, δ_j is calculated in step 6, and x_j is further considered only if $\delta_j < 0$ in step 7. In the non-convex case, it may be that a total decrease of the objective is reached when moving from x to x' even though $\delta_j \geq 0$ at (x, λ). In any case, the new solution

$$\begin{pmatrix} x'_B \\ x'_j \end{pmatrix} = \begin{pmatrix} x_B - \phi d \\ \phi \end{pmatrix} \text{ and } x'_k = 0 \text{ for all } k \in \{1, \ldots, n+m\} \setminus (B \cup \{j\}) \tag{2}$$

will only differ from x if ϕ is strictly positive which is checked at line 10. If $\phi > 0$, then a pivot from x to x' shall and will take place if the change Δ of the objective is negative (line 13).

Clearly, $f(x) = c_B x_B + \frac{1}{2} x_B^\intercal Q_B x_B$. Moreover, by exploiting (2), we have

$$\begin{aligned}
f(x') &= c_j \phi + c_B^\intercal (x_B - \phi d) + \frac{1}{2} \begin{bmatrix} (x_B - \phi d) & \phi \end{bmatrix} \begin{bmatrix} Q_B & q_j \\ q_j^\intercal & q_{jj} \end{bmatrix} \begin{bmatrix} (x_B - \phi d) \\ \phi \end{bmatrix} \\
&= c_j \phi + c_B^\intercal (x_B - \phi d) + \frac{1}{2} \left((x_B - \phi d)^\intercal Q_B (x_B - \phi d) + 2 q_j^\intercal (x_B - \phi d) + q_{jj} \phi^2 \right) \\
&= c_B^\intercal x_B + c_j \phi - c_B^\intercal \phi d + \frac{1}{2} \left(x_B^\intercal Q_B x_B - x_B^\intercal Q_B \phi d - \phi d^\intercal Q_B x_B + \phi d^\intercal Q_B \phi d + 2 q_j^\intercal x_B - 2 q_j^\intercal \phi d + q_{jj} \phi^2 \right) \\
&= c_B^\intercal x_B + \frac{1}{2} x_B^\intercal Q_B x_B + c_j \phi - \phi c_B^\intercal d - \phi x_B^\intercal Q_B d + q_j^\intercal x_B - \phi q_j^\intercal d + \frac{1}{2} \left(\phi d^\intercal Q_B \phi d + q_{jj} \phi^2 \right) \\
&= c_B^\intercal x_B + \frac{1}{2} x_B^\intercal Q_B x_B + c_j \phi - \phi c_B^\intercal d - \phi x_B^\intercal v + w_j - \phi q_j^\intercal d + \frac{1}{2} \left(\phi^2 v^\intercal d + q_{jj} \phi^2 \right) [\text{with } v = Q_B d \text{ and } w = x_B^\intercal q_j] \\
&= c_B^\intercal x_B + \frac{1}{2} x_B^\intercal Q_B x_B + \Delta \\
&= f(x) + \Delta.
\end{aligned}$$

We can thus move to a strictly better basic feasible solution only if $\Delta < 0$ for some $j \in N$, in which case a corresponding pivot operation is carried out. If this does not happen for any $j \in N$, line 20 is reached, and Algorithm 1 terminates. Since only strictly improving pivots take place, and P is bounded, termination is guaranteed in a finite number of steps with a solution $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$. ◀

Remark 2. *If f is convex, and $\delta_j \geq 0$ holds for all $j \in N$, then the corresponding $x = (x_B, x_N)$ is an optimum solution to the program* $\min\{f(x) : x \in P\}$ *as it satisfies the entire Karush-Kuhn-Tucker conditions.*

Remark 3. *If $\Delta \geq 0$ for all $j \in N$, still a pivot operation with $\phi = 0$ might exist such that the objective function can be strictly decreased* after *performing that pivot. However, in general, cycling then needs to be prevented by further means.*

4 Computational Study

We evaluate Algorithm 1 on established Quadratic Assignment, Unconstrained Binary Quadratic Programming, and Maximum Cut instances while modeling these as described in the respective subsections below.

The corresponding table columns list (in this order) the instance name, the Gilmore-Lawler Bound (GLB) [9, 12, 4], the optimum value, the trivial upper bound (TUB) obtained by just evaluating the objective value for the feasible solution obtained when computing the GLB, the upper bound (UB) obtained when starting the proposed algorithm with this feasible solution, the optimality gap (in percent, cut with no rounding after the second decimal digit) associated with UB, and the number of iterations (pivots) until termination.

For these experiments, the entering variable in a pivot operation of Algorithm 1 is always chosen to be the one with the smallest index having $\Delta < 0$, and the leaving variable is as well the candidate with the smallest index. Of course different selection rules (such as e.g. striving for the smallest $\Delta < 0$) might lead to better results on average. In addition to the results displayed, we found that a repeated application of Algorithm 1 at the nodes of a GLB-based branch-and-bound framework led to further improvements of the primal bound quickly.

4.1 Quadratic Assignment

Given $T, D \in \mathbb{R}^{n \times n}$ and $c \in \mathbb{R}^n$, a QAP in Koopmans-Beckmann form can be written as:

$$\begin{aligned}
\min & \sum_{i=1}^{n}\sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{q=1}^{n} t_{ij} d_{pq}\; x_{ip} x_{jq} + \sum_{i=1}^{n}\sum_{p=1}^{n} c_{ip} x_{ip} && \\
& \sum_{i=1}^{n} x_{ip} = 1 && \text{for all } p \in \{1, \ldots, n\} \\
& \sum_{p=1}^{n} x_{ip} = 1 && \text{for all } i \in \{1, \ldots, n\} \\
& x_{ip} \geq 0 && \text{for all } i, p \in \{1, \ldots, n\} \\
& x_{ip} \in \mathbb{Z} && \text{for all } i, p \in \{1, \ldots, n\}
\end{aligned}$$

As is well known, the Birkhoff Polytope associated to the convex hull of the feasible solutions to this problem is integral as the constraint matrix and right hand side vector of the above formulation are totally unimodular and integral, respectively.

We look at Koopmans-Beckmann QAPs with known optima as given from the QAPLIB [3]. The results are displayed in Table 1 while the optimal values were retrieved from the current QAPLIB website.

4.2 Unconstrained Binary Quadratic Optimization

Here, the feasible set are the vertices of the unit hypercube of dimension n, i.e., given $Q \in \mathbb{R}^{n \times n}$ and $c \in \mathbb{R}^n$, the problem under consideration is:

$$\begin{aligned}
\min\ & \frac{1}{2} x^\mathsf{T} Q x + c^\mathsf{T} x \\
\text{s.t.}\ & x \leq 1 \\
& x \geq 0 \\
& x \in \mathbb{Z}^n
\end{aligned}$$

We look at the unconstrained BQP instances in the BiqMac Library [14] that have known optima, and we remark that the GLB is typically weak for this setting, in particular considerably weaker than e.g. the bound obtained with the "standard linearization" [10]. The results are displayed in Table 2.

4.3 Maximum Cut

Let $G = (V, E)$ be an undirected graph with edge weights $w : E \mapsto \mathbb{R}$. We here treat the Maximum Cut problem on G as a special binary quadratic optimization problem where the vertices $v \in V$ whose variables x_v are assigned the value zero and one form the two partitions of a cut, respectively. Clearly, an edge $\{i, j\} \in E$ is then a "cut edge" if and only if $x_i = 0$ and $x_j = 1$ or vice versa, i.e., the weight of a cut is $\sum_{\{i,j\} \in E} w_{ij}\, (x_i(1 - x_j) + x_j(1 - x_i))$.

Assuming $n := |V|$, it is thus easily verified that the following BQP models the Maximum Cut problem on G:

$$\begin{aligned} \min \sum_{\{i,j\}\in E} & -w_{ij}\left(x_i + x_j - 2x_i x_j\right) \\ \text{s.t. } & x \le 1 \\ & x \ge 0 \\ & x \in \mathbb{Z}^n \end{aligned}$$

The results are displayed in Tables 3 and 4, and of course the weakness of the GLB bounds applies again.

5 Conclusion

In a computational study, we found that an inexact simplex algorithm that is tailored to quadratic optimization over zero-one polyhedra may deliver good locally optimum vertex solutions when applied to well-established quadratic assignment, unconstrained binary quadratic optimization, and maximum cut instances. Especially for the latter two problems, the solutions obtained when starting the algorithm at a feasible solution obtained when

Instance	GLB	OPT	TUB	UB	Gap	nIter
bur26a	5315200	5426670	6070481	5747970	5.92	19
bur26b	3714750	3817852	4319551	4154702	8.82	18
bur26c	5312038	5426795	5912727	5679118	4.64	17
bur26d	3711739	3821225	4291087	4004870	4.80	21
bur26e	5307079	5386879	6161183	5835796	8.33	23
bur26f	3706888	3782044	4378791	4143061	9.54	18
bur26g	9978273	10117172	11196006	10738840	6.14	27
bur26h	6973253	7098658	8130587	7370491	3.82	24
chr12a	7245	9552	44232	10214	6.93	7
chr12b	7146	9742	25580	15182	55.84	3
chr12c	7976	11156	18784	13316	19.36	4
chr15a	5625	9896	50174	20566	107.82	6
chr15b	4653	7990	54254	24000	200.37	6
chr15c	6165	9504	44602	23124	143.30	3
chr18a	6779	11098	71964	30756	177.13	7
chr18b	1534	1534	2926	2926	90.74	0
chr20a	2150	2192	11262	5126	133.85	5
chr20b	2196	2298	8942	5922	157.70	8
chr20c	8601	14142	82846	40294	184.92	9
chr22a	5924	6156	12872	10316	67.57	11
chr22b	5936	6194	13334	8774	41.65	12
chr25a	2765	3796	17436	11426	201.00	16
els19	11971949	17212548	24357730	23179404	34.66	6
esc128	2	64	202	202	215.62	0
esc16a	38	68	94	94	38.23	0
esc16b	220	292	320	320	9.58	0
esc16c	83	160	196	196	22.50	0
esc16d	3	16	62	62	287.50	0
esc16e	12	28	50	50	78.57	0
esc16f	0	0	0	-0	0.00	0
esc16g	12	26	30	30	15.38	0
esc16h	625	996	1518	1518	52.40	0
esc16i	0	14	40	40	185.71	0
esc16j	1	8	22	22	175.00	0
esc32a	35	130	368	368	183.07	0
esc32b	96	168	320	320	90.47	0
esc32c	350	642	866	866	34.890	0
esc32d	106	200	340	340	70.00	0
esc32e	0	2	30	30	1400.00	0
esc32g	0	6	28	28	366.66	0
esc32h	257	438	630	630	43.83	0
esc64a	47	116	254	254	118.96	0
had12	1536	1652	1748	1684	1.93	4
had14	2492	2724	3036	2800	2.79	8
had16	3358	3720	4196	3894	4.67	9
had18	4776	5358	5810	5614	4.77	10
had20	6166	6922	7464	7136	3.09	3
kra30a	68360	88900	117470	105390	18.54	17
kra30b	69065	91420	123590	105090	14.95	16
kra32	67390	88900	120320	112900	26.99	13
lipa20a	3667	3683	3954	3842	4.31	11
lipa20b	27076	27076	27076	27076	0.00	0
lipa30a	13147	13178	13902	13678	3.79	19
lipa30b	151426	151426	151426	151426	0.00	0
lipa40a	31497	31538	32753	32384	2.68	31
lipa40b	476581	476581	476581	476581	0.00	0
lipa50a	62020	62093	64141	63561	2.36	37
lipa50b	1210244	1210244	1210244	1210244	0.00	0
lipa60a	107123	107218	110203	109517	2.14	27
lipa60b	2520135	2520135	2520135	2520135	0.00	0
lipa70a	169647	169755	173687	172935	1.87	32
lipa70b	4603200	4603200	4603200	4603200	0.00	0
lipa80a	253041	253195	258791	257131	1.55	62
lipa80b	7763962	7763962	7763962	7763962	0.00	0
lipa90a	360446	360630	367591	365545	1.36	68
lipa90b	12490441	12490441	12490441	12490441	0.00	0
nug12	493	578	822	606	4.84	11
nug14	852	1014	1202	1024	0.98	2
nug15	963	1150	1470	1214	5.56	7
nug16a	1314	1610	2094	1802	11.92	9
nug16b	1022	1240	1660	1382	11.45	12
nug17	1388	1732	2256	1932	11.54	12
nug18	1554	1930	2424	2186	13.26	7
nug20	2057	2570	3204	2772	7.85	7
nug21	1833	2438	3244	2764	13.37	10
nug22	2483	3596	4692	3988	10.90	12
nug24	2676	3488	4336	3836	9.97	20
nug25	2869	3744	4576	4200	12.17	11
nug27	3701	5234	6310	5874	12.22	11
nug28	3786	5166	6184	5686	10.06	18
nug30	4539	6124	7644	7014	14.53	8
rou12	202272	235528	282622	269294	14.33	1
rou15	298548	354210	416852	408030	15.19	1
rou20	599948	725520	820286	768426	5.91	8
scr12	27858	31410	47002	42574	35.54	3
scr15	44737	51140	85408	65134	27.36	7
scr20	86766	110030	189670	145460	32.20	11
ste36a	7124	9526	16568	13776	44.61	22
ste36b	8653	15852	59640	35310	122.74	19
ste36c	6393629	8239110	15323974	13363510	62.19	11
tai10a	110828	135028	171022	155448	15.12	2
tai10b	577324	1183760	1944490	1767441	49.30	3
tai12a	195918	224416	288476	250486	11.61	5
tai12b	9788461	39464925	97832072	57359953	45.34	10
tai15a	327501	388214	477922	446822	15.09	5
tai15b	11242074	51765268	496268635	52717471	1.83	20
tai17a	412722	491812	589836	545284	10.87	6
tai20a	580674	703482	877710	799536	13.65	14
tai20b	14205869	122455319	193877883	188528152	53.95	7
tai25a	962417	1167256	1457662	1306470	11.92	23
tai25b	47692620	344355646	714839287	660102683	91.69	23
tai30b	41183334	637117113	1310018943	901649709	41.52	32
tho30	90578	149936	199940	174992	16.71	18

Table 1 Results on Quadratic Assignment instances.

Instance	GLB	OPT	TUB	UB	Gap	nIter
be100.1	-125802	-19412	-492	-19025	1.99	121
be100.2	-125598	-17290	184	-17213	0.44	83
be100.3	-124248	-17565	2826	-17532	0.18	83
be100.4	-126525	-19125	-2263	-19025	0.52	96
be100.5	-123756	-15868	2406	-15396	2.97	101
be100.6	-125247	-17368	1271	-17291	0.44	95
be100.7	-128542	-18629	-4680	-18256	2.00	85
be100.8	-127508	-18649	-3578	-18488	0.86	84
be100.9	-120449	-13294	9219	-13233	0.45	122
be100.10	-123917	-15352	1449	-15352	0.00	93
be120.3.1	-54598	-13067	-796	-12814	1.93	101
be120.3.2	-55188	-13046	2	-12616	3.29	84
be120.3.3	-53691	-12418	-1239	-12069	2.81	88
be120.3.4	-55598	-13867	-3830	-13605	1.88	94
be120.3.5	-51392	-11403	3248	-11403	0.00	110
be120.3.6	-55042	-12915	-318	-12513	3.11	82
be120.3.7	-53792	-14068	-2142	-14036	0.22	78
be120.3.8	-57190	-14701	-6298	-14490	1.43	66
be120.3.9	-50742	-10458	4546	-9569	8.50	117
be120.3.10	-53148	-12201	876	-12129	0.59	97
be120.8.1	-144721	-18691	-79	-18629	0.33	110
be120.8.2	-142061	-18827	2747	-17485	7.12	119
be120.8.3	-144963	-19302	1595	-19111	0.98	101
be120.8.4	-143877	-20765	1159	-19770	4.79	125
be120.8.5	-141172	-20417	4668	-20178	1.17	137
be120.8.6	-143242	-18482	2210	-18374	0.58	129
be120.8.7	-148077	-22194	-5559	-21809	1.73	89
be120.8.8	-144745	-19534	-2413	-19515	0.09	116
be120.8.9	-140387	-18195	7475	-17947	1.36	131
be120.8.10	-140553	-19049	3805	-19049	0.00	113
be150.3.1	-86965	-18889	-2135	-18703	0.98	116
be150.3.2	-87158	-17816	-1156	-17543	1.53	108
be150.3.3	-83363	-17314	1135	-16832	2.78	135
be150.3.4	-86909	-19884	-4737	-19350	2.68	110
be150.3.5	-83705	-16817	1657	-16717	0.59	148
be150.3.6	-87389	-16780	-729	-16395	2.29	117
be150.3.7	-85306	-18001	-1280	-17936	0.36	125
be150.3.8	-87567	-18303	-5267	-17732	3.11	108
be150.3.9	-77471	-12838	7557	-12153	5.33	151
be150.3.10	-84271	-17963	-1699	-17526	2.43	91
be150.8.1	-227218	-27089	-348	-26431	2.42	129
be150.8.2	-226679	-26779	1065	-26355	1.58	134
be150.8.3	-227891	-29438	25	-29302	0.46	185
be150.8.4	-226145	-26911	2369	-25624	4.78	142
be150.8.5	-221172	-28017	6398	-27827	0.67	170
be150.8.6	-227776	-29221	-4376	-29221	0.00	126
be150.8.7	-230788	-31209	-8208	-30511	2.23	141
be150.8.8	-227653	-29730	-2807	-29001	2.45	160
be150.8.9	-221911	-25388	8699	-24806	2.29	191
be150.8.10	-224376	-28374	-866	-28269	0.37	118
be200.3.1	-152401	-25453	-555	-25151	1.18	172
be200.3.2	-151703	-25027	2929	-24809	0.87	197
be200.3.3	-148092	-28023	1098	-28014	0.03	114
be200.3.4	-149072	-27434	2900	-27234	0.72	205
be200.3.5	-148723	-26355	3277	-25829	1.99	153
be200.3.6	-153205	-26146	-735	-25303	3.22	174
be200.3.7	-153803	-30483	-5039	-30473	0.03	156
be200.3.8	-152233	-27355	-1423	-27032	1.18	163
be200.3.9	-148438	-24683	5726	-24429	1.02	178
be200.3.10	-148828	-23842	2724	-23626	0.90	211
be200.8.1	-405825	-48534	-3603	-48212	0.66	257
be200.8.2	-401323	-40821	-1045	-39731	2.67	289
be200.8.3	-401356	-43207	5064	-42179	2.37	227
be200.8.4	-400720	-43757	6956	-43479	0.63	219
be200.8.5	-395073	-41482	10213	-40876	1.46	264
be200.8.6	-404929	-49492	-2345	-49492	0.00	210
be200.8.7	-409960	-46828	-8800	-46819	0.01	233
be200.8.8	-403311	-44502	-3237	-43966	1.20	218
be200.8.9	-398446	-43241	7070	-42252	2.28	221
be200.8.10	-398471	-42832	1897	-41662	2.73	214
be250.1	-78724	-24076	-2250	-23768	1.27	154
be250.2	-79245	-22540	-1051	-22286	1.12	179
be250.3	-78674	-22923	-570	-22665	1.12	192
be250.4	-81066	-24649	-3951	-24527	0.49	170
be250.5	-78482	-21057	1385	-20516	2.56	197
be250.6	-80863	-22735	-1573	-22691	0.19	173
be250.7	-80225	-24095	-2652	-23606	2.02	155
be250.8	-80387	-23801	-5333	-23205	2.50	135
be250.9	-73173	-20051	6601	-19767	1.41	226
be250.10	-79375	-23159	-1433	-22995	0.70	248

Instance	GLB	OPT	TUB	UB	Gap	nIter
bqp50-1	-3956	-2098	2065	-1825	13.01	26
bqp50-2	-6598	-3702	-700	-3359	9.26	16
bqp50-3	-7552	-4626	-2262	-4626	0.00	23
bqp50-4	-5591	-3544	-1485	-3430	3.21	17
bqp50-5	-6432	-4012	-1696	-3780	5.78	19
bqp50-6	-5944	-3693	-2206	-3664	0.78	8
bqp50-7	-7196	-4520	-2996	-4460	1.32	7
bqp50-8	-7386	-4216	-1692	-4216	0.00	21
bqp50-9	-6471	-3780	-1163	-3748	0.84	21
bqp50-10	-5879	-3507	-1821	-3359	4.22	12
bqp100-1	-20269	-7970	2807	-7257	8.94	65
bqp100-2	-24571	-11036	-1455	-10508	4.78	57
bqp100-3	-27329	-12723	-6069	-12494	1.79	34
bqp100-4	-24145	-10368	221	-10216	1.46	56
bqp100-5	-21226	-9083	978	-8791	3.21	52
bqp100-6	-25925	-10210	-771	-10202	0.07	65
bqp100-7	-24033	-10125	-707	-9935	1.87	45
bqp100-8	-25198	-11435	-766	-11252	1.60	49
bqp100-9	-24466	-11455	-2102	-11154	2.62	46
bqp100-10	-27330	-12565	-4686	-12415	1.19	39
bqp250-1	-156642	-45607	1214	-44695	1.99	177
bqp250-2	-156517	-44810	-5797	-42470	5.22	139
bqp250-3	-161838	-49037	-16642	-49013	0.04	134
bqp250-4	-150822	-41274	7978	-40676	1.44	189
bqp250-5	-159945	-47961	-4665	-47441	1.08	160
bqp250-6	-156905	-41014	2917	-40625	0.94	195
bqp250-7	-160080	-46757	-8740	-46226	1.13	179
bqp250-8	-145199	-35726	13323	-33077	7.41	161
bqp250-9	-163677	-48916	-12071	-48228	1.40	173
bqp250-10	-151505	-40442	1657	-39378	2.63	200
gka1a	-5284	-3414	-1607	-3378	1.05	15
gka1b	-782	-133	17568	-98	26.31	20
gka1c	-16749	-5058	-1627	-5058	0.00	21
gka1d	-14010	-6333	-2396	-6209	1.95	52
gka1e	-47602	-16464	-396	-16122	2.07	112
gka2a	-9664	-6063	-3627	-6063	0.00	13
gka2b	-1037	-121	41031	-121	0.00	33
gka2c	-20353	-6213	-2343	-6213	0.00	16
gka2d	-24612	-6579	1814	-6484	1.44	64
gka2e	-102696	-23395	-3196	-23285	0.47	176
gka3a	-12581	-6037	-2216	-5896	2.33	24
gka3b	-1249	-118	70827	-60	49.15	42
gka3c	-19201	-6665	-2517	-6665	0.00	29
gka3d	-36107	-9261	1839	-9148	1.22	78
gka3e	-145766	-25243	4410	-24867	1.48	173
gka4a	-16758	-8598	-3509	-8589	0.10	29
gka4b	-1645	-129	116669	-54	58.13	51
gka4c	-20222	-7398	-2538	-7398	0.00	40
gka4d	-50313	-10727	1439	-10705	0.20	66
gka4e	-204221	-35594	-5813	-35346	0.69	176
gka5a	-13141	-5737	-1915	-5737	0.00	27
gka5b	-1898	-150	175052	-150	0.00	65
gka5c	-17910	-7362	-3672	-7272	1.22	32
gka5d	-61464	-11626	110	-11194	3.71	96
gka5e	-255512	-35154	-6924	-34550	1.71	207
gka6a	-10219	-3980	-1381	-3980	0.00	16
gka6b	-2274	-146	240964	-62	57.53	70
gka6c	-11319	-5824	-2513	-5779	0.77	43
gka6d	-74669	-14207	-1245	-14121	0.60	62
gka7a	-11671	-4541	-1597	-4541	0.00	18
gka7b	-2520	-160	312422	-86	46.25	81
gka7c	-13829	-7225	-4207	-7160	0.89	39
gka7d	-88343	-14476	-3063	-14368	0.74	89
gka8a	-16926	-11109	-5204	-10947	1.45	47
gka8b	-2871	-145	388233	-85	41.37	90
gka8d	-100479	-16352	-501	-16352	0.0	87
gka9b	-3313	-137	484119	-97	29.19	100
gka9d	-110260	-15656	3992	-15153	3.21	113
gka10b	-3930	-154	732500	-90	41.55	127
gka10d	-127661	-19102	-5309	-18823	1.46	83

Table 2 Results on unconstrained binary quadratic programming instances.

Instance	GLB	OPT	TUB	UB	Gap	nIter
g05_60.0	-1770	-536	0	-529	1.30	60
g05_60.1	-1770	-532	0	-516	3.00	56
g05_60.2	-1770	-529	0	-526	0.56	52
g05_60.3	-1770	-538	0	-512	4.83	47
g05_60.4	-1770	-527	0	-518	1.70	57
g05_60.5	-1770	-533	0	-530	0.56	57
g05_60.6	-1770	-531	0	-527	0.75	53
g05_60.7	-1770	-535	0	-530	0.93	65
g05_60.8	-1770	-530	0	-512	3.39	52
g05_60.9	-1770	-533	0	-522	2.06	62
g05_80.0	-3160	-929	0	-917	1.29	84
g05_80.1	-3160	-941	0	-941	0.00	100
g05_80.2	-3160	-934	0	-910	2.56	80
g05_80.3	-3160	-923	0	-903	2.16	74
g05_80.4	-3160	-932	0	-911	2.25	78
g05_80.5	-3160	-926	0	-919	0.75	88
g05_80.6	-3160	-929	0	-913	1.72	78
g05_80.7	-3160	-929	0	-910	2.04	81
g05_80.8	-3160	-925	0	-914	1.18	82
g05_80.9	-3160	-923	0	-902	2.27	73
g05_100.0	-4950	-1430	0	-1406	1.67	106
g05_100.1	-4950	-1425	0	-1406	1.33	94
g05_100.2	-4950	-1432	0	-1414	1.25	115
g05_100.3	-4950	-1424	0	-1413	0.77	106
g05_100.4	-4950	-1440	0	-1415	1.73	98
g05_100.5	-4950	-1436	0	-1413	1.60	110
g05_100.6	-4950	-1434	0	-1418	1.11	113
g05_100.7	-4950	-1431	0	-1403	1.95	104
g05_100.8	-4950	-1432	0	-1386	3.21	97
g05_100.9	-4950	-1430	0	-1408	1.53	84
pm1d_80.0	-3048	-227	0	-203	10.57	60
pm1d_80.1	-3114	-245	0	-223	8.97	45
pm1d_80.2	-3194	-284	0	-248	12.67	76
pm1d_80.3	-3184	-291	0	-267	8.24	72
pm1d_80.4	-3112	-251	0	-251	0.00	61
pm1d_80.5	-3118	-242	0	-233	3.71	70
pm1d_80.6	-3022	-205	0	-189	7.80	41
pm1d_80.7	-3120	-249	0	-223	10.44	65
pm1d_80.8	-3202	-293	0	-257	12.28	51
pm1d_80.9	-3132	-258	0	-219	15.11	58
pm1d_100.0	-4872	-340	0	-310	8.82	79
pm1d_100.1	-4840	-324	0	-319	1.54	85
pm1d_100.2	-4972	-389	0	-362	6.94	132
pm1d_100.3	-4984	-400	0	-397	0.75	97
pm1d_100.4	-4906	-363	0	-318	12.39	70
pm1d_100.5	-5034	-441	0	-432	2.04	90
pm1d_100.6	-4886	-367	0	-343	6.53	80
pm1d_100.7	-4858	-361	0	-358	0.83	94
pm1d_100.8	-4932	-385	0	-356	7.53	90
pm1d_100.9	-4980	-405	0	-382	5.67	97
pm1s_100.0	-520	-127	0	-110	13.38	48
pm1s_100.1	-524	-126	0	-115	8.73	38
pm1s_100.2	-522	-125	1	-99	20.80	38
pm1s_100.3	-494	-111	8	-90	18.91	43
pm1s_100.4	-528	-128	0	-118	7.81	56
pm1s_100.5	-524	-128	2	-110	14.06	43
pm1s_100.6	-518	-122	0	-105	13.93	43
pm1s_100.7	-482	-112	0	-93	16.96	42
pm1s_100.8	-504	-120	3	-111	7.50	53
pm1s_100.9	-516	-127	0	-104	18.11	42
pm1s_80.0	-308	-79	4	-65	17.72	30
pm1s_80.1	-326	-85	0	-65	23.52	34
pm1s_80.2	-330	-82	3	-63	23.17	34
pm1s_80.3	-316	-81	14	-75	7.40	39
pm1s_80.4	-294	-70	10	-53	24.28	26
pm1s_80.5	-328	-87	3	-74	14.94	38
pm1s_80.6	-304	-73	0	-60	17.80	32
pm1s_80.7	-322	-83	0	-66	20.48	37
pm1s_80.8	-316	-81	3	-68	16.04	37
pm1s_80.9	-294	-70	10	-59	15.71	30
pw01_100.0	-5422	-2019	0	-1908	5.49	84
pw01_100.1	-5530	-2060	0	-2024	1.74	110
pw01_100.2	-5464	-2032	0	-1939	4.57	87
pw01_100.3	-5572	-2067	0	-1920	7.11	81
pw01_100.4	-5430	-2039	0	-1985	2.64	98
pw01_100.5	-5658	-2108	0	-2022	4.07	107
pw01_100.6	-5452	-2032	0	-1915	5.75	109
pw01_100.7	-5564	-2074	0	-1981	4.48	91
pw01_100.8	-5358	-2022	0	-1905	5.78	83
pw01_100.9	-5446	-2005	0	-1957	2.39	104
pw05_100.0	-27602	-8190	0	-8094	1.17	114
pw05_100.1	-27080	-8045	0	-7976	0.85	126
pw05_100.2	-27022	-8039	0	-7932	1.33	148
pw05_100.3	-27328	-8139	0	-7905	2.87	126
pw05_100.4	-27406	-8125	0	-8070	0.67	103
pw05_100.5	-27400	-8169	0	-8014	1.89	105
pw05_100.6	-27646	-8217	0	-8048	2.05	119
pw05_100.7	-27648	-8249	0	-8079	2.06	108
pw05_100.8	-27474	-8199	0	-7846	4.30	95
pw05_100.9	-27312	-8099	0	-7844	3.14	125
pw09_100.0	-49214	-13585	0	-13558	0.19	156
pw09_100.1	-48534	-13417	0	-13152	1.97	110
pw09_100.2	-48784	-13461	0	-13240	1.64	114
pw09_100.3	-49404	-13656	0	-13468	1.37	119
pw09_100.4	-48942	-13514	0	-13408	0.78	128
pw09_100.5	-49254	-13574	0	-13559	0.11	132
pw09_100.6	-49492	-13640	0	-13475	1.20	143
pw09_100.7	-49032	-13501	0	-13357	1.06	106
pw09_100.8	-49210	-13593	0	-13446	1.08	162
pw09_100.9	-49398	-13658	0	-13535	0.90	142
w01_100.0	-2528	-651	0	-559	14.13	60
w01_100.1	-2650	-719	0	-691	3.89	70
w01_100.2	-2672	-676	47	-589	12.86	50
w01_100.3	-2906	-813	22	-702	13.65	59
w01_100.4	-2520	-668	52	-621	7.03	67
w01_100.5	-2538	-643	0	-584	9.17	57
w01_100.6	-2546	-654	0	-572	12.53	56
w01_100.7	-2722	-725	22	-672	7.31	71
w01_100.8	-2650	-721	0	-575	20.24	57
w01_100.9	-2676	-729	28	-636	12.75	76
w05_100.0	-13164	-1646	0	-1552	5.71	87
w05_100.1	-13256	-1606	0	-1437	10.52	77
w05_100.2	-13832	-1902	0	-1768	7.04	123
w05_100.3	-13154	-1627	0	-1484	8.78	94
w05_100.4	-12848	-1546	0	-1449	6.27	103
w05_100.5	-13174	-1581	0	-1297	17.96	83
w05_100.6	-13066	-1479	0	-1422	3.85	94
w05_100.7	-13698	-1987	0	-1857	6.54	123
w05_100.8	-12500	-1311	0	-1210	7.70	73
w05_100.9	-13068	-1752	0	-1449	17.29	95
w09_100.0	-23574	-2121	0	-1926	9.19	97
w09_100.1	-23584	-2096	0	-1941	7.39	110
w09_100.2	-24880	-2738	0	-2715	0.84	127
w09_100.3	-23284	-1990	0	-1753	11.90	93
w09_100.4	-23114	-2033	0	-1942	4.47	121
w09_100.5	-23932	-2433	0	-2318	4.72	152
w09_100.6	-23574	-2220	0	-1974	11.08	94
w09_100.7	-23588	-2252	0	-2133	5.28	98
w09_100.8	-22588	-1843	0	-1597	13.34	65
w09_100.9	-23136	-2043	0	-1773	13.21	82

Table 3 Results on Maximum Cut instances (part 1).

computing the Gilmore-Lawler bound are frequently near-optimal. Also for the quadratic assignment problem some of the results are of unexpected quality. Repeated application from different starting solutions arising at branch and bound subproblems, varying pivoting rules, or a continuation at degenerate solutions allow for a potential further improvement of the sustained primal bounds. As often only a few iterations are necessary, the proposed algorithm may be a worthwhile and decent extension to existing fast (linear) simplex implementations.

Instance	GLB	OPT	TUB	UB	Gap	nIter
ising2.5-100_5555	-6763984	-2460049	0	-2255008	8.33	43
ising2.5-100_6666	-5692994	-2031217	0	-1618939	20.29	32
ising2.5-100_7777	-8289108	-3363230	0	-2665859	20.73	43
ising3.0-100_5555	-6023188	-2448189	0	-2050616	16.23	46
ising3.0-100_6666	-4994192	-1984099	0	-1603031	19.20	31
ising3.0-100_7777	-7607142	-3335814	0	-2617134	21.54	42
ising2.5-150_5555	-11030002	-4363532	0	-3681682	15.62	79
ising2.5-150_6666	-10613166	-4057153	0	-3510997	13.46	64
ising2.5-150_7777	-10930924	-4243269	0	-3844107	9.40	77
ising3.0-150_5555	-9933270	-4279261	0	-3582038	16.29	75
ising3.0-150_6666	-9452712	-3949317	0	-3536593	10.45	66
ising3.0-150_7777	-9892884	-4211158	0	-3826998	9.12	77
ising2.5-200_5555	-16020758	-6294701	0	-5737340	8.85	105
ising2.5-200_6666	-16949312	-6795365	0	-5866134	13.67	100
ising2.5-200_7777	-14543784	-5568272	0	-4770166	14.33	95
ising3.0-200_5555	-14546106	-6215531	0	-5463613	12.09	90
ising3.0-200_6666	-15557560	-6756263	0	-5937254	12.12	98
ising3.0-200_7777	-13204608	-5560824	0	-4676010	15.91	92
ising2.5-250_5555	-19987568	-7919449	0	-6771724	14.49	131
ising2.5-250_6666	-18256236	-6925717	0	-5722132	17.37	114
ising2.5-250_7777	-17123808	-6596797	0	-5469504	17.08	115
ising3.0-250_5555	-18159360	-7823791	0	-6894210	11.88	127
ising3.0-250_6666	-16549550	-6903351	0	-5599899	18.88	110
ising3.0-250_7777	-15203308	-6418276	0	-5442786	15.19	120
ising2.5-300_5555	-22020298	-8579363	0	-7461514	13.02	123
ising2.5-300_6666	-22972450	-9102033	0	-7842520	13.83	133
ising2.5-300_7777	-21708690	-8323804	0	-7267723	12.68	142
ising3.0-300_5555	-19969384	-8493173	0	-7377370	13.13	122
ising3.0-300_6666	-20715532	-8915110	0	-7833814	12.12	136
ising3.0-300_7777	-19575008	-8242904	0	-7284165	11.63	130
t2g10_5555	-15104216	-6049461	1494808	-5147565	14.90	47
t2g10_6666	-14262308	-5757868	1293820	-5010666	12.97	45
t2g10_7777	-15390632	-6509837	2017820	-5570543	14.42	51
t2g15_5555	-36113996	-15051133	2893351	-12957103	13.91	126
t2g15_6666	-37758984	-15763716	2656590	-14229871	9.73	130
t2g15_7777	-36579956	-15269399	2872254	-13592918	10.97	127
t2g20_5555	-61189318	-24838942	8338738	-19606668	21.06	212
t2g20_6666	-70568028	-29290570	4417471	-25139029	14.17	236
t2g20_7777	-67639612	-28349398	6640458	-23134556	18.39	209
t3g5_5555	-29831860	-10933215	460797	-9745070	10.86	76
t3g5_6666	-31868890	-11582216	1271464	-10264905	11.37	74
t3g5_7777	-31587994	-11552046	873021	-10545927	8.70	74
t3g6_5555	-49527028	-17434469	2337267	-14683760	15.77	116
t3g6_6666	-55809704	-20217380	1482923	-17631134	12.79	117
t3g6_7777	-53552316	-19475011	890133	-16166386	16.98	119
t3g7_5555	-80295256	-28302918	2213824	-23671726	16.36	163
t3g7_6666	-90402124	-33611981	2591174	-28809161	14.28	199
t3g7_7777	-82520696	-29118445	3062622	-24699321	15.17	162

Table 4 Results on Maximum Cut instances (part 2).

References

1 E.W. Barankin and R. Dorfman. *On Quadratic Programming*, volume 2 of *Publications in statistics*. University of California Press, 1958.

2 E. M. L. Beale. On quadratic proramming. *Naval Research Logistics Quarterly*, 6(3):227–243, 1959.

3 Rainer E. Burkard, Stefan E. Karisch, and Franz Rendl. QAPLIB – a quadratic assignment problem library. *Journal of Global Optimization*, 10(4):391–403, Jun 1997.

4 Alberto Caprara. Constrained 0-1 quadratic programming: Basic approaches and extensions. *European Journal of Operational Research*, 187(3):1494–1503, 2008.

5 George B. Dantzig. Quadratic programming. a variant of the wolfe-markowitz algorithms. Technical report, Operations Research Center, University of California-Berkeley, 4 1961.

6 George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

7 B. Curtis Eaves. On quadratic programming. *Management Science*, 17(11):698–711, 1971.

8 Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

9 P. C. Gilmore. Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of the Society for Industrial and Applied Mathematics*, 10(2):305–313, 1962.

10 Fred Glover and Eugene Woolsey. Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Oper. Res.*, 22(1):180–182, Feb. 1974.

11 Edward L. Keller. The general quadratic optimization problem. *Mathematical Programming*, 5(1):311–337, Dec 1973.

12 Eugene L. Lawler. The quadratic assignment problem. *Management Science*, 9(4):586–599, 1963.

13 Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

14 Franz Rendl, Giovanni Rinaldi, and Angelika Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 121(2):307–335, 2010.

15 M. H. Rusin. A revised simplex method for quadratic programming. *SIAM Journal on Applied Mathematics*, 20(2):143–160, 1971.

16 C. M. Shetty. A simplified procedure for quadratic programming. *Operations Research*, 11(2):248–260, 1963.

17 C. van de Panne and Andrew Whinston. The simplex and the dual method for quadratic programming. *OR*, 15(4):355–388, 1964.

18 Philip Wolfe. The simplex method for quadratic programming. *Econometrica*, 27(3):382–398, 1959.