Inductive Linearization for Binary Quadratic Programs with Linear Constraints: A Computational Study

Sven Mallach

High Performance Computing & Analytics Lab
University of Bonn,
Friedrich-Hirzebruch-Allee 8, 53115 Bonn, Germany

June 11, 2021

Abstract

The computational performance of inductive linearizations for binary quadratic programs in combination with a mixed-integer programming solver is investigated for several combinatorial optimization problems and established benchmark instances. Apparently, a few of these are solved to optimality for the first time.

1 Introduction

Given a binary quadratic programs (BQP) comprising linear (and possibly quadratic) constraints, the inductive linearization technique [18] may serve as a computationally attractive compromise between the well-known "standard" linearization, and a complete application of the Reformulation Linearization Technique (RLT) [2]. In various relevant cases, inductive linearizations are more constraint-side compact than the "standard" linearization and provide a continuous relaxation that is at least as tight. Structural proofs of the latter property have been derived for some of these cases, e.g. when all the factors of products $x_i x_j$ are involved in at least one linear less-or-equal inequality or equation whose non-zero left hand coefficients and right hand side are equal to one [18]. Prominent combinatorial optimization problems where this applies are e.g. the Quadratic Assignment and the Quadratic Matching Problem. Further conditions for provably strong relaxations exist, covering for instance the Quadratic Traveling Salesman Problem.

In this light, the central contribution of this paper is a systematic computational study in order to address a number of research questions: Does the mentioned compactness without any tightness trade-off concerning the continuous relaxation also translate into a faster solution time of the corresponding mixed-integer programs? For which (further) problem structures is the inductive linearization technique (not) well-suited and why? Can an inductive linearization be obtained quickly in practice, and if so, how? To this end, we investigate the performance of the inductive as well as the "standard" linearization in combination with (and in comparison to) a professional mixed-integer programming solver on various BQPs with linear constraints. More precisely, we look at the Quadratic Assignment Problem, the Quadratic Knapsack Problem, the Quadratic Matching Problem, the Quadratic Shortest Path Problem, and on further instances of the well-established QPLIB[1] and MINLPLib[2]. In all conscience, a few instances from the latter libraries are solved to optimality for the first time using the proposed inductive linearization. Besides that, an algorithmic frame is presented to derive inductive linearizations in practice.

The outline of this paper is as follows: In the beginning of Sect. 2, we briefly review the inductive linearization technique, in order to strongly emphasize on its practical application in the subsequent subsections. In Sect. 3, we present the aforementioned applications along with the respective problem formulations as well as the benchmark instances used for the computational study, and a discussion of the respective results. Finally, a conclusion is formulated in Sect. 4.

2 Inductive Linearization

Let $n, m_L \in \mathbb{N} \setminus \{0\}$ and let $m_Q \in \mathbb{N}$. Suppose further we are given matrices $Q_k \in \mathbb{R}^{n \times n}$ for $k \in \{0, \ldots, m_Q\}$ and $A \in \mathbb{R}^{m_L \times n}$, vectors $h_k \in \mathbb{R}^n$ for $k \in \{1, \ldots, m_Q\}$ and $c \in \mathbb{R}^n$, and finally scalars $\beta_k \in \mathbb{R}$ for $k \in \{1, \ldots, m_Q\}$. Then the inductive linearization technique addresses the following general form of optimization problems:

$$\begin{aligned} \min\ & x^\top Q_0 x + c^\top x \\ \text{s.t.}\ & x^\top Q_k x + h_k^\top x \leq \beta_k \qquad k = 1, \ldots, m_Q \\ & Ax \leqq b \\ & x \in \{0,1\}^n \end{aligned}$$

Throughout this paper, we will denote by $N := \{1, \ldots, n\}$ the index set of the binary variables $x \in \{0,1\}^n$. The set of products P truly present in the problem is determined by the matrices $Q_k \in \mathbb{R}^{n \times n}$, $k \in \{0, \ldots, m_Q\}$, in the objective function and in the $m_Q \in \mathbb{N}_0$ quadratic restrictions as follows:

$$P := \{(i,j) \subseteq N \times N \mid \exists\, k \in \{0, \ldots, m_Q\} : Q_{k_{ij}} \neq 0\}$$

Naturally, we will assume that P is non-empty and, since we have for binary solutions that $x_i x_j = x_j x_i$ for $i, j \in N$, $i \neq j$, and $x_i = x_i^2$ for all $i \in N$, we may further assume the matrices Q_k, $k \in \{0, \ldots, m_Q\}$, to be strictly upper triangular, and thus $i < j$ for $(i,j) \in P$.

While quadratic constraints may or may not exist, some linear constraints on the binary variables are actually necessary to apply the inductive linearization technique. We assume w.l.o.g. that these are given as equations and less-or-equal inequalities, hence denoted $Ax \leqq b$. More precisely, the requirement is that each binary variable x_i, $i \in N$, being a factor of a product in P (to be linearized with the technique proposed), appears in at least one of these (with a non-zero coefficient). Clearly, this can be assumed (or established) for a binary problem without loss of generality. Indeed, if there is a factor x_i, $i \in N$, that is entirely free (unconstrained) in the original problem then in principle any equation or less-or-equal inequality may be employed that is valid for its feasible set (even, though rather not so desirable, $x_i \leq 1$). Moreover, if this requirement is not fulfilled for some factor, this does not affect a successful inductive linearization of all the products whose factors do. For simplicity, we thus assume from now on that all products in P are linearly constrained.

The inductive linearization technique is a generalization of a principle proposed by Liberti in [14] for the special case of equations with right hand side and left hand side coefficients equal to one, and of its later revision [16]. In his original article, Liberti coined the name "compact linearization" because it typically

[1] `https://qplib.zib.de`
[2] `https://www.minlplib.org`

adds fewer constraints to the mentioned problems than the "standard" linearization whose general form reads

$$\begin{aligned}
\min\ & d^\mathsf{T} y + c^\mathsf{T} x \\
\text{s.t. } & g_k^\mathsf{T} y + h_k^\mathsf{T} x && \leq \beta_k && k = 1,\dots,m_Q \\
& Ax && \leqq b && && (1) \\
& y_{ij} - x_i && \leq 0 && \{i,j\} \in P && (2) \\
& y_{ij} - x_j && \leq 0 && \{i,j\} \in P && (3) \\
& x_i + x_j - y_{ij} && \leq 1 && \{i,j\} \in P && (4) \\
& y && \geq 0 \\
& x && \in \{0,1\}^n
\end{aligned}$$

for our original problem. With $m := |P|$, we here use $y \in \mathbb{R}^m$ to denote the linearized products, $d \in \mathbb{R}^m$ to denote the quadratic objective coefficients, and $g_k \in \mathbb{R}^m$ to express the quadratic constraint coefficients for all $k = 1,\dots,m_Q$. Thereby, we use the subscript notation y_{ij} for $i < j$ and analogously define $d_{ij} = (Q_O)_{ij} + (Q_0)_{ji}$ and $g_{kij} = (Q_k)_{ij} + (Q_k)_{ji}$. Of course, if d_{ij} is less than zero, (4) can be omitted, and if d_{ij} is larger than zero, (2) and (3) can be omitted.

As we will see, in many cases, the generalized approach achieves constraint-side compactness as well. However, this cannot be guaranteed for any kind of BQP with linear constraints. Moreover, depending on how the method is applied, more than $|P|$ linearization variables may be induced (although this can, in principle, always be circumvented as described in Sect. 2.6). Therefore, and to have a clear distinction from other linearizations being called "compact", as well as to emphasize that the proposed method aims at "inducing" the products associated to the set P by multiplying original constraints with original variables, the technique is referred to as "inductive linearization" since its generalization first presented in [18].

2.1 Mathematical Derivation of Inductive Linearizations

Given a problem as introduced at the beginning of this section, suppose that we identify a working (sub-)set of the linear constraints $Ax \leqq b$ to actually induce the linearization with. Let us denote the index set of the selected equations and inequalities with K_E and K_I, respectively. That is, we consider the constraints

$$\sum_{i \in I_k} a_k^i x_i = b_k \qquad \text{for all } k \in K_E \qquad (5)$$

$$\sum_{i \in I_k} a_k^i x_i \leq b_k \qquad \text{for all } k \in K_I \qquad (6)$$

where $I_k := \{i \in N \mid a_k^i \neq 0\}$ denotes the respective support index set for each $k \in K_E$ or $k \in K_I$.

As already mentioned, we require w.l.o.g. a choice of $K := K_E \cup K_I$ such that there exist indices $k, \ell \in K$ with $i \in I_k$ and $j \in I_\ell$ for all $(i,j) \in P$. To refer to the respective constraints, we will use the notation $K(i) := \{k \in K : i \in I_k\}$, as well as $K_E(i)$ and $K_I(i)$ analogously defined if more preciseness is in order. Moreover, although we do not require this for the original problem, let us temporarily assume in addition that $b_k > 0$ for all $k \in K$ and $a_k^i > 0$ for all $i \in I_k$, $k \in K$. We will elaborate in Sect. 2.4 on how to handle constraints not fulfilling these prerequisites.

The first step of the inductive linearization approach now associates to each equation $k \in K_E$ another index set $M_k^E \subseteq N$ that is supposed to specify original variables used as multipliers. To each inequality $k \in K_I$, two such index sets $M_k^+, M_k^- \subseteq N$ are associated. The corresponding interpretation is as follows: If $j \in M_k^E$ ($j \in M_k^+$) the equation $k \in K_E$ (inequality $k \in K_I$) is multiplied by x_j, and if $j \in M_k^-$, the inequality $k \in K_I$ is multiplied by $(1 - x_j)$.

This leads to the following subset of the first level RLT constraints:

$$\sum_{i\in I_k} a_k^i x_i x_j = b_k x_j \qquad \text{for all } j \in M_k^E,\ k \in K_E \tag{7}$$

$$\sum_{i\in I_k} a_k^i x_i x_j \le b_k x_j \qquad \text{for all } j \in M_k^+,\ k \in K_I \tag{8}$$

$$\sum_{i\in I_k} a_k^i x_i (1-x_j) \le b_k (1-x_j) \qquad \text{for all } j \in M_k^-,\ k \in K_I \tag{9}$$

Let $M_k := M_k^E$ if $k \in K_E$, and $M_k := M_k^+ \cup M_k^-$ if $k \in K_I$. Then

$$Q := \{(i,j) \mid i \le j \text{ and } \exists k \in K : i \in I_k \text{ and } j \in M_k, \text{ or } j \in I_k \text{ and } i \in M_k\}$$

is the index set of the products induced by (7)–(9).

For ease of reference, we also define

$$M := \bigcup_{k\in K} M_k$$

which is to be regarded as a multiset of multiplier indices.

Remark 1. *The induced set Q may contain tuples that correspond to squares. Eliminating them is a simple and worthwhile optimization (see also Theorem 4 in Sect. 2.2). Recognition is already possible when squares are (or rather would be) induced: If x_j is used as a multiplier for a constraint with $j \in I_k$, the result may be instantly strengthened to*[3]*:*

$$\sum_{i\in I_k, i\neq j} a_k^i x_i x_j = (b_k - a_k^j) x_j \qquad \textit{for all } j \in M_k^E,\ k \in K_E$$

$$\sum_{i\in I_k, i\neq j} a_k^i x_i x_j \le (b_k - a_k^j) x_j \qquad \textit{for all } j \in M_k^+,\ k \in K_I$$

$$\sum_{i\in I_k, i\neq j} a_k^i x_i (1-x_j) \le b_k (1-x_j) \qquad \textit{for all } j \in M_k^-,\ k \in K_I$$

One has to ensure, however, that the coefficient of x_j is not turned zero by this operation. In this case the generated constraint would still be valid, but it would not link the products on its left hand side to their factor x_j as required (from the conditions described below).

If we now rewrite (7)–(9) by substituting for each $(i,j) \in Q$ the product $x_i x_j$ by a continuous linearization variable y_{ij} that has explicit lower and upper bounds, i.e., $0 \le y_{ij} \le 1$, we obtain the *linearization constraints*:

$$\sum_{i\in I_k,(i,j)\in Q} a_k^i y_{ij} + \sum_{i\in I_k,(j,i)\in Q} a_k^i y_{ji} = b_k x_j \qquad \text{for all } j \in M_k^E,\ k \in K_E \tag{10}$$

$$\sum_{i\in I_k,(i,j)\in Q} a_k^i y_{ij} + \sum_{i\in I_k,(j,i)\in Q} a_k^i y_{ji} \le b_k x_j \qquad \text{for all } j \in M_k^+,\ k \in K_I \tag{11}$$

$$\sum_{i\in I_k,(i,j)\in Q} a_k^i (x_i - y_{ij}) + \sum_{i\in I_k,(j,i)\in Q} a_k^i (x_i - y_{ji}) \le b_k (1-x_j) \qquad \text{for all } j \in M_k^-,\ k \in K_I \tag{12}$$

Now, as is expressed by the following theorem, for all the induced $(i,j) \in Q$ and binary x_i, x_j, one has $y_{ij} = x_i x_j$ if the following three consistency conditions are met:

Condition 1. *There is a $k \in K(i)$ such that $j \in M_k^E$ or $j \in M_k^+$, respectively.*

Condition 2. *There is a $k \in K(j)$ such that $i \in M_k^E$ or $i \in M_k^+$, respectively.*

[3] With respect to the final inequality, one readily observes that $a_k^j x_j (1-x_j) = 0$ for binary x_j.

Condition 3. *There is a* $k \in K(i)$ *such that* $j \in M_k^E$ *or* $j \in M_k^-$*, respectively,* **or** *a* $k \in K(j)$ *such that* $i \in M_k^E$ *or* $i \in M_k^-$*, respectively.*

Theorem 1. *([18]) For any integer solution* $x \in \{0,1\}^n$*, the linearization constraints (10)–(12) imply* $y_{ij} = x_i x_j$ *for all* $(i, j) \in Q$ *if and only if Conditions 1–3 are satisfied.*

So altogether, if we choose M consistently in terms of the conditions and such that Q contains P, we obtain a linearization for our original problem. In fact, at the potential expense of losing some continuous relaxation strength, it is always possible to have $Q = P$ as described in Sect. 2.6.

2.2 Linear Relaxation Strength of Inductive Linearizations

The linear programming relaxation obtained from an inductive linearization is provably at least as tight as the one obtained from the "standard" linearization if the constraints employed to satisfy the consistency conditions have only zero-one coefficients on the left, and a right hand side of one.

Theorem 2. *([18]) Consider a (sub-)set* K_E' *of equations (5) with* $b_k = 1$ *for all* $k \in K_E'$*, and* $a_k^i = 1$ *for each* $i \in I_k$*,* $k \in K_E'$*. Let* $Q' \subseteq Q$ *be the set of tuples induced by the multipliers* M_k^E*,* $k \in K_E'$*, and suppose that these multipliers satisfy the Conditions 1 and 2 for all* $(i, j) \in Q'$*. Then, for any* $0 \le x \le 1$*, we have* $y_{ij} \le x_i$*,* $y_{ij} \le x_j$ *and* $y_{ij} \ge x_i + x_j - 1$ *for all* $(i, j) \in Q'$*.*

Theorem 3. *([18]) Consider a (sub-)set* K_I' *of inequalities (6) with* $b_k = 1$ *for all* $k \in K_I'$*, and* $a_k^i = 1$ *for each* $i \in I_k$*,* $k \in K_I'$*. Let* $Q' \subseteq Q$ *be the set of tuples induced by the multipliers* M_k^+ *and* M_k^-*,* $k \in K_I'$*, and suppose that these multipliers satisfy the Conditions 1–3 for all* $(i, j) \in Q'$*. Then, for any* $0 \le x \le 1$*, we have* $y_{ij} \le x_i$*,* $y_{ij} \le x_j$ *and* $y_{ij} \ge x_i + x_j - 1$ *for all* $(i, j) \in Q'$*.*

A provably strong inductive linearization is also achieved for equations with a right hand side of two and left hand side coefficients one if these equations are multiplied by all variables on their left hand sides (i.e., $M_k = I_k$) and squares are ruled out as described in Remark 1.

Theorem 4. *([18]) Consider a (sub-)set* K_E' *of equations (5) with* $b_k = 2$ *for all* $k \in K_E'$*,* $a_k^i = 1$ *for each* $i \in I_k$*,* $k \in K_E'$*, and suppose that* $M_k^E = I_k$ *for all* $k \in K_E'$*. Let* $Q' \subseteq Q$ *be the set of tuples induced by these multipliers after eliminating squares. Then, for any* $0 \le x \le 1$*, we have* $y_{ij} \le x_i$*,* $y_{ij} \le x_j$ *and* $y_{ij} \ge x_i + x_j - 1$ *for all* $(i, j) \in Q'$*.*

Remark 2. *The referenced proofs of these theorems make apparent that the tightness of the relaxations of inductively linearized BQPs relates (besides other criteria) to the ratio between the right hand side and the left hand side coefficients. The referenced article also provides an example for a case where an inductive linearization provably has a* strictly *stronger continuous relaxation than the "standard" linearization.*

2.3 Practical Derivation of Inductive Linearizations

Given a problem formulation on sheet, a multiset M that induces a set $Q \supseteq P$ and establishes consistency can often be derived by inspection once the necessary implications of Conditions 1–3 are understood. This is likely in particular for combinatorial optimization problems. Nevertheless, especially if P is sparse, or if the formulation is complicated, it may be non-trivial to find a combination of constraints and multipliers that best suits compactness or other objectives. Moreover, an automated derivation is desirable especially for larger problem instances and allows for a linearization framework to be coupled with a mixed-integer programming solver.

Concerning the computation of an inductive linearization, it has been shown in [18] on the negative side, that the associated optimization problem (allowing e.g. to derive a linearization that is as compact as possible in terms of additional variables and constraints) is NP-hard in its general form. On the positive side, the problem can be solved well in practice, frequently even exactly, and there exist polynomial-time algorithms for specifically structured BQPs. Moreover, in the exact as well as in the heuristic case, computation times can be reduced by carefully preselecting the set K of original constraints considered for inductions. For many applications, the number of candidate constraints to induce a certain product, respectively to satisfy one of the three conditions, is anyway rather small.

We first review the mixed-integer program from [18] to model and solve the exact case:

$$
\begin{aligned}
\min \sum_{j \in N} \Bigg(\sum_{k \in K_E} w^E_{kj}\, z^E_{kj} + \sum_{k \in K_I} \Big(w^+_{kj}\, z^+_{kj} + w^-_{kj}\, z^-_{kj} \Big) \Bigg) + \Bigg(\sum_{i,j \in N, i \le j} w_{ij}\, f_{ij} \Bigg) & & & \\
\text{s.t. } f_{ij} &= 1 && \text{for all } (i,j) \in P && (13)\\
f_{ij} &\ge z^E_{kj} && \text{for all } k \in K_E, i \in I_k, j \in N, i \le j && (14)\\
f_{ji} &\ge z^E_{kj} && \text{for all } k \in K_E, i \in I_k, j \in N, j < i && (15)\\
f_{ij} &\ge z^+_{kj} && \text{for all } k \in K_I, i \in I_k, j \in N, i \le j && (16)\\
f_{ji} &\ge z^+_{kj} && \text{for all } k \in K_I, i \in I_k, j \in N, j < i && (17)\\
f_{ij} &\ge z^-_{kj} && \text{for all } k \in K_I, i \in I_k, j \in N, i \le j && (18)\\
f_{ji} &\ge z^-_{kj} && \text{for all } k \in K_I, i \in I_k, j \in N, j < i && (19)\\
\sum_{k \in K_E(i)} z^E_{kj} + \sum_{k \in K_I(i)} z^+_{kj} &\ge f_{ij} && \text{for all } i,j \in N, i \le j && (20)\\
\sum_{k \in K_E(j)} z^E_{ki} + \sum_{k \in K_I(j)} z^+_{ki} &\ge f_{ij} && \text{for all } i,j \in N, i \le j && (21)\\
\sum_{k \in K_E(j)} z^E_{ki} + \sum_{k \in K_I(j)} z^-_{ki} + & && && \\
\sum_{k \in K_E(i)} z^E_{kj} + \sum_{k \in K_I(i)} z^-_{kj} &\ge f_{ij} && \text{for all } i,j \in N, i \le j && (22)\\
f_{ij} &\in [0,1] && \text{for all } i,j \in N, i \le j && \\
z^E_{kj} &\in \{0,1\} && \text{for all } k \in K_E, j \in N && \\
z^+_{kj},\, z^-_{kj} &\in \{0,1\} && \text{for all } k \in K_I, j \in N &&
\end{aligned}
$$

Here, the binary variables z^E_{kj} are supposed to be equal to one if $j \in M^E_k$ for $k \in K_E$ and to be equal to zero otherwise. Similarly, the binary variables z^+_{kj} and z^-_{kj} are supposed to express whether or not $j \in M^+_k$ and $j \in M^-_k$ for $k \in K_I$, respectively. A further continuous variable f_{ij} for all $i,j \in N, i \le j$ accounts for whether $(i,j) \in Q$ in which case it will be equal to one whereas otherwise it will be equal to zero. Constraints (13) ensure that f_{ij} will be one if $(i,j) \in P$. Whenever a multiplier $j \in N$ is assigned to some set M_k, the corresponding products $(i,j) \in Q$ or $(j,i) \in Q$ for all $i \in I_k$ are induced by the inequalities (14)–(19). Finally, if $(i,j) \in Q$, then Conditions 1–3 are enforced by the inequalities (20)–(22), respectively.

It is apparent from this formulation (respectively, from the covering problem modeled by it) that the compactness of the resulting linearization is influenced by the support of the employed constraint set K and its relation to the factor pairs given by P. A possible choice of the objective function coefficients to obtain a "most compact" inductive linearization is described in [18]. Moreover, the mixed-integer program clearly simplifies significantly if only equations are considered, especially because (22) then becomes obsolete. If then in addition $|K(i)| = 1$ for each $i \in N$ holds, its constraint matrix becomes totally unimodular (cf. [16]), and the inductive linearization may be alternatively derived using a simple combinatorial algorithm described in the same reference.

Algorithm 1 is an extension of this algorithm to the general case and to incorporate "weights" in a similar way as the mixed-integer program does. It may serve as a heuristic to quickly derive inductive linearizations in practice that worked quite well in our experiments. In this context, it is also worth to mention that the presolve routines of a MIP solver may well eliminate some of the variables and constraints imposed by an inductive linearization that is not "most compact". Moreover, a few additional constraints may sometimes improve the relaxation strength, so compactness need not necessarily be an ultimate goal.

2.4 Normalization, Inductive Linearizations with General Linear Constraint Sets

We shall now discuss how to deal with the case that some (or all) of the original constraints $k \in K$ to be employed, i.e., (5) and (6), do not satisfy $b_k > 0$ and $a_k^i > 0$ for all $i \in I_k$. To this end, suppose that

$$\sum_{i \in I_k} a_k^i x_i = b_k$$

is an equation (the following also holds for a $\leq$-inequality) in K, and let $I_k^- \subseteq I_k$ be the set of variable indices such that $a_k^i < 0$ for each $i \in I_k^-$. For ease of notation, define also $I_k^+ = I_k \setminus I_k^-$.

The *explicit* approach to deal with such constraints is to define a new *complement* variable $\bar{x}_i$ for each $i \in I_k^-$, $k \in K$, along with the corresponding equation:

$$x_i + \bar{x}_i = 1 \tag{23}$$

Apparently, the equations (23) have only positive coefficients on the left hand side and a positive right hand side. Moreover, we may now replace any of the original equations with

$$\begin{aligned}
& \sum_{i \in I_k^+} a_k^i x_i + \sum_{i \in I_k^-} a_k^i (1 - \bar{x}_i) && = b_k \\
\Leftrightarrow & \sum_{i \in I_k^+} a_k^i x_i + \sum_{i \in I_k^-} a_k^i + \sum_{i \in I_k^-} -a_k^i \bar{x}_i && = b_k \\
\Leftrightarrow & \sum_{i \in I_k^+} a_k^i x_i + \sum_{i \in I_k^-} -a_k^i \bar{x}_i && = b_k + \sum_{i \in I_k^-} -a_k^i
\end{aligned}$$

where the term $-\sum_{i \in I_k^-} a_k^i$ on the left and on the right hand side is non-negative as well. We will also refer to the latter as the *normalized right hand side*.

Carrying out this procedure for every equation or inequality with negative coefficients on the left hand side clearly gives a system with only non-negative coefficients on the left. Now if any of the normalized right hand sides is negative, the system is obviously infeasible. Furthermore, if any of them is zero then all the variables on the respective left hand side can be fixed (original ones to zero, complemented ones to one) and thus be removed from the formulation.

As a consequence, it is possible to satisfy the prerequisites $b_k > 0$ for all $k \in K$ and $a_k^i \geq 0$ for all $i \in I_k$, $k \in K$, without loss of generality. A clear drawback of the explicit approach is however that it may add up to n variables and equations while the new variables become potential multipliers and the new equations are rather undesirable candidates for multiplications. Further, complementing variables "beforehand" may also incur avoidable (or only tediously removable) overhead if it is not a priori clear that the resulting normalized constraints will be at all employed for multiplications.

A more economical strategy is to keep the original constraints as they are, and to consider them for multiplication with each x_j (and $(1 - x_j)$ for case (9)) and each $\bar{x}_j$ (and $(1 - \bar{x}_j)$ for case (9)) without ever really introducing the complement variables. Instead, the idea of this *implicit* approach is to choose, for each $i \in I_k$, the "right" of the four possible combinations $x_i x_j$, $\bar{x}_i x_j$, $x_i \bar{x}_j$, and $\bar{x}_i \bar{x}_j$ to be induced, such that the respective linearization constraint imposes the necessary implications on their value.

Algorithm 1 A simple heuristic to construct an inductive linearization.

function CONSTRUCTSETS(Sets P, K)
    for all $k \in K_E$ **do**
        $M_k^E \leftarrow \emptyset$
    for all $k \in K_I$ **do**
        $M_k^+ \leftarrow \emptyset$; $M_k^- \leftarrow \emptyset$
    $Q \leftarrow P$
    $Q_{\text{new}} \leftarrow P$
    while $\emptyset \neq Q_{\text{add}} \leftarrow \text{APPEND}(Q, Q_{\text{new}}, K, M)$ **do**
        $Q \leftarrow Q \cup Q_{\text{add}}$
        $Q_{\text{new}} \leftarrow Q_{\text{add}}$
function APPEND(Sets Q, Q_{new}, K, M)
    $Q_{\text{add}} \leftarrow \emptyset$
    for all $(i, j) \in Q_{\text{new}}$ **do**
        if $j \notin M_k^E \; \forall\, k \in K_E(i) \wedge j \notin M_k^+ \; \forall\, k \in K_I(i)$ **then**
            $\bar{k} \leftarrow nil$; $\bar{w} \leftarrow 0$
            if $K_E(i) \neq \emptyset$ **then**
                $\bar{k} \leftarrow \arg\min_{k \in K_E(i)} w_{kj}^E$; $\bar{w} \leftarrow w_{\bar{k}j}^E$
            if $K_I(i) \neq \emptyset \wedge (\bar{k} = nil \vee \min_{k \in K_I(i)} w_{kj}^+ < \bar{w})$ **then**
                $\bar{k} \leftarrow \arg\min_{k \in K_I(i)} w_{kj}^+$
                $M_{\bar{k}}^+ \leftarrow M_{\bar{k}}^+ \cup \{j\}$
            else
                $M_{\bar{k}}^E \leftarrow M_{\bar{k}}^E \cup \{j\}$
            for all $a \in I_{\bar{k}}$ **do**
                if $a \leq j \wedge (a, j) \notin Q \cup Q_{\text{add}}$ **then** $Q_{\text{add}} \leftarrow Q_{\text{add}} \cup \{(a, j)\}$
                else if $a > j \wedge (j, a) \notin Q \cup Q_{\text{add}}$ **then** $Q_{\text{add}} \leftarrow Q_{\text{add}} \cup \{(j, a)\}$
        if $i \notin M_\ell^E \; \forall\, \ell \in K_E(j) \wedge i \notin M_\ell^+ \; \forall\, \ell \in K_I(j)$ **then**
            $\bar{\ell} \leftarrow nil$; $\bar{w} \leftarrow 0$
            if $K_E(j) \neq \emptyset$ **then**
                $\bar{\ell} \leftarrow \arg\min_{\ell \in K_E(j)} w_{\ell i}^E$; $\bar{w} \leftarrow w_{\bar{\ell} i}^E$
            if $K_I(j) \neq \emptyset \wedge (\bar{\ell} = nil \vee \min_{\ell \in K_I(j)} w_{\ell i}^+ < \bar{w})$ **then**
                $\bar{\ell} \leftarrow \arg\min_{\ell \in K_I(j)} w_{\ell i}^+$
                $M_{\bar{\ell}}^+ \leftarrow M_{\bar{\ell}}^+ \cup \{i\}$
            else
                $M_{\bar{\ell}}^E \leftarrow M_{\bar{\ell}}^E \cup \{i\}$
            for all $a \in I_{\bar{\ell}}$ **do**
                if $a \leq i \wedge (a, i) \notin Q \cup Q_{\text{add}}$ **then** $Q_{\text{add}} \leftarrow Q_{\text{add}} \cup \{(a, i)\}$
                else if $a > i \wedge (i, a) \notin Q \cup Q_{\text{add}}$ **then** $Q_{\text{add}} \leftarrow Q_{\text{add}} \cup \{(i, a)\}$
        if $j \notin M_k \; \forall\, k \in K_E(i) \wedge j \notin M_k^- \; \forall\, k \in K_I(i) \wedge i \notin M_\ell \; \forall\, \ell \in K_E(j) \wedge i \notin M_\ell^- \; \forall\, \ell \in K_I(j)$ **then**
            $\bar{k} \leftarrow nil$; $\bar{\ell} \leftarrow nil$; $\bar{w}_1 \leftarrow 0$; $\bar{w}_2 \leftarrow 0$
            if $K_E(i) \neq \emptyset$ **then**
                $\bar{k} \leftarrow \arg\min_{k \in K_E(i)} w_{kj}^E$; $\bar{w}_1 \leftarrow w_{\bar{k}j}^E$
            if $K_I(i) \neq \emptyset \wedge (\bar{k} = nil \vee \min_{k \in K_I(i)} w_{kj}^- < \bar{w}_1)$ **then**
                $\bar{k} \leftarrow \arg\min_{k \in K_I(i)} w_{kj}^-$; $\bar{w}_1 \leftarrow w_{\bar{k}j}^-$
            if $K_E(j) \neq \emptyset$ **then**
                $\bar{\ell} \leftarrow \arg\min_{\ell \in K_E(j)} w_{\ell i}^E$; $\bar{w}_2 \leftarrow w_{\bar{\ell} i}^E$
            if $K_I(j) \neq \emptyset \wedge (\bar{\ell} = nil \vee \min_{\ell \in K_I(j)} w_{\ell i}^- < \bar{w}_2)$ **then**
                $\bar{\ell} \leftarrow \arg\min_{\ell \in K_I(j)} w_{\ell i}^-$; $\bar{w}_2 \leftarrow w_{\bar{\ell} i}^-$
            if $\bar{w}_1 \leq \bar{w}_2$ **then**
                if $\bar{k} \in K_E(i)$ **then**
                    $M_{\bar{k}}^E \leftarrow M_{\bar{k}}^E \cup \{j\}$
                else
                    $M_{\bar{k}}^- \leftarrow M_{\bar{k}}^- \cup \{j\}$
                for all $a \in I_{\bar{k}}$ **do**
                    if $a \leq j \wedge (a, j) \notin Q \cup Q_{\text{add}}$ **then** $Q_{\text{add}} \leftarrow Q_{\text{add}} \cup \{(a, j)\}$
                    else if $a > j \wedge (j, a) \notin Q \cup Q_{\text{add}}$ **then** $Q_{\text{add}} \leftarrow Q_{\text{add}} \cup \{(j, a)\}$
            else
                if $\bar{\ell} \in K_E(j)$ **then**
                    $M_{\bar{\ell}}^E \leftarrow M_{\bar{\ell}}^E \cup \{i\}$
                else
                    $M_{\bar{\ell}}^- \leftarrow M_{\bar{\ell}}^- \cup \{i\}$
                for all $a \in I_{\bar{\ell}}$ **do**
                    if $a \leq i \wedge (a, i) \notin Q \cup Q_{\text{add}}$ **then** $Q_{\text{add}} \leftarrow Q_{\text{add}} \cup \{(a, i)\}$
                    else if $a > i \wedge (i, a) \notin Q \cup Q_{\text{add}}$ **then** $Q_{\text{add}} \leftarrow Q_{\text{add}} \cup \{(i, a)\}$
    return Q_{add}

More precisely, as can be verified from the proof of Theorem 4 in [18], the inequalities (8) (respectively their linearized counterparts (11)) have the effect of enforcing all the products (respectively, linearization variables) on the left hand side to be zero if the multiplier x_j is zero. Similarly, the constraints (9) (respectively, (12)) enforce any $x_i x_j$ (y_{ij}) on the left hand side to coincide with x_i if x_j is one. The equations (7) respectively (10) even directly impose both relationships at once. These implications are exactly what is established by Conditions 1–3 if the coefficients and right hand sides of the original constraints are non-negative respectively positive.

To achieve the same in the general case, one has to replace (8) by

$$
\begin{aligned}
&\sum_{i\in I_k^+} a_k^i x_i x_j + \sum_{i\in I_k^-} a_k^i (1-\bar{x}_i) x_j && \leq b_k x_j \\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i x_i x_j + \sum_{i\in I_k^-} a_k^i (x_j - \bar{x}_i x_j) && \leq b_k x_j \\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i x_i x_j + \sum_{i\in I_k^-} -a_k^i \bar{x}_i x_j && \leq (b_k + \sum_{i\in I_k^-} -a_k^i) x_j
\end{aligned}
$$

respectively by

$$
\begin{aligned}
&\sum_{i\in I_k^+} a_k^i x_i \bar{x}_j + \sum_{i\in I_k^-} a_k^i (1-\bar{x}_i) \bar{x}_j && \leq b_k \bar{x}_j \\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i x_i \bar{x}_j + \sum_{i\in I_k^-} a_k^i (\bar{x}_j - \bar{x}_i \bar{x}_j) && \leq b_k \bar{x}_j \\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i x_i \bar{x}_j + \sum_{i\in I_k^-} -a_k^i \bar{x}_i \bar{x}_j && \leq (b_k + \sum_{i\in I_k^-} -a_k^i) \bar{x}_j \\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i x_i \bar{x}_j + \sum_{i\in I_k^-} -a_k^i \bar{x}_i \bar{x}_j && \leq (b_k + \sum_{i\in I_k^-} -a_k^i)(1-x_j) \\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i x_i \bar{x}_j + \sum_{i\in I_k^-} -a_k^i \bar{x}_i \bar{x}_j && \leq (b_k + \sum_{i\in I_k^-} -a_k^i) - (b_k + \sum_{i\in I_k^-} -a_k^i) x_j.
\end{aligned}
$$

The equations (7) are to be replaced analogously. It is easy to see that, as desired, the linearization variables to be substituted for the products are forced to zero whenever the multiplier x_j respectively $\bar{x}_j$ is zero, and in particular that this effect is preserved when re-substituting the conceptual $\bar{x}_j$ by $(1-x_j)$ in *linear* terms.

Further, the inequalities (9) need to be replaced by

$$
\begin{aligned}
&\sum_{i\in I_k^+} a_k^i x_i (1-x_j) + \sum_{i\in I_k^-} a_k^i (1-\bar{x}_i)(1-x_j) && \leq b_k (1-x_j) \\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i (x_i - x_i x_j) + \sum_{i\in I_k^-} a_k^i ((1-\bar{x}_i) - (1-\bar{x}_i) x_j) && \leq b_k (1-x_j) \\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i (x_i - x_i x_j) + \sum_{i\in I_k^-} a_k^i (x_i - x_j + \bar{x}_i x_j) && \leq b_k - b_k x_j \\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i (x_i - x_i x_j) + \sum_{i\in I_k^-} a_k^i (x_i + \bar{x}_i x_j) && \leq b_k - (b_k + \sum_{i\in I_k^-} -a_k^i) x_j
\end{aligned}
$$

respectively by

$$\begin{aligned}
&\sum_{i\in I_k^+} a_k^i x_i(1-\bar{x}_j) + \sum_{i\in I_k^-} (a_k^i(1-\bar{x}_i))(1-\bar{x}_j) && \le b_k(1-\bar{x}_j)\\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i (x_i - x_i\bar{x}_j) + \sum_{i\in I_k^-} a_k^i((1-\bar{x}_i)-(1-\bar{x}_i)\bar{x}_j) && \le b_k x_j\\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i (x_i - x_i\bar{x}_j) + \sum_{i\in I_k^-} a_k^i(x_i-(\bar{x}_j-\bar{x}_i\bar{x}_j)) && \le b_k x_j\\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i (x_i - x_i\bar{x}_j) + \sum_{i\in I_k^-} a_k^i(x_i-((1-x_j)-\bar{x}_i\bar{x}_j)) && \le b_k x_j\\
\Leftrightarrow &\sum_{i\in I_k^+} a_k^i (x_i - x_i\bar{x}_j) + \sum_{i\in I_k^-} a_k^i(x_i+\bar{x}_i\bar{x}_j) && \le (b_k + \sum_{i\in I_k^-} -a_k^i)x_j + \sum_{i\in I_k^-} a_k^i.
\end{aligned}$$

Here, one may again verify that, if x_j respectively $\bar{x}_j$ is equal to one, then the inequalities enforce the linearization variables to be substituted for the products to equal x_i if $i \in I_k^+$ and to equal $(1-x_i)$ if $i \in I_k^-$, just as desired. Again, this effect is preserved when re-substituting the conceptual $\bar{x}_i$ or $\bar{x}_j$ by respectively $(1-x_i)$ and $(1-x_j)$ in linear expressions.

As a result, we achieved that complement variables and the associated equations need not be introduced. The handling of negative coefficients becomes even almost oblivious as it certainly does not matter whether to introduce a linearization variable for $x_i x_j$ or for any of $\bar{x}_i x_j$, $x_i\bar{x}_j$, and $\bar{x}_i\bar{x}_j$ instead. Moreover, if $(i, j) \in P$ but $x_i x_j$ is not induced, quadratic constraints and terms in the objective function referring to this product can still be expressed using the following equations for (implicit) substitutions[4]:

$$\begin{aligned}
x_i x_j &= x_j - \bar{x}_i x_j & \Leftrightarrow \quad \bar{x}_i x_j &= x_j - x_i x_j\\
x_i x_j &= x_i - x_i\bar{x}_j & \Leftrightarrow \quad x_i\bar{x}_j &= x_i - x_i x_j\\
x_i x_j &= \bar{x}_i\bar{x}_j + x_i + x_j - 1 & \Leftrightarrow \quad \bar{x}_i\bar{x}_j &= x_i x_j - x_i - x_j + 1
\end{aligned}$$

Nonetheless, even with the implicit normalization approach, negative coefficients are still not *entirely* free of charge, as it may be (like also in the explicit approach) that not just one but several of the four possible products need to be induced for $i, j \in N$, $i < j$. Moreover, negative coefficients increase the complexity of the mixed-integer program from Sect. 2.3 (besides that, for the implicit approach equations (13) need to be relaxed to the enforcement of *at least* one of the four combinations) as well as the complexity of implementing Algorithm 1.

2.5 Optional Preparations: Equation Splitting and Replication

Depending on the problem structure, one might consider to split equations into two inequalities before starting to derive an inductive linearization (see also the next subsection). Moreover, if an equation has positive and negative coefficients on the left hand side, one may consider both the original and the equation after multiplication with -1 as candidates to create linearization constraints (and induce linearization variables).

2.6 Postprocessing: Variable Elimination by Possible Constraint Weakening

It is possible to eliminate (all) variables in $Q \setminus P$ as a postprocessing step if (all) of these have been generated from inequalities. As is clear from inequalities (11) and (12) as well as from the discussion in Sect. 2.4, removing summands on their left hand sides will neither harm their validity nor their necessary implications

[4] Observe however that it is not possible (though tempting) to use the indicated reverse substitutions also for the derived linearization constraints as this would actually also reverse the construction and thus destroy the established consistency implications in the presence of negative original coefficients. It is the linearization constraints which ensure validity of the substitutions in arrears, not vice versa.

on the respective remaining linearization variables. In general, however, the feasible region of the continuous relaxation may of course be enlarged by this procedure.

3 A Preliminary Computational Study

Some evidence for a computational utility of inductive linearizations for specific applications is already available in the literature (e.g. [7, 14, 15, 16, 17]). Here, our aim is a systematic study and a structured overview of the computational performance on various well-known and *well-suited as well as not so well-suited* binary quadratic programs with linear constraints. To this end, we identified a number of prominent combinatorial optimization problems and benchmark instances commonly used by the community in order to evaluate inductive linearizations in comparison with "standard" linearizations. Concerning the former, we distinguish a usual inductive linearization (IL) and, if inequalities are present, a weakened inductive linearization (ILW) which employs the postprocessing from Sect. 2.6. We further distinguish the complete "standard" linearization (SLC) causing $|P|$ additional variables, $3|P|$ additional inequalities, and $7|P|$ additional non-zeros, and the reduced "standard" linearization (SLR) which only contains those of the inequalities (2)–(4) that are required due to the objective coefficients.

While specific details are given in the respective subsections, the following paragraphs subsume those parts of the setting that apply to all the experiments.

As a preprocessing, all linear greater-or-equal inequalities were turned into less-or-equal ones, and if a left hand side has only integer coefficients, fractional right hand sides were rounded down.

To derive the inductive linearizations, we employed Algorithm 1 with implicit normalization as described in Sect. 2.4. In each iteration of the function `Append`, the weights of the constraint-multiplier combinations $w^E_{k,j}$ and $w^+_{k,j}$ ($w^-_{k,j}$) are recomputed as the negated number of products in Q_{add} for which Conditions 1, 2 or 3 would be satisfied if constraint k was multiplied with x_j $(1-x_j)$. Absolute coefficients or right hand sides as well as the number of non-zero coefficients are however not taken into account. Due to the order of looking up constraints being candidates for multiplications, a slight implicit preference of equations over inequalities is inherent to the implementation. The table columns titled "Get [s]" in the following subsections will display the running time of this algorithm (i.e., the wall clock time to derive the respective inductive linearization) in seconds. As the results eventually show, they are adequate with only very few exceptions. Nevertheless, it is just a simple prototype implementation that could be tailored and optimized in several ways.

In order to finally solve the resulting MIPs, we employed Gurobi[5] in version 9.03 with its seed parameter set to one. All computations were carried out using a single thread on a Debian Linux system equipped with an Intel Xeon E5-2690 CPU (3 GHz) and 128 GB RAM. Each run had a time limit of 48 hours. If it was exceeded, this is indicated by "–" in the respective table column. Apart from that, we would like to emphasize that the displayed (wall clock) running times should only be considered as an indicator for which kind of problems the respective methods appear particularly suited or rather not suited, especially as the absolute running times for each single method may vary significantly depending on various influences (as e.g. different seeds, parameters, branching choices, solver versions) whose possible combinations would justify a computational study on their own. Further, the results in this preprint are also preliminary in the sense that access to the mentioned system was not always exclusive to a single experiment.

[5] `https://www.gurobi.com/`

3.1 The Quadratic Assignment Problem

Given $T, D \in \mathbb{R}^{n\times n}$ and $c \in \mathbb{R}^n$, a quadratic assignment problem (QAP) in the form by Koopmans and Beckmann [13] can be written as follows.

$$
\begin{aligned}
\min \sum_{i=1}^{n}\sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{q=1}^{n} t_{ij}d_{pq}\, x_{ip}x_{jq} &+ \sum_{i=1}^{n}\sum_{p=1}^{n} c_{ip}x_{ip} \\
\text{s.t. } \sum_{i=1}^{n} x_{ip} &= 1 && \text{for all } p \in \{1,\dots,n\} && (24)\\
\sum_{p=1}^{n} x_{ip} &= 1 && \text{for all } i \in \{1,\dots,n\} && (25)\\
x_{ip} &\geq 0 && \text{for all } i,p \in \{1,\dots,n\}\\
x_{ip} &\in \mathbb{Z} && \text{for all } i,p \in \{1,\dots,n\}
\end{aligned}
$$

Frieze and Yadegar [8] derived a linearization of the QAP that is actually an inductive one, but that is not yet most compact. To characterize a most compact inductive linearization for the case where all (meaningfully) possible products are of interest, observe first that each of the variables $X := \{x_{ip} \mid i,p \in \{1,\dots,n\}\}$ occurs exactly once in the equation set (24) and exactly once in the equation set (25). Thus, in order to induce all products and to satisfy Conditions 1 and 2 for them, it would suffice to *either* multiply all of the constraints (24) with X, *or* to multiply all of the constraints (25) with X. Moreover, since objective coefficients for x_{ip}^2 may be moved to the linear part, and since the variables y_{ipiq} for all $p,q \in \{1,\dots,n\}$ as well as all variables y_{ipjp} for all $i,j \in \{1,\dots,n\}$ can be eliminated, it suffices to formulate the resulting constraints only for $i \neq j$, or $p \neq q$, respectively. If one further identifies y_{jqip} with y_{ipjq} whenever $i < j$, a most compact inductive linearization of all meaningfully possible products is:

$$
\begin{aligned}
\min \sum_{i=1}^{n}\sum_{p=1}^{n}\sum_{j=i+1}^{n}\sum_{p\neq q=1}^{n} (t_{ij}d_{pq} + t_{ji}d_{qp})y_{ipjq} &+ \sum_{i=1}^{n}\sum_{p=1}^{n} (c_{ip} + t_{ii}d_{pp})x_{ip} \\
\text{s.t. } \sum_{i=1}^{n} x_{ip} &= 1 && \text{for all } p \in \{1,\dots,n\}\\
\sum_{p=1}^{n} x_{ip} &= 1 && \text{for all } i \in \{1,\dots,n\}\\
\sum_{i=1}^{j-1} y_{ipjq} + \sum_{i=j+1}^{n} y_{jqip} &= x_{jq} && \text{for all } p,j,q \in \{1,\dots,n\}, p \neq q && (26)\\
y_{ipjq} &\in [0,1] && \text{for all } i,j,p,q \in \{1,\dots,n\}, i < j, p \neq q\\
x_{ip} &\in \{0,1\} && \text{for all } i,p \in \{1,\dots,n\}
\end{aligned}
$$

In this formulation, (26) could also be replaced by the constraints

$$
\begin{aligned}
\sum_{p=1}^{q-1} y_{ipjq} + \sum_{p=q+1}^{n} y_{ipjq} &= x_{jq} && \text{for all } i,j,q \in \{1,\dots,n\}, i < j\\
\sum_{p=1}^{q-1} y_{jqip} + \sum_{p=q+1}^{n} y_{jqip} &= x_{jq} && \text{for all } i,j,q \in \{1,\dots,n\}, j < i
\end{aligned}
$$

which resembles once more the original freedom to choose one of (24) and (25) as the basis for inductions.

The total number of additional equations thus amounts to only $n^3 - n^2$ instead of $3 \cdot \left(\frac{1}{2}(n^2-n)(n^2-n)\right) = \frac{3}{2}(n^4 - 2n^3 + n^2)$ inequalities when using the complete "standard" linearization and creating y_{ipjq} only for

$i < j$ and $p \neq q$ as well. However, these most compact formulations have a weaker linear programming relaxation than the ones by Frieze and Yadegar, and Adams and Johnson, that comprise more constraints. If P does not contain all meaningfully possible products, it is clear that once more only a subset of the possible linearization constraints is required.

For our experiments, we employed the established QAPLIB ([6]) instances. In Table 1, we list those of them that could be solved within 48 hours using either IL or one of the standard linearizations SLC and SLR. For each instance, the corresponding value of n is part of the instance name. The SLR has of course $|P|$ additional variables, and – as all the listed instances have positive objective coefficients only – $|P|$ additional inequalities, and $3|P|$ additional non-zeros (column "NZ+"). For IL, we display the number of linearization variables, equations, and the associated number of additional non-zeros in the constraint matrix. The task to find a good combination satisfying Conditions 1 and 2 for all induced products could be solved very quickly with Algorithm 1. In some further experiments, the mixed-integer program from Sect. 2.3 could also be solved by Gurobi within the root of its branch-and-bound tree.

The results obtained using IL are clearly better than with the two "standard" linearizations while of course still not competitive to state-of-the-art methods for the QAP. Nevertheless, it is apparent that most QAPLIB instances with up to about 20000 products can be handled within the time limit. The tendential superiority compared to the "standard linearization" is also in line with the results that have been obtained earlier for other optimization problems with "(semi-)assignment" (or rather "one-out-of-many selection") constraints. Some examples are graph partitioning ([16]), multiprocessor scheduling ([15]), graph layering ([17]), and quadratic semi-assignment problems ([4]). Nevertheless, additional constraint sets and the structure of the objective coefficients may always impact the outcome significantly as will also become apparent in Sections 3.5 and 3.6.

3.2 The Quadratic 0-1 Knapsack Problem

The possibly simplest inequality-only application for the inductive linearization technique is the quadratic 0-1 knapsack problem (QKP). It is particularly interesting for the computational study because here the left hand side coefficients (item sizes) and the right hand sides (knapsack capacity) vary considerably and can also be large.

Starting from the canonical formulation with a capacity $b \in \mathbb{R}$, and a variable x_j for each item j of a ground set J with size $a_j \in \mathbb{R}$,

$$\begin{aligned} \max \sum_{i,j \in J, i<j} q_{ij} x_i x_j + \sum_{i \in J} c_i x_i & & \\ \text{s.t.} \sum_{i \in J} a_i x_i & \leq b & \\ x_i & \in \{0,1\} & \text{for all } i \in J, \end{aligned} \tag{27}$$

it has been observed in the literature that inequalities of type (8) could be used in combination with the "standard" linearization (see e.g. [3]), and also inequalities of type (9) have been applied in the context of semidefinite relaxations to improve the obtained dual bounds ([11]). A corresponding square-reduced (cf. Remark 1) inductive linearization is the following mixed-integer program:

$$\begin{aligned}
\max \sum_{i,j\in J, i<j} q_{ij}y_{ij} + \sum_{i\in J} c_i x_i & & \\
\text{s.t.} \sum_{i\in J} a_i x_i &\leq b & \\
\sum_{i\in J, i\neq j} a_i y_{ij} &\leq (b-a_j)x_j & \text{for all } j\in J\\
\sum_{i\in J, i\neq j} a_i (x_i - y_{ij}) &\leq b(1-x_j) & \text{for all } j\in J\\
y_{ij} &\in [0,1] & \text{for all } i,j\in J, i<j\\
x_i &\in \{0,1\} & \text{for all } i\in J
\end{aligned}$$

Once more, the formulation is more compact than a "standard" linearization, especially if all possible products are of interest: Assuming $|J| = n$, the $\binom{n}{2}$ products are then linearized using only $2n-1$ ($n-1$ inequalities of type (9) suffice to satisfy Condition 3 for all of them while n of (8) are needed for Conditions 1 and 2) instead of $3\binom{n}{2}$ inequalities. However, in the general case of arbitrary a_j, $j \in J$, and b, an implication of the "standard" linearization inequalities (2)–(4) cannot be expected.

Moreover and irrespective of the cardinality of P, since there is only one original inequality, the satisfaction of Conditions 1–3 must be established using (27) for each product induced, and it is hence entirely predetermined which products will be induced. It is thus not surprising that Algorithm 1 as well as the MIP solver could derive the unique solution quickly in our experiments.

For these, we employed the randomly generated instances by [5]. The results are displayed in Tables 2 and 3. The respective number of items is given by the first number in the instance name. The presentation is restricted to ILW that clearly outperformed IL. The SLR has here $|P|$ additional variables, $2|P|$ additional inequalities, and $4|P|$ additional non-zeros as all listed instances have positive objective coefficients only, and the objective is to be maximized. In case of `jeu_200_100_5` we needed to change Gurobi's `MIPGap` parameter to 10^{-6} (while the default used instead is 10^{-4}) in order to retrieve an optimum solution with the SLC.

As could be expected from the large constraint coefficients and right hand sides, the performance of the MIP solver using the inductive linearizations and the bounds of their root relaxations are here significantly weaker than with the "standard" linearization. Moreover, while the numbers of linearization constraints are much smaller for the former, the converse is true for the induced numbers of non-zeros. We observe that there is a considerable variation in the solution times even within each size and product density group. For the larger instances, only those with a comparably small capacity remain solvable using ILW. As the left hand side coefficient ranges do (almost) not vary, this once more underlines the inductive linearization's sensitivity to the "ratio" between the right hand side and the left hand side coefficients.

3.3 The Quadratic Matching Problem

Given an undirected graph $G = (V,E)$, a canonical BQP that models the quadratic matching problem (QMP) on G (see e.g. [12]) can be expressed as

$$\begin{aligned}
\max \sum_{e,f\in E, e\neq f} q_{ef}x_e x_f + \sum_{e\in E} c_e x_e & & \\
\text{s.t.} \sum_{e\in\delta(i)} x_e &\leq 1 & \text{for all } i\in V \qquad (28)\\
x_i &\in \{0,1\} & \text{for all } i\in E,
\end{aligned}$$

where $\delta(i) := \{\{i,j\}\in E : j\in V\setminus\{i\}\}$ for $i\in V$.

It is easily observed that each edge (factor) occurs in exactly two constraints, namely those inequalities (28) associated with its endpoints. In this sense, the situation is similar as in case of the QAP. However, the constraints need not be as regular as they depend on the structure of G, and the mixed-integer programs to compute the inductive linearizations turned out to be more difficult for the MIP solver employed. On the contrary, the heuristic delivered good linearizations quickly.

For our experiments, we employed the instances used by [12]. The presentation in the Tables 4, 5, and 6 is restricted to those instances called "BM" in this reference, as the other ones with less variables or products could be solved very quickly using all linearization approaches, and the other instances of about the same size produced quite similar results as has been the case also in the reference. In particular, nearly all instances could be solved within 48 hours by all methods, only in three exceptions this was not the case with SLR. The exact number of additional inequalities and non-zeros for SLR here vary with the (objective coefficients in the) different instances which is why they are displayed in the tables.

From a coarse perspective, Gurobi could solve the problems better using the inductive linearizations than when using the standard linearizations. Taking a closer look, the ILW led to the best running time in most of the cases.

3.4 The Quadratic Shortest Path Problem

Given a directed graph $G = (V, A)$, a source node $s \in V$ and a target node $t \in V$, a canonical BQP that models the quadratic shortest s-t-path problem (QSPP) on G (see e.g. [19]) can be expressed as

$$\begin{aligned} \min \; & \sum_{a,b \in A, a \neq b} q_{ab} x_a x_b + \sum_{a \in A} c_a x_a \\ \text{s.t.} \; & \sum_{a \in \delta^+(i)} x_a - \sum_{a \in \delta^-(i)} x_a = b(i) && \text{for all } i \in V \\ & x_a \in \{0,1\} && \text{for all } a \in A, \end{aligned} \tag{29}$$

where $\delta(i)^+ := \{(i,j) \in A, j \in V \setminus \{i\}\}$, $\delta(i)^- := \{(j,i) \in A, j \in V \setminus \{i\}\}$, $b(i) = 0$ for all $i \in V \setminus \{s,t\}$, $b(s) = 1$, and $b(t) = -1$.

One can see immediately that constraints (29) have -1 and 1 coefficients on their left hand sides, requiring normalization. As it has been the case with the QMP, each potential factor (arc) occurs on the left hand sides of two constraints (those associated with its endpoints), but now with opposite signs. This impacts the MIP solution times to derive an inductive linearization negatively while this is not the case for Algorithm 1. Not surprisingly, for the the largest instances with $|P| > 10^6$, a noticeable increase in the derivation time takes place nevertheless.

For the experiments, we generated instances as described by [19], three for each type and size. However, even though ruling out some products that are implied to be zero by the problem structure, the number of products remained relatively large and only a few instances could be solved within an 48 hour time limit when passing the inductive linearization to the MIP solver. As one can see from Table 7, the standard linearization clearly works better with Gurobi for these instances. The SLR has $|P|$ additional variables, $|P|$ additional inequalities, and $3|P|$ additional non-zeros as the instances have positive objective coefficients only. Using IL, there are less linearization constraints but the number of induced products and non-zeros is significantly larger. We found that we can further reduce the number of linearization constraints and non-zeros for IL by a factor of about three when investing more time to derive the inductive linearization using the MIP. For the `grid1_10` instances, this led to a faster solution of the resulting MIP by a factor of about six. On the large `grid2_16_16` instances, the size reduction was the same but the solution times were less than halved only. Hence, there is evidently room for improvement but it currently appears to be too challenging to compete with the "standard" linearization and especially specialized methods for the QSPP as in [19] that work even much better.

3.5 MINLPLib

As another benchmark and to further broaden the experimental setting, we incorporated library collections such as the MINLPLib providing also instance formats that we could pass on directly to Gurobi (we used the `.lp` format) in order to have an additional comparison (table columns "GRB"). More precisely, we identified all linearly constrained BQPs except for six QSPP instances and `qap.lp` (as similar instances have been covered above). We remark that all except ten of these instances have equations only. The first two exceptions are `crossdock_15x7` which has 30 equations and 14 inequalities, and `crossdock_15x8` which has 30 equations and 16 inequalities. Also for these two instances, however, only equations are employed to derive the inductive linearizations. Finally, there are the `pbXXYYZZ` instances which have `XX` equations and `YY` inequalities, and only equations are employed for the inductive linearizations as well. Especially, partly when investing some more computational resources and time, the following of these instances in Table 8 were solved using IL – to the best of our knowledge for the first time.

Instance	Optimal Value
pb302035	3379359
pb302095	5710645
pb351535	4456670

Table 8: Optimal solution values for some apparently previously unsolved MINLPLib instances.

As opposed to that, the instances in Table 9 have been solved before (respectively `celar6-sub0` is solved as well by Gurobi as a standalone solver), but still confirmations of the claimed optima found are demanded whence we list the values that we retrieved.

celar6-sub0	159
crossdock_15x7	14409
crossdock_15x8	15595
color_lab3_3x0	79.93
maxcsp-ehi-85-297-12	9
maxcsp-ehi-85-297-36	9
maxcsp-ehi-85-297-71	9
maxcsp-ehi-90-315-70	9
maxcsp-langford-3-11	2

Table 9: Optimal solution values for some MINLPLib instances.

Tables 10 and 11 display the entire results. A first observation is that even among those instances with only equations and left hand side coefficients as well as right hand sides equal to one, the inductive as well as the "standard" linearization(s) can be clearly superior or inferior. For instance, for `celar6-sub0`, the inductive linearization is orders of magnitude faster even though there are many products and induced non-zeros, and e.g. for `color_lab3_3x0` the converse is true. Similar extremes are also identified when comparing with Gurobi as a standalone solver. Although there is a more comprehensive study in [16], the graph partitioning problems were not sorted out from the experiments for two reasons. Firstly, these instances differ from those in the reference in that they lack an additional normalization constraint, and there are a few outliers in the results where IL turns out to be not as robust despite the typically good results for optimization problems with similar constraints as mentioned already in Sect. 3.1. Secondly, it becomes apparent that the `graphpart-clique`-instances become especially challenging with increasing size. In these instances, each node must be assigned to one of three partitions and highly symmetric costs are associated to placing any pair of nodes in the same partition. On the maximum constraint satisfiability problems, the SLR performs clearly best with Gurobi.

		SLC	SLR			IL		
Instance	$\lvert P \rvert$	Solve [s]	Solve [s]	$\lvert Q \rvert$	$\lvert M \rvert$	NZ+	Solve [s]	Get [s]
chr12a	1430	6.63	8.13	1584	264	3432	0.37	0.01
chr12b	1430	5.59	3.69	1584	264	3432	0.40	0.01
chr12c	1430	36.14	54.65	1584	264	3432	0.50	0.01
chr15a	2940	187.01	410.09	3150	420	6720	0.86	0.04
chr15b	2940	24.41	23.41	3150	420	6720	0.93	0.04
chr15c	2940	537.12	521.41	3150	420	6720	0.07	0.04
chr18a	5202	8107.68	3862.78	5508	612	11628	1.85	0.08
chr18b	5202	–	–	5508	612	11628	0.76	0.08
chr20a	7220	13174.46	12398.96	7600	760	15960	5.73	0.13
chr20b	7220	–	–	7600	760	15960	8.65	0.10
chr20c	7220	236.81	392.37	7600	760	15960	8.89	0.13
chr22a	9702	–	–	10164	924	21252	3.62	0.16
chr22b	9702	–	–	10164	924	21252	3.34	0.18
chr25a	14400	–	–	15000	1200	31200	32.42	0.32
els19	19152	430.99	356.64	20216	2128	42560	21.11	0.14
esc16a	6688	101.96	67.06	9728	1216	20672	19.21	0.06
esc16b	16192	–	–	22608	2848	48416	25.55	0.09
esc16c	8976	55015.74	97523.58	13056	1632	27744	2283.03	0.06
esc16d	3696	23.29	62.51	5376	672	11424	5.11	0.05
esc16e	3696	20.00	5.18	5376	672	11424	2.52	0.05
esc16f	0	0.00	0.00	0	0	0	0.00	0.04
esc16g	3696	13.07	6.89	5376	672	11424	3.74	0.05
esc16h	20240	86443.74	39008.04	23640	3120	53040	388.37	0.10
esc16i	2640	4.15	4.11	3840	480	8160	1.88	0.05
esc16j	2112	2.27	0.81	3072	384	6528	0.81	0.05
esc32e	4992	1.00	0.66	6144	384	12672	0.91	0.82
esc32g	7488	79.63	44.08	9216	576	19008	36.14	0.84
had12	8712	114777.36	100204.88	9504	1584	20592	2123.90	0.03
had14	16562	–	–	17836	2548	38220	58057.98	0.06
nug12	5940	16651.26	11286.97	6480	1080	14040	84.40	0.02
nug14	12376	–	–	13328	1904	28560	8666.60	0.05
nug15	15750	–	–	16875	2250	36000	13718.10	0.07
nug16b	20160	–	–	21504	2688	45696	17306.12	0.09
rou12	8580	57873.42	27991.90	10020	1680	21720	3806.73	0.03
scr12	3696	65.91	124.91	4032	672	8736	5.58	0.02
scr15	8820	1698.92	4323.83	9450	1260	20160	38.34	0.05
scr20	23560	–	–	24800	2480	52080	2024.03	0.16
tai10a	3870	647.56	1123.16	4570	920	10060	33.76	0.01
tai10b	3150	50.89	80.86	3545	710	7800	1.48	0.01
tai12a	8448	18834.46	14534.13	9744	1632	21120	760.56	0.02
tai12b	7040	1308.84	3414.35	8933	1565	20277	91.38	0.02
tai15b	17010	58183.21	153704.11	18225	2430	38880	190.97	0.07
tai20b	60040	–	–	63200	6320	132720	1569.43	0.26

Table 1: QAPLIB results: Instances solved at least once within 48 hours.

				SLC	SLR			ILW	
Instance	$\|P\|$	LHS	RHS	Solve [s]	Solve [s]	$\|M\|$	NZ+	Solve [s]	Get [s]
jeu_100_25_1	1280	1-50	669	1.53	0.82	199	7815	5.27	0.01
jeu_100_25_2	1239	1-50	2366	0.13	0.13	199	7561	0.91	0.01
jeu_100_25_3	1251	1-49	156	0.51	0.43	199	7635	1.76	0.01
jeu_100_25_4	1273	1-50	2298	2.19	3.62	199	7769	10.89	0.01
jeu_100_25_5	1259	3-50	2536	0.15	0.10	199	7685	0.88	0.01
jeu_100_25_6	1250	1-49	1236	2.71	1.42	199	7633	7.62	0.01
jeu_100_25_7	1261	1-47	466	0.65	0.68	199	7703	4.04	0.01
jeu_100_25_8	1242	1-50	619	0.15	0.11	199	7579	4.13	0.01
jeu_100_25_9	1192	1-50	1597	0.56	0.40	199	7283	0.77	0.01
jeu_100_25_10	1219	1-50	1040	1.91	1.06	199	7439	6.36	0.01
jeu_100_50_1	2519	1-50	1562	1.15	0.73	199	15189	7.47	0.01
jeu_100_50_2	2548	1-49	2019	3.78	3.89	199	15365	13.46	0.01
jeu_100_50_3	2511	1-50	568	6.36	3.64	199	15145	9.46	0.01
jeu_100_50_4	2488	1-50	2208	1.40	2.77	199	15005	8.23	0.01
jeu_100_50_5	2476	1-50	983	10.60	11.49	199	14939	13.48	0.01
jeu_100_50_6	2469	1-50	321	6.74	2.78	199	14895	4.61	0.01
jeu_100_50_7	2495	1-50	1065	1.72	1.22	199	15041	10.88	0.01
jeu_100_50_8	2473	1-50	1041	10.99	5.00	199	14917	12.04	0.01
jeu_100_50_9	2441	2-50	1459	4.08	11.14	199	14729	22.38	0.01
jeu_100_50_10	2469	1-50	1918	58.98	68.85	199	14899	127.67	0.01
jeu_100_75_1	3752	1-50	2466	0.87	0.50	199	22543	0.36	0.01
jeu_100_75_2	3736	2-50	1346	564.23	56.16	199	22443	90825.43	0.01
jeu_100_75_3	3758	2-50	847	6.57	5.24	199	22573	13.14	0.01
jeu_100_75_4	3703	1-50	953	266.00	75.79	199	22249	997.86	0.01
jeu_100_75_5	3723	1-50	370	5.07	3.66	199	22371	4.04	0.01
jeu_100_75_6	3744	1-50	1924	1.38	0.88	199	22491	17.05	0.01
jeu_100_75_7	3776	1-50	1498	43.73	31.74	199	22683	7696.74	0.01
jeu_100_75_8	3719	1-50	382	5.31	7.00	199	22349	9.06	0.01
jeu_100_75_9	3724	1-50	1542	149.73	332.60	199	22375	32512.48	0.01
jeu_100_75_10	3686	1-49	1918	2.83	4.68	199	22155	32.24	0.01
jeu_100_100_1	4950	1-50	616	4.54	4.17	199	29701	17.61	0.01
jeu_100_100_2	4950	1-50	1978	20.91	86.71	199	29701	41794.80	0.01
jeu_100_100_3	4950	1-49	2232	2.52	3.36	199	29701	72.90	0.01
jeu_100_100_4	4950	1-49	574	1.84	1.65	199	29701	13.15	0.01
jeu_100_100_5	4950	1-50	2257	5.46	4.23	199	29701	89.78	0.01
jeu_100_100_6	4950	1-50	726	6.03	12.34	199	29701	19.02	0.01
jeu_100_100_7	4950	1-50	100	2.21	1.32	199	29701	0.68	0.01
jeu_100_100_8	4950	1-49	608	2.06	6.78	199	29701	13.64	0.01
jeu_100_100_9	4950	1-50	2319	10.36	626.47	199	29701	483.19	0.01
jeu_100_100_10	4950	2-50	1779	757.81	686.29	199	29701	3949.44	0.01

Table 2: QKP results (time limit 48 hours), 1st part.

				SLC	SLR	ILW			
Instance	$\|P\|$	LHS	RHS	Solve [s]	Solve [s]	$\|M\|$	NZ+	Solve [s]	Get [s]
jeu_200_25_1	5142	1-50	4027	61.11	60.87	399	31125	32.35	0.03
jeu_200_25_2	5013	1-50	4271	1.20	0.94	399	30351	8.88	0.03
jeu_200_25_3	4984	1-50	4763	1.16	0.88	399	30173	9.20	0.03
jeu_200_25_4	5105	1-50	4518	5.62	8.54	399	30895	19.76	0.03
jeu_200_25_5	4965	1-50	3688	7.31	4.90	399	30063	13.45	0.03
jeu_200_25_6	5053	1-50	1700	166.15	82.40	399	30591	56.59	0.03
jeu_200_25_7	4930	1-50	1242	14.11	7.29	399	29849	98.00	0.03
jeu_200_25_8	4880	1-50	2861	5.36	4.13	399	29555	17.42	0.03
jeu_200_25_9	4846	1-50	1012	13.50	7.05	399	29353	146.09	0.03
jeu_200_25_10	4885	1-50	990	4.45	1.22	399	29579	30.31	0.03
jeu_200_50_1	10028	1-50	3547	1.77	1.47	399	60321	12.38	0.03
jeu_200_50_2	10047	1-50	2245	3557.96	2220.59	399	60445	–	0.03
jeu_200_50_3	9906	1-50	2120	55.85	43.15	399	59609	23509.34	0.03
jeu_200_50_4	10002	1-50	2132	1.71	1.28	399	60159	21.57	0.03
jeu_200_50_5	9910	1-50	4833	27.60	160.69	399	59617	117.95	0.03
jeu_200_50_6	9930	1-50	4354	2701.34	8274.65	399	59745	9645.05	0.03
jeu_200_50_7	9995	1-50	2430	2199.71	1841.56	399	60141	4642.18	0.03
jeu_200_50_8	9877	1-50	3159	7.66	9.38	399	59419	102.94	0.03
jeu_200_50_9	9857	2-50	1105	253.66	86.17	399	59313	–	0.03
jeu_200_50_10	9881	1-50	2756	51.28	14.17	399	59443	152.03	0.03
jeu_200_75_1	14894	1-50	2866	138.72	122.30	399	89425	–	0.03
jeu_200_75_2	14935	1-50	2139	4299.71	2784.87	399	89685	–	0.03
jeu_200_75_3	14869	1-50	408	77.71	83.18	399	89283	71.22	0.03
jeu_200_75_4	14839	1-50	1220	75.82	227.70	399	89105	–	0.03
jeu_200_75_5	14920	1-50	859	247.95	64.08	399	89589	–	0.03
jeu_200_75_6	14898	1-50	1349	13.61	9.44	399	89457	–	0.03
jeu_200_75_7	15051	1-50	1736	13.86	15.09	399	90373	–	0.03
jeu_200_75_8	14836	1-50	3412	7.15	5.05	399	89079	181.14	0.03
jeu_200_75_9	14890	1-50	3412	3140.42	6478.61	399	89407	–	0.03
jeu_200_75_10	14863	1-50	955	24.47	17.60	399	89237	–	0.03
jeu_200_100_1	19900	1-50	4785	58742.92	–	399	119401	–	0.04
jeu_200_100_2	19900	1-50	1325	435.66	561.56	399	119401	–	0.04
jeu_200_100_3	19900	1-50	173	23.43	15.11	399	119401	5.32	0.04
jeu_200_100_4	19900	1-50	498	16.24	10.40	399	119401	35.08	0.04
jeu_200_100_5	19900	1-50	4005	70.09	21256.13	399	119401	–	0.04
jeu_200_100_6	19900	1-50	195	12.82	4.19	399	119401	5.84	0.04
jeu_200_100_7	19900	1-50	3583	199.87	4583.16	399	119401	–	0.04
jeu_200_100_8	19900	1-50	3677	6564.93	53150.59	399	119401	–	0.04
jeu_200_100_9	19900	1-50	3256	216.32	1041.47	399	119401	–	0.04
jeu_200_100_10	19900	1-50	1859	22860.08	2552.15	399	119401	–	0.04
jeu_300_25_1	11339	1-50	376	15.77	5.93	599	68447	71.47	0.07
jeu_300_25_2	11201	1-50	3878	280.98	107.91	599	67623	95.40	0.07
jeu_300_25_3	11157	1-50	3128	70.62	38.47	599	67341	88.89	0.07
jeu_300_25_4	11256	1-50	5759	38.74	16.42	599	67951	71.56	0.07
jeu_300_25_5	11403	1-50	197	13.65	11.21	599	68833	61.05	0.07
jeu_300_25_6	11406	1-50	3451	11.91	17.97	599	68861	90.19	0.07
jeu_300_25_7	11192	1-50	6282	18.39	8.07	599	67569	39.43	0.07
jeu_300_25_8	11157	1-50	114	2.18	1.42	599	67351	6.75	0.07
jeu_300_25_9	11212	1-50	3150	49.67	26.34	599	67689	78.90	0.07
jeu_300_25_10	11340	1-50	5322	126.45	82.65	599	68431	93.54	0.07
jeu_300_50_1	22534	1-50	3550	53394.57	5334.56	599	135457	57006.53	0.08
jeu_300_50_2	22436	1-50	794	362.91	84.72	599	134873	–	0.08
jeu_300_50_3	22351	1-50	5946	275.29	4221.59	599	134359	2550.10	0.08
jeu_300_50_4	22403	1-50	1957	1162.95	217.46	599	134655	–	0.08
jeu_300_50_5	22400	1-50	4912	1397.65	619.52	599	134645	1205.55	0.08
jeu_300_50_6	22444	1-50	5106	1132.39	8731.03	599	134905	4189.81	0.08
jeu_300_50_7	22580	1-50	328	60.50	39.59	599	135725	75.91	0.08
jeu_300_50_8	22253	1-50	5069	2275.33	4230.15	599	133765	1570.17	0.08
jeu_300_50_9	22321	1-50	4907	139.97	36.65	599	134177	298.16	0.08
jeu_300_50_10	22476	1-50	7040	45.47	5605.31	599	135099	2915.94	0.08

Table 3: QKP results (time limit 48 hours), 2nd part.

			SLC	SLR			IL					ILW			
Instance	$\|N\|$	$\|P\|$	Solve [s]	Ineq	NZ+	Solve [s]	$\|Q\|$	$\|M\|$	NZ+	Solve [s]	Get [s]	$\|M\|$	NZ+	Solve [s]	Get [s]
K18_17_100	153	9180	865.33	13781	32141	3620.47	11625	3410	77230	574.23	0.03	3245	64145	516.59	0.04
K18_17_80	153	7344	560.35	11025	25713	1449.24	11628	4154	102690	397.08	0.03	3924	69564	291.99	0.03
K18_17_90	153	8262	656.07	12398	28922	1155.30	11628	4013	97835	425.58	0.03	3800	74039	418.13	0.03
K18_23_100	153	9180	706.29	13770	32130	1400.14	11625	3410	77230	310.35	0.03	3245	64145	361.07	0.04
K18_23_80	153	7344	536.46	11041	25729	907.38	11626	4144	102409	257.13	0.03	3918	69498	230.70	0.03
K18_23_90	153	8262	2464.16	12393	28917	1125.91	11626	4031	98662	493.00	0.03	3841	75084	461.89	0.03
K18_28_100	153	9180	2507.32	13796	32156	1308.11	11625	3410	77230	350.32	0.03	3245	64145	408.15	0.04
K18_28_80	153	7344	1784.70	11023	25711	980.72	11627	4126	101828	496.79	0.03	3908	69163	432.59	0.03
K18_28_90	153	8262	797.29	12416	28940	1247.24	11628	4074	99847	537.55	0.03	3838	75112	511.06	0.03
K18_3_100	153	9180	1164.83	13806	32166	1628.40	11625	3410	77230	385.12	0.03	3245	64145	473.21	0.04
K18_3_80	153	7344	670.46	11057	25745	974.66	11628	4150	102255	438.14	0.03	3883	68566	357.28	0.03
K18_3_90	153	8262	738.99	12450	28974	1336.97	11626	4025	98357	452.35	0.03	3800	74007	357.84	0.04
K18_39_100	153	9180	812.63	13775	32135	1238.80	11625	3410	77230	280.32	0.03	3245	64145	297.92	0.03
K18_39_80	153	7344	419.22	11040	25728	895.65	11628	4133	101915	321.06	0.03	3899	69020	249.34	0.03
K18_39_90	153	8262	533.14	12402	28926	1675.70	11628	4015	98061	300.98	0.03	3811	74278	295.92	0.04
K18_41_100	153	9180	1015.73	13774	32134	2491.62	11625	3410	77230	449.82	0.03	3245	64145	535.49	0.04
K18_41_80	153	7344	684.34	10995	25683	1322.49	11625	4146	102450	400.73	0.03	3912	69360	405.80	0.03
K18_41_90	153	8262	812.61	12399	28923	1436.62	11624	4019	98085	377.39	0.03	3800	74036	367.67	0.03
K18_58_100	153	9180	1158.26	13796	32156	1495.88	11625	3410	77230	446.82	0.03	3245	64145	400.53	0.03
K18_58_80	153	7344	467.05	11039	25727	960.88	11627	4104	100820	277.66	0.03	3859	68047	225.02	0.04
K18_58_90	153	8262	629.37	12401	28925	1039.72	11628	4028	98111	337.26	0.03	3782	73693	299.02	0.04
K18_63_100	153	9180	851.81	13797	32157	1512.70	11625	3410	77230	325.32	0.03	3245	64145	326.24	0.04
K18_63_80	153	7344	524.97	11049	25737	931.84	11628	4146	102422	409.83	0.03	3932	69685	350.39	0.03
K18_63_90	153	8262	912.01	12409	28933	1317.29	11628	4062	99375	518.24	0.03	3822	74734	471.44	0.04
K18_67_100	153	9180	1144.29	13773	32133	1476.88	11625	3410	77230	393.78	0.03	3245	64145	434.28	0.03
K18_67_80	153	7344	1625.90	10993	25681	933.28	11627	4154	102599	462.23	0.03	3909	69218	403.05	0.04
K18_67_90	153	8262	1934.24	12385	28909	1377.55	11627	4046	98967	407.65	0.04	3822	74609	411.82	0.03
K18_93_100	153	9180	814.22	13770	32130	1291.07	11625	3410	77230	393.24	0.03	3245	64145	376.34	0.03
K18_93_80	153	7344	1657.91	11016	25704	857.98	11626	4134	101988	422.04	0.03	3895	68821	369.46	0.03
K18_93_90	153	8262	2117.53	12394	28918	1878.21	11628	4037	98758	567.49	0.03	3826	74717	570.01	0.04
K19_17_100	171	11628	4330.20	17493	40749	2831.58	14532	4040	96584	549.60	0.05	3856	81040	879.03	0.05
K19_17_80	171	9303	3988.32	13954	32560	2476.80	14534	4930	128865	1588.97	0.04	4661	87813	979.17	0.05
K19_17_90	171	10466	2515.63	15744	36676	459.61	14535	4856	126216	287.67	0.04	4609	96283	400.12	0.04
K19_23_100	171	11628	1105.39	17479	40735	2064.09	14532	4040	96584	504.38	0.04	3856	81040	514.52	0.05
K19_23_80	171	9303	1395.96	13971	32577	1767.92	14535	4936	129082	801.81	0.04	4673	88047	602.78	0.04
K19_23_90	171	10466	1865.48	15733	36665	3237.52	14535	4770	123013	1097.34	0.04	4511	93317	1121.83	0.05
K19_28_100	171	11628	1504.00	17484	40740	3415.36	14532	4040	96584	981.26	0.05	3856	81040	685.72	0.05
K19_28_80	171	9303	4289.59	13965	32571	2579.83	14535	4934	129083	1326.34	0.04	4693	88593	1926.92	0.04
K19_28_90	171	10466	4667.09	15743	36675	3718.81	14535	4810	124643	961.30	0.04	4587	95508	1111.58	0.05
K19_3_100	171	11628	1611.25	17460	40716	4545.11	14532	4040	96584	878.08	0.04	3856	81040	892.10	0.05
K19_3_80	171	9303	831.57	13991	32597	1722.54	14535	4916	128482	812.64	0.04	4676	88145	539.92	0.05
K19_3_90	171	10466	1771.22	15725	36657	2996.45	14535	4773	123205	876.92	0.04	4539	94123	878.19	0.05
K19_39_100	171	11628	1585.78	17484	40740	3856.66	14532	4040	96584	887.73	0.04	3856	81040	671.85	0.05
K19_39_80	171	9303	1313.47	13972	32578	2303.94	14533	4884	127210	1088.51	0.04	4631	87001	699.72	0.04
K19_39_90	171	10466	4646.78	15724	36656	2928.30	14535	4842	125813	1348.36	0.04	4600	96032	1134.49	0.05
K19_41_100	171	11628	2160.67	17441	40697	2776.68	14532	4040	96584	787.29	0.05	3856	81040	847.96	0.05
K19_41_80	171	9303	1232.71	13975	32581	1668.04	14534	4914	128389	590.59	0.04	4661	87843	514.71	0.04
K19_41_90	171	10466	1491.75	15675	36607	2537.36	14534	4823	124898	987.99	0.04	4565	95111	894.99	0.04
K19_58_100	171	11628	4751.17	17466	40722	3085.90	14532	4040	96584	1455.29	0.04	3856	81040	1474.16	0.05
K19_58_80	171	9303	3197.03	13990	32596	2624.64	14535	4986	130997	1303.99	0.04	4724	89423	836.77	0.05
K19_58_90	171	10466	4169.97	15726	36658	2846.35	14535	4778	123523	1173.04	0.04	4554	94487	1357.35	0.05
K19_63_100	171	11628	4910.11	17465	40721	3098.66	14532	4040	96584	1365.15	0.04	3856	81040	942.62	0.05
K19_63_80	171	9303	3559.21	13971	32577	1950.50	14535	4868	126449	978.38	0.04	4604	86317	558.83	0.04
K19_63_90	171	10466	1234.94	15724	36656	896.90	14535	4793	123940	722.45	0.04	4567	94836	493.80	0.05
K19_67_100	171	11628	1674.57	17433	40689	4697.87	14532	4040	96584	751.23	0.05	3856	81040	768.02	0.05
K19_67_80	171	9303	2945.44	13959	32565	1855.40	14535	4941	129158	1344.06	0.04	4674	88044	845.54	0.04
K19_67_90	171	10466	4122.96	15705	36637	2277.33	14535	4799	124212	1097.08	0.04	4562	94775	947.90	0.05
K19_93_100	171	11628	6118.43	17420	40676	3640.49	14532	4040	96584	1406.71	0.04	3856	81040	1430.16	0.05
K19_93_80	171	9303	993.14	13948	32554	3029.26	14528	4919	128485	1044.29	0.04	4640	87369	770.05	0.04
K19_93_90	171	10466	1827.76	15697	36629	2599.38	14535	4804	124230	1366.21	0.04	4549	94374	971.88	0.05

Table 4: QMP results for “BM” instances (time limit 48 hours), 1st part.

			SLC	SLR			IL					ILW			
Instance	$\|N\|$	$\|P\|$	Solve [s]	Ineq	NZ+	Solve [s]	$\|Q\|$	$\|M\|$	NZ+	Solve [s]	Get [s]	$\|M\|$	NZ+	Solve [s]	Get [s]
K20_17_100	190	14535	19733.08	21838	50908	15507.11	17952	4743	119357	5616.03	0.06	4539	101065	5182.32	0.06
K20_17_80	190	11628	7147.72	17490	40746	3385.88	17955	5822	160775	1596.70	0.05	5531	110755	1261.97	0.06
K20_17_90	190	13082	15194.27	19658	45822	10218.31	17955	5689	155493	3574.06	0.06	5403	119442	6977.35	0.06
K20_23_100	190	14535	13595.14	21796	50866	10361.05	17952	4743	119357	2347.86	0.06	4539	101065	2091.11	0.06
K20_23_80	190	11628	7804.73	17497	40753	4404.60	17952	5819	160415	2158.95	0.06	5519	110555	3930.81	0.06
K20_23_90	190	13082	12490.33	19645	45809	10007.09	17955	5649	154183	3315.55	0.06	5378	118434	3244.97	0.06
K20_28_100	190	14535	12360.49	21823	50893	14044.73	17952	4743	119357	1997.05	0.06	4539	101065	2414.43	0.06
K20_28_80	190	11628	7620.56	17476	40732	4865.67	17954	5802	160005	2654.43	0.06	5526	110669	2313.64	0.06
K20_28_90	190	13082	12972.67	19651	45815	8508.79	17955	5685	155637	7697.55	0.06	5421	119832	6116.36	0.06
K20_3_100	190	14535	13150.075	21839	50909	9154.45	17952	4743	119357	2675.69	0.06	4539	101065	2656.56	0.06
K20_3_80	190	11628	1998.41	17468	40724	4257.13	17954	5795	159684	1752.27	0.05	5511	110214	1218.31	0.05
K20_3_90	190	13082	13921.00	19631	45795	9572.28	17955	5620	153153	3539.23	0.06	5372	118167	6313.30	0.06
K20_39_100	190	14535	15663.16	21803	50873	12613.88	17952	4743	119357	7049.16	0.06	4539	101065	7288.75	0.06
K20_39_80	190	11628	8515.76	17467	40723	8896.66	17955	5796	159727	2467.92	0.06	5515	110228	2399.33	0.06
K20_39_90	190	13082	11369.24	19614	45778	7817.29	17954	5652	154273	2841.31	0.06	5388	118925	2510.63	0.06
K20_41_100	190	14535	17282.99	21845	50915	9125.29	17952	4743	119357	5301.75	0.06	4539	101065	3313.35	0.06
K20_41_80	190	11628	10032.50	17436	40692	6818.22	17955	5776	158795	5520.09	0.05	5483	109205	2655.13	0.06
K20_41_90	190	13082	12018.00	19616	45780	7977.47	17954	5631	153548	2749.40	0.06	5366	117988	2988.19	0.06
K20_58_100	190	14535	13943.73	21811	50881	8110.51	17952	4743	119357	2261.89	0.06	4539	101065	2358.86	0.06
K20_58_80	190	11628	9262.58	17469	40725	6214.46	17954	5830	160969	2431.40	0.06	5530	110816	4236.39	0.06
K20_58_90	190	13082	12944.56	19670	45834	7350.97	17954	5690	155796	2259.98	0.05	5434	120209	2780.77	0.06
K20_63_100	190	14535	9948.42	21770	50840	10298.26	17952	4743	119357	1276.78	0.06	4539	101065	1238.20	0.06
K20_63_80	190	11628	7864.67	17475	40731	5235.16	17955	5790	159572	4404.42	0.05	5505	109866	1980.01	0.05
K20_63_90	190	13082	11654.11	19638	45802	9500.01	17955	5690	155774	2396.45	0.06	5433	120259	1998.13	0.06
K20_67_100	190	14535	13283.53	21805	50875	10154.22	17952	4743	119357	2807.47	0.06	4539	101065	5675.81	0.06
K20_67_80	190	11628	9466.09	17427	40683	6120.35	17955	5790	159305	2805.08	0.05	5493	109625	4329.78	0.06
K20_67_90	190	13082	10500.06	19620	45784	5120.34	17955	5672	155096	2131.94	0.06	5408	119406	2245.00	0.06
K20_93_100	190	14535	17306.87	21761	50831	8160.72	17952	4743	119357	2100.87	0.06	4539	101065	5124.05	0.06
K20_93_80	190	11628	9565.66	17424	40680	8530.73	17954	5759	158191	5051.62	0.05	5474	109106	4391.31	0.06
K20_93_90	190	13082	11687.70	19609	45773	10931.63	17953	5655	154408	3239.85	0.06	5391	119089	3182.61	0.06
K21_17_100	210	17955	26507.70	26914	62824	35364.42	21942	5523	145929	5194.61	0.07	5298	124584	9171.96	0.07
K21_17_80	210	14364	19435.40	21566	50294	15908.32	21945	6733	194597	5313.99	0.07	6407	134842	4823.73	0.07
K21_17_90	210	16160	25108.02	24245	56565	17545.39	21945	6521	186283	10005.48	0.07	6229	144244	10358.20	0.07
K21_23_100	210	17955	26843.80	26893	62803	35514.31	21942	5523	145929	8888.96	0.07	5298	124584	9694.70	0.07
K21_23_80	210	14364	17276.43	21543	50271	12442.11	21945	6749	195283	4495.04	0.07	6419	135307	8659.43	0.07
K21_23_90	210	16160	26725.55	24179	56499	22315.60	21944	6608	189618	11074.30	0.07	6290	146588	10826.18	0.08
K21_28_100	210	17955	28102.11	26956	62866	38996.85	21942	5523	145929	9256.28	0.07	5298	124584	5828.11	0.07
K21_28_80	210	14364	18913.74	21556	50284	26264.49	21945	6760	195694	6237.01	0.07	6430	135523	9184.36	0.07
K21_28_90	210	16160	19765.16	24244	56564	20937.61	21945	6635	190884	5050.20	0.07	6343	148236	4443.90	0.07
K21_3_100	210	17955	29022.89	26943	62853	67846.55	21942	5523	145929	8406.56	0.07	5298	124584	8315.26	0.07
K21_3_80	210	14364	19720.13	21570	50298	36056.48	21944	6820	198086	6641.79	0.07	6480	137139	8995.66	0.07
K21_3_90	210	16160	26041.21	24254	56574	59973.55	21945	6652	191657	12512.26	0.07	6352	148398	10149.51	0.07
K21_39_100	210	17955	30487.96	26920	62830	19365.35	21942	5523	145929	8738.27	0.07	5298	124584	5490.23	0.07
K21_39_80	210	14364	14310.99	21543	50271	10526.63	21942	6795	197336	7839.65	0.07	6483	137180	5852.11	0.07
K21_39_90	210	16160	17706.16	24224	56544	12397.71	21945	6622	190599	3881.51	0.07	6352	148376	3861.49	0.07
K21_41_100	210	17955	33877.18	26990	62900	68331.28	21942	5523	145929	11947.83	0.07	5298	124584	11939.19	0.07
K21_41_80	210	14364	15725.40	21581	50309	20543.89	21944	6779	196529	5570.79	0.07	6457	136536	7517.71	0.07
K21_41_90	210	16160	19689.54	24294	56614	37318.89	21943	6615	190147	5013.34	0.07	6324	147722	4720.60	0.07
K21_58_100	210	17955	25166.54	26943	62853	30625.05	21942	5523	145929	11382.75	0.07	5298	124584	9809.66	0.07
K21_58_80	210	14364	17280.63	21578	50306	21460.62	21942	6731	194637	5695.43	0.07	6410	134970	8295.13	0.07
K21_58_90	210	16160	25028.40	24256	56576	21632.76	21945	6633	190835	11727.69	0.07	6348	148337	10531.51	0.07
K21_63_100	210	17955	22302.23	26877	62787	54681.61	21942	5523	145929	4222.91	0.07	5298	124584	4418.43	0.07
K21_63_80	210	14364	18244.70	21525	50253	14482.44	21945	6771	196370	6053.73	0.07	6471	136537	4183.36	0.07
K21_63_90	210	16160	19105.08	24214	56534	15381.13	21943	6594	189405	4446.82	0.07	6308	147107	4451.25	0.07
K21_67_100	210	17955	7568.28	26927	62837	16083.19	21942	5523	145929	7444.78	0.07	5298	124584	3235.57	0.07
K21_67_80	210	14364	18957.16	21539	50267	11717.01	21945	6792	197172	5360.14	0.07	6482	137013	7611.21	0.07
K21_67_90	210	16160	20355.23	24250	56570	20138.18	21945	6654	191785	3767.25	0.07	6358	148684	3520.95	0.07
K21_93_100	210	17955	20833.15	26907	62817	21159.29	21942	5523	145929	3366.20	0.07	5298	124584	3448.91	0.07
K21_93_80	210	14364	19251.05	21508	50236	26899.74	21944	6785	196813	3568.50	0.07	6456	136275	6473.67	0.07
K21_93_90	210	16160	22573.38	24185	56505	24833.93	21944	6571	188247	10168.41	0.07	6271	145783	9105.03	0.07

Table 5: QMP results for “BM” instances (time limit 48 hours), 2nd part.

			SLC	SLR			IL					ILW			
Instance	$\lvert N\rvert$	$\lvert P\rvert$	Solve [s]	Ineq	NZ+	Solve [s]	$\lvert Q\rvert$	$\lvert M\rvert$	NZ+	Solve [s]	Get [s]	$\lvert M\rvert$	NZ+	Solve [s]	Get [s]
K22_17_100	231	21945	134715.36	32912	76802	59315.10	26562	6384	176700	25921.97	0.09	6137	151981	25570.69	0.09
K22_17_80	231	17556	49155.36	26352	61464	50815.57	26565	7934	242313	24368.53	0.08	7589	169812	21343.08	0.09
K22_17_90	231	19751	73700.54	29614	69116	92042.90	26565	7678	231595	25708.40	0.09	7341	180674	24676.10	0.09
K22_23_100	231	21945	114690.48	32852	76742	99941.95	26562	6384	176700	28517.16	0.09	6137	151981	26451.17	0.09
K22_23_80	231	17556	77987.06	26291	61403	76674.25	26565	7829	237957	35706.13	0.08	7484	166344	32771.95	0.09
K22_23_90	231	19751	60223.44	29539	69041	48079.22	26564	7647	230190	20039.35	0.09	7303	179176	22186.75	0.09
K22_28_100	231	21945	81810.26	32966	76856	91688.15	26562	6384	176700	26769.19	0.09	6137	151981	29432.23	0.09
K22_28_80	231	17556	43757.63	26365	61477	34442.58	26565	7815	237455	14036.81	0.09	7483	166149	18541.75	0.09
K22_28_90	231	19751	64520.19	29650	69152	–	26565	7744	234667	11309.04	0.09	7438	183801	10668.57	0.09
K22_3_100	231	21945	88358.50	32892	76782	115135.66	26562	6384	176700	26647.00	0.09	6137	151981	26321.95	0.09
K22_3_80	231	17556	62804.12	26354	61466	67380.08	26564	7918	241565	25463.16	0.09	7562	168914	24510.65	0.09
K22_3_90	231	19751	94883.29	29605	69107	–	26561	7701	232681	36820.61	0.09	7359	181402	31725.86	0.09
K22_39_100	231	21945	86177.15	32886	76776	125030.83	26562	6384	176700	22689.85	0.09	6137	151981	26811.37	0.09
K22_39_80	231	17556	66293.51	26309	61421	80277.97	26565	7888	240257	27662.56	0.09	7521	167719	21807.59	0.09
K22_39_90	231	19751	78190.30	29596	69098	77390.00	26565	7678	231731	26696.36	0.09	7352	180851	25894.73	0.09
K22_41_100	231	21945	104780.79	33014	76904	155869.14	26562	6384	176700	24698.87	0.09	6137	151981	25542.68	0.09
K22_41_80	231	17556	50778.36	26387	61499	83371.21	26565	7804	236769	24042.93	0.08	7442	164956	22577.01	0.09
K22_41_90	231	19751	45250.67	29669	69171	34215.19	26565	7648	230251	12455.20	0.09	7322	179977	19790.26	0.09
K22_58_100	231	21945	128789.42	32906	76796	–	26562	6384	176700	25280.56	0.09	6137	151981	29321.37	0.10
K22_58_80	231	17556	68430.03	26344	61456	44474.52	26563	7860	239095	26747.84	0.08	7495	166865	25631.77	0.09
K22_58_90	231	19751	78739.05	29619	69121	58741.74	26565	7586	227702	19864.40	0.09	7261	177643	18300.49	0.09
K22_63_100	231	21945	109292.68	32864	76754	110556.34	26562	6384	176700	30489.56	0.09	6137	151981	31302.66	0.10
K22_63_80	231	17556	80274.70	26276	61388	34528.20	26565	7862	239234	24523.12	0.09	7493	166786	20518.02	0.09
K22_63_90	231	19751	99028.71	29582	69084	72471.07	26564	7696	232471	26118.10	0.09	7374	181664	24837.86	0.10
K22_67_100	231	21945	83185.53	32936	76826	84625.58	26562	6384	176700	23129.73	0.09	6137	151981	23993.10	0.09
K22_67_80	231	17556	54065.74	26339	61451	56263.24	26564	7874	239957	21566.50	0.09	7532	167882	20066.27	0.09
K22_67_90	231	19751	84870.24	29606	69108	57069.36	26565	7686	232040	33142.07	0.09	7360	181280	27562.69	0.09
K22_93_100	231	21945	92519.36	32914	76804	94917.85	26562	6384	176700	21706.25	0.09	6137	151981	25738.66	0.09
K22_93_80	231	17556	42619.02	26310	61422	113003.64	26565	7841	238451	22538.49	0.09	7497	166591	17926.61	0.09
K22_93_90	231	19751	63255.83	29597	69099	103743.99	26563	7695	232368	24148.28	0.10	7356	181157	19589.18	0.09

Table 6: QMP results for "BM" instances (time limit 48 hours), 3rd part.

				SLC	SLR			IL		
Instance	$\|A\|$	$\|V\|$	$\|P\|$	Solve [s]	Solve [s]	$\|Q\|$	$\|M\|$	NZ+	Solve [s]	Get [s]
grid1_10_0815	180	100	8010	102.70	56.75	33404	19258	88652	1275.51	0.12
grid1_10_42	180	100	8010	60.38	34.98	33404	19258	88652	1275.67	0.12
grid1_10_4711	180	100	8010	107.19	33.28	33404	19258	88652	1367.56	0.13
grid1_11_0815	220	121	11990	309.51	178.96	50026	28468	132112	6567.49	0.20
grid1_11_42	220	121	11990	283.15	194.84	50026	28468	132112	7658.73	0.19
grid1_11_4711	220	121	11990	297.72	198.52	50026	28468	132112	6660.81	0.20
grid1_12_0815	264	144	17292	1742.70	731.23	72134	40610	189708	30168.53	0.36
grid1_12_42	264	144	17292	1792.81	713.82	72134	40610	189708	31964.53	0.34
grid1_12_4711	264	144	17292	2154.91	694.96	72134	40610	189708	33530.25	0.34
grid1_13_0815	312	169	24180	5019.94	3452.68	100808	56242	264182	160524.46	0.50
grid1_13_42	312	169	24180	4653.13	3313.86	100808	56242	264182	152468.89	0.49
grid1_13_4711	312	169	24180	4560.50	4243.47	100808	56242	264182	137000.06	0.50
grid1_14_0815	364	196	32942	24308.45	17007.25	137224	75970	358516	–	0.70
grid1_14_42	364	196	32942	22106.88	13360.12	137224	75970	358516	–	0.70
grid1_14_4711	364	196	32942	27085.16	11203.53	137224	75970	358516	–	0.70
grid1_15_0815	420	225	43890	91504.54	103304.10	182654	100448	475932	–	0.99
grid1_15_42	420	225	43890	50628.34	29848.88	182654	100448	475932	–	0.99
grid1_15_4711	420	225	43890	95390.47	49241.94	182654	100448	475932	–	0.99
grid1_16_0815	480	256	57360	–	–	238466	130378	619892	–	1.38
grid1_16_42	480	256	57360	–	–	238466	130378	619892	–	1.38
grid1_16_4711	480	256	57360	–	–	238466	130378	619892	–	1.38
grid2_16_16_0815	512	258	65416	4557.18	1595.99	284983	144664	729008	126203.27	1.70
grid2_16_16_42	512	258	65416	4055.84	397.97	284983	144664	729008	134164.53	1.75
grid2_16_16_4711	512	258	65416	3419.21	445.44	284983	144664	729008	105840.64	1.76
grid2_16_32_0815	1008	514	253968	–	–	1077311	555000	2753504	–	9.87
grid2_16_32_42	1008	514	253968	–	–	1077311	555000	2753504	–	9.87
grid2_16_32_4711	1008	514	253968	–	–	1077311	555000	2753504	–	9.87
grid2_16_64_0815	2000	1026	1000480	–	–	4183375	2172856	10690112	–	68.17
grid2_16_64_42	2000	1026	1000480	–	–	4183375	2172856	10690112	–	67.18
grid2_16_64_4711	2000	1026	1000480	–	–	4183375	2172856	10690112	–	66.52
grid2_23_23_0815	1058	531	279588	–	–	1195606	602338	3037895	–	11.40
grid2_23_23_42	1058	531	279588	–	–	1195606	602338	3037895	–	11.31
grid2_23_23_4711	1058	531	279588	–	–	1195606	602338	3037895	–	11.38
grid2_32_16_0815	1040	514	269960	21638.48	82398.28	1174935	579952	2976456	–	10.86
grid2_32_16_42	1040	514	269960	82993.64	19127.31	1174935	579952	2976456	–	10.86
grid2_32_16_4711	1040	514	269960	43731.58	52388.82	1174935	579952	2976456	–	10.86
grid2_64_16_0815	2096	1026	1096840	–	–	4770391	2322208	12027896	–	74.89
grid2_64_16_42	2096	1026	1096840	–	–	4770391	2322208	12027896	–	75.06
grid2_64_16_4711	2096	1026	1096840	–	–	4770391	2322208	12027896	–	74.80

Table 7: QSPP results (time limit 48 hours).

Instance	$\lvert N\rvert$	$\lvert K\rvert$	$\lvert P\rvert$	LHS	RHS	SLC Solve [s]	SLR Ineq	SLR NZ+	SLR Solve [s]	IL $\lvert Q\rvert$	IL $\lvert M\rvert$	IL NZ+	IL Solve [s]	IL Get [s]	GRB Solve [s]
cardqp_inlp	50	1	1225	1-1	10-10	24.04	1225	3675	172.43	1225	50	2500	257.67	0.00	17.59
cardqp_iqp	50	1	1225	1-1	10-10	23.21	1225	3675	172.11	1225	50	2500	250.80	0.00	17.67
celar6-sub0	640	16	64032	1-1	1-1	1728.75	64032	192096	1434.51	93744	4624	192112	55.96	0.33	95.12
color_lab2_4x0	300	61	44850	1-1	0-1	–	74401	164101	–	44250	17700	106200	–	0.15	–
color_lab3_3x0	316	80	2241	1-1	0-1	228.67	2241	6723	40.11	3984	1992	9960	56702.56	0.06	61.67
color_lab3_4x0	395	80	3984	1-1	0-1	122791.97	3984	11952	–	6225	2490	14940	–	0.08	–
color_lab6b_4x20	235	48	27495	1-1	1-20	–	53291	108281	–	27025	10810	64860	–	0.08	–
crossdock_15x7	210	44	2793	1-233	1-302	796.20	2793	8379	398.74	2793	798	6384	129.43	0.04	5927.33
crossdock_15x8	240	46	3648	1-209	1-233	1886.86	3648	10944	1702.54	3648	912	8208	352.47	0.06	20740.32
graphpart_2g-0044-1601	48	16	96	1-1	1-1	0.03	141	333	0.02	288	192	768	0.04	0.00	0.02
graphpart_2g-0055-0062	75	25	150	1-1	1-1	0.15	219	519	0.04	450	300	1200	0.08	0.00	0.05
graphpart_2g-0066-0066	108	36	216	1-1	1-1	0.18	342	774	0.15	648	432	1728	0.17	0.01	0.13
graphpart_2g-0077-0077	147	49	294	1-1	1-1	0.47	426	1014	0.18	882	588	2352	0.28	0.01	0.15
graphpart_2g-0088-0088	192	64	384	1-1	1-1	0.44	612	1380	0.16	1152	768	3072	0.40	0.02	0.14
graphpart_2g-0099-9211	243	81	486	1-1	1-1	0.69	711	1683	0.42	1458	972	3888	1.98	0.04	0.29
graphpart_2g-1010-0824	300	100	600	1-1	1-1	0.73	891	2091	0.33	1800	1200	4800	1.53	0.06	0.18
graphpart_2pm-0044-0044	48	16	96	1-1	1-1	0.04	144	336	0.03	288	192	768	0.05	0.00	0.03
graphpart_2pm-0055-0055	75	25	150	1-1	1-1	0.11	225	525	0.05	450	300	1200	0.08	0.00	0.05
graphpart_2pm-0066-0066	108	36	216	1-1	1-1	0.28	324	756	0.10	648	432	1728	0.20	0.01	0.07
graphpart_2pm-0077-0777	147	49	294	1-1	1-1	0.34	441	1029	0.18	882	588	2352	0.78	0.01	0.08
graphpart_2pm-0088-0888	192	64	384	1-1	1-1	0.37	576	1344	0.21	1152	768	3072	0.64	0.02	0.16
graphpart_2pm-0099-0999	243	81	486	1-1	1-1	0.93	729	1701	0.31	1458	972	3888	25.65	0.04	0.58

Table 10: MINLPLIB results: Linearly constrained BQPs (time limit 48 hours), 1st part.

Instance	$\lvert N\rvert$	$\lvert K\rvert$	$\lvert P\rvert$	LHS	RHS	SLC Solve [s]	SLR Ineq	SLR NZ+	SLR Solve [s]	IL $\lvert Q\rvert$	IL $\lvert M\rvert$	IL NZ+	IL Solve [s]	IL Get [s]	GRB Solve [s]
graphpart_3g-0234-0234	72	24	180	1-1	1-1	0.32	264	624	0.08	540	360	1440	0.11	0.00	0.09
graphpart_3g-0244-0244	96	32	240	1-1	1-1	0.35	354	834	0.24	720	480	1920	0.17	0.01	0.17
graphpart_3g-0333-0333	81	27	243	1-1	1-1	0.32	345	831	0.12	729	486	1944	0.16	0.00	0.07
graphpart_3g-0334-0334	108	36	324	1-1	1-1	0.40	471	1119	0.20	972	648	2592	0.48	0.01	0.13
graphpart_3g-0344-0344	144	48	432	1-1	1-1	0.97	648	1512	0.46	1296	864	3456	0.70	0.01	0.25
graphpart_3g-0444-0444	192	64	576	1-1	1-1	3.19	900	2052	1.19	1728	1152	4608	4.77	0.02	2.66
graphpart_3pm-0234-0234	72	24	180	1-1	1-1	0.21	264	624	0.10	540	360	1440	0.23	0.00	0.09
graphpart_3pm-0244-0244	96	32	240	1-1	1-1	0.32	360	840	0.23	720	480	1920	0.60	0.01	0.15
graphpart_3pm-0333-0333	81	27	243	1-1	1-1	0.18	363	849	0.05	729	486	1944	0.55	0.00	0.07
graphpart_3pm-0334-0334	108	36	324	1-1	1-1	0.62	486	1134	0.22	972	648	2592	1.88	0.01	0.25
graphpart_3pm-0344-0344	144	48	432	1-1	1-1	3.26	648	1512	0.78	1296	864	3456	13.95	0.01	1.38
graphpart_3pm-0444-0444	192	64	576	1-1	1-1	158.73	864	2016	22.97	1728	1152	4608	248.86	0.02	14.93
graphpart_clique-20	60	20	570	1-1	1-1	1.10	570	1710	0.42	1710	1140	4560	0.43	0.00	1.33
graphpart_clique-30	90	30	1305	1-1	1-1	10.33	1305	3915	4.59	3915	2610	10440	8.04	0.01	6.32
graphpart_clique-40	120	40	2340	1-1	1-1	161.23	2340	7020	46.80	7020	4680	18720	66.65	0.01	46.21
graphpart_clique-50	150	50	3675	1-1	1-1	827.74	3675	11025	309.64	11025	7350	29400	1132.64	0.02	419.64
graphpart_clique-60	180	60	5310	1-1	1-1	4702.26	5310	15930	4524.61	15930	10620	42480	7758.67	0.03	17350.87
graphpart_clique-70	210	70	7245	1-1	1-1	18288.97	7245	21735	27741.62	21735	14490	57960	62382.71	0.05	–
maxcsp-ehi-85-297-12	2071	297	101333	1-1	1-1	45719.27	101333	303999	2286.35	199509	57192	456210	22252.52	3.60	–
maxcsp-ehi-85-297-36	2046	297	99286	1-1	1-1	15499.17	99286	297858	1339.03	196266	56782	449314	42026.42	3.32	33702.74
maxcsp-ehi-85-297-71	2075	297	101923	1-1	1-1	–	101923	305769	18673.46	200794	57454	459042	35185.74	3.79	–
maxcsp-ehi-90-315-70	2203	315	108662	1-1	1-1	138493.45	108662	325986	14105.88	214502	61338	490342	51774.76	4.46	–
maxcsp-geo50-20-d4-75-36	1000	50	33837	1-1	1-1	1.28	33837	101511	0.29	135600	13560	284760	1872.53	0.70	7.31
maxcsp-langford-3-11	627	33	14196	1-1	1-1	52981.63	14196	42588	61752.76	182604	18908	384116	–	0.33	–
pb302035	600	50	165300	1-98	1-310	–	165300	495900	–	174000	17400	365400	–	0.97	–
pb302055	600	50	165300	1-98	1-201	–	165300	495900	–	174000	17400	365400	–	0.97	–
pb302075	600	50	165300	1-98	1-150	–	165300	495900	–	174000	17400	365400	–	0.97	–
pb302095	600	50	165300	1-98	1-129	–	165300	495900	–	174000	17400	365400	147727.04	0.97	–
pb351535	525	50	124950	1-98	1-525	–	124950	374850	–	133875	17850	285600	–	0.69	–
pb351555	525	50	124950	1-98	1-336	–	124950	374850	–	133875	17850	285600	–	0.69	–
pb351575	525	50	124950	1-98	1-246	–	124950	374850	–	133875	17850	285600	–	0.69	–
pb351595	525	50	124950	1-98	1-196	–	124950	374850	–	133875	17850	285600	–	0.69	–

Table 11: MINLPLIB results: Linearly constrained BQPs (time limit 48 hours), 2nd part.

3.6 QPLIB

Finally, we look at the linearly constrained BQPs in the QPLIB ([9]) except instance `QPLIB_3380` and those that are also part of MINLPLib, and we again also pass the `.lp` files directly to Gurobi. Tables 12 (equations only), 13 and 14 (inequalities only), and 15 (mixed) display the results. In case of the latter, only the equations were employed for inductions which is why only IL is displayed.

						SLC	SLR			IL					GRB
Instance	$\lvert N\rvert$	$\lvert K_E\rvert$	$\lvert P\rvert$	LHS	RHS	Solve [s]	Ineq	NZ+	Solve [s]	$\lvert Q\rvert$	$\lvert M\rvert$	NZ+	Solve [s]	Get [s]	Solve [s]
QPLIB_0633	75	1	2775	1-1	15-15	1011.74	2775	8325	71280.06	2775	75	5625	–	0.01	767.21
QPLIB_2492	196	28	16562	1-1	1-1	–	16562	49686	–	17836	2548	38220	88514.69	0.05	–
QPLIB_2512	100	20	3870	1-1	1-1	557.81	3870	11610	579.52	4570	920	10060	34.92	0.01	76.17
QPLIB_2733	324	36	46818	1-1	1-1	–	46818	140454	–	49572	5508	104652	–	0.18	–
QPLIB_2880	625	50	176410	1-1	1-1	–	176410	529230	–	191159	15495	402333	–	1.07	–
QPLIB_2957	484	44	70686	1-1	1-1	–	70686	212058	–	74052	6732	154836	–	0.39	–
QPLIB_3307	256	32	20160	1-1	1-1	–	20160	60480	–	21504	2688	45696	16599.42	0.09	–
QPLIB_3347	676	52	195780	1-1	1-1	–	195780	587340	–	199160	20956	565396	–	1.21	–
QPLIB_3361	1024	64	163680	1-1	1-1	–	163680	491040	–	168960	10560	348480	–	1.93	–
QPLIB_3402	144	24	8448	1-1	1-1	21988.51	8448	25344	32906.48	9744	1632	21120	816.23	0.02	2563.28
QPLIB_3413	400	40	7220	1-1	1-1	12158.50	7220	21660	18284.62	7600	760	15960	3.14	0.13	299.55
QPLIB_3703	225	30	21424	1-1	1-1	–	21424	64272	–	24346	3300	52616	–	0.08	–
QPLIB_3714	120	40	2340	1-1	1-1	60.01	2340	7020	70.46	7020	4680	18720	80.04	0.01	79.37
QPLIB_3750	210	70	7245	1-1	1-1	35492.51	7245	21735	48740.67	21735	14490	57960	73178.77	0.05	–
QPLIB_3751	150	50	3675	1-1	1-1	1736.65	3675	11025	539.56	11025	7350	29400	600.44	0.02	673.35
QPLIB_3775	180	60	5310	1-1	1-1	3527.61	5310	15930	4006.11	15930	10620	42480	6836.08	0.03	17390.82
QPLIB_3815	192	64	576	1-1	1-1	103.92	864	2016	22.69	1728	1152	4608	242.31	0.02	14.47
QPLIB_3834	50	1	1225	1-1	10-10	24.46	1225	3675	184.23	1225	50	2500	281.86	0.00	17.82
QPLIB_6487	618	309	39806	1-1	1-1	–	39806	119418	–	84536	84536	253608	–	0.64	–
QPLIB_6597	600	60	175229	1-1	1-1	–	175229	525687	–	177000	35400	389400	–	0.67	–

Table 12: QPLIB results: (Linear) Equation-constrained BQPs (time limit 48 hours).

First of all, the overall results support once more the impression from the last subsection that even among the instances with left hand side coefficients as well as right hand sides equal to one, the inductive as well as the standard linearization can potentially outperform the other, or solve the respective instance at all within the time limit. To exemplify an extreme, when using IL the running times are orders of magnitude faster than with SLR or SLC and even Gurobi as a stand-alone solver is clearly outperformed for `QPLIB_3413`. On the other hand, IL cannot solve `QPLIB_0633`, which has just one equation asking to select 15 out of 75 variables such that the objective is minimum, within the time limit while with SLR less than twenty minutes are needed.

For the instances with equations in Tables 12 and 15 the overall performance of IL compares rather favorably. Here, it is often even faster than Gurobi standalone, and e.g. the instance `QPLIB_3709` is solved to optimality (according to the QPLIB website, this was not achieved before). The optimal value is 5710645 which is also found by Gurobi when applying it standalone and with many threads on an according system, but even then the search tree becomes too large so that an optimality proof seems yet out of reach. On the other hand, on the inequality-only instances in Tables 13 and 14, IL is outperformed first by ILW and often also by the standard linearizations and Gurobi as a standalone solver. Thus, only ILW is displayed in these tables. As a remark, the "standard" linearization and WIL results for the depicted instances `QPLIB_59XX` could be significantly improved using a preprocessing that rules out products guaranteed to be zero, and the results for `QPLIB_0067` are in line with those from Sect. 3.2 as this is a small QKP. Moreover, it becomes once more visible that especially large coefficients and right hand sides (as in the `QPLIB_100XX` instances), as well as negative left hand side coefficients, yet lead to less effective inductive linearizations.

						SLC	SLR			ILW				GRB
Instance	$\lvert N\rvert$	$\lvert K_I\rvert$	$\lvert P\rvert$	LHS	RHS	Solve [s]	Ineq	NZ+	Solve [s]	$\lvert M\rvert$	NZ+	Solve [s]	Get [s]	Solve [s]
QPLIB_0067	80	1	2844	2-50	1555-1555	24.44	5688	11376	142.07	159	17067	148.37	0.00	133.69
QPLIB_0752	250	1	3114	-1–1	-1–1	19508.62	4686	10914	14939.01	499	19113	–	0.05	8292.78
QPLIB_2315	595	13090	13233	-1-1	0-1	–	21942	48408	–	45029	151857	–	72.14	–
QPLIB_2357	240	2240	2254	-1-1	0-1	815.22	3418	7926	1145.32	6687	22661	8280.32	2.96	435.97
QPLIB_2359	306	3264	1740	-1-1	0-1	587.33	2553	6033	449.35	4752	15909	2065.47	5.60	498.70
QPLIB_3584	528	10912	11191	-1-1	0-1	–	18536	40918	–	38035	128495	–	49.12	–
QPLIB_3752	462	6160	3988	-1-1	0-1	126341.44	5836	13812	129153.49	11666	39601	–	16.54	57862.84
QPLIB_3757	552	8096	1463	-1-1	0-1	914.37	2189	5115	1002.23	3887	12039	4789.89	24.42	306.05
QPLIB_3762	90	480	1133	-1-1	0-1	121.89	1680	3946	397.66	3554	12271	698.51	0.30	149.38
QPLIB_3772	380	4560	2761	-1-1	0-1	2198.66	4117	9639	3880.57	7887	26577	26686.29	10.54	4558.53
QPLIB_3803	190	2280	2538	-1-1	0-1	394.59	4250	9326	1557.79	8833	28905	5415.00	2.86	275.16
QPLIB_3841	300	4600	4600	-1-1	0-1	129444.55	7603	16803	157600.00	15367	51290	–	9.87	69994.31
QPLIB_3860	435	8120	8204	-1-1	0-1	–	13571	29979	–	27838	93563	–	28.16	–
QPLIB_3883	182	1456	2947	-1-1	0-1	39226.59	4324	10218	19283.58	9068	31406	–	1.32	18844.30
QPLIB_5935	100	1237	4950	1-1	1-1	3579.99	8725	18625	2014.93	9090	32948	954.46	0.28	243.70
QPLIB_5944	100	2475	4950	1-1	1-1	150.54	8730	18630	114.25	9090	32948	266.63	0.64	16.74
QPLIB_5962	150	2793	11175	1-1	1-1	98462.02	19705	42055	60706.68	21876	78994	30225.99	1.29	11072.12
QPLIB_5971	150	5587	11175	1-1	1-1	4852.78	19672	42022	3462.85	21876	78994	3377.20	2.50	104.09
QPLIB_5980	150	8381	11175	1-1	1-1	181.58	19649	41999	120.81	21876	78994	278.54	3.71	48.84

Table 13: QPLIB results: (Linear) Inequality-constrained BQPs (time limit 48 hours), 1st part.

4 Conclusion

A framework to derive inductive linearizations in practice has been outlined and it has been demonstrated that it can be effectively applied to a variety of binary quadratic programs with linear constraints. The experiments covering the Quadratic Assignment, Knapsack, Matching and Shortest Path Problems, as well as instances from the MINLBLib and QPLIB, show that the performance that can be expected when combining an inductive linearization with a professional MIP solver depends on the target application. As is also predicted from theory, as a rule of thumb, small ratios between right hand sides and (non-negative) left hand side coefficients (ideally, ratios of one) of the employed constraints are favorable, and especially equations appear to be effective to obtain a compact and strong linearization at the same time. But also inequalities can be effectively used to build well-suited inductive linearizations as especially the results for the Quadratic Matching Problem showed. Conversely, large ratios and negative coefficients are not yet handled as routinely. This is a clear field for further research, as well as additional computational studies providing even more mosaics and tailored approaches for particular applications. There is also room for developments and improvements regarding the exact and heuristic derivation of inductive linearizations that has just been prototyped so far. Yet, depending also on further parameters such as the rate of truly present product terms and the distribution of their factors over the set of (employed) constraints, the well-known "standard" linearization as well as commercial solvers may either be clearly outperformed or superior. Notwithstanding that, in some fortunate cases, the methodology allows to solve established benchmark library instances to optimality for the first time as is the case with the instances `pb302035`, `pb302095`, and `pb351535` from the MINLPLib, and apparently also with `QPLIB_3709` from the QPLIB.

						SLC	SLR			ILW				GRB
Instance	$\|N\|$	$\|K_I\|$	$\|P\|$	LHS	RHS	Solve [s]	Ineq	NZ+	Solve [s]	$\|M\|$	NZ+	Solve [s]	Get [s]	Solve [s]
QPLIB_10040	125	6	7174	1-998	62-31468	1793.51	10822	25170	844.45	247	43045	–	0.05	0.00
QPLIB_10041	125	6	7615	1-998	62-31468	–	11395	26625	–	247	45691	–	0.05	0.23
QPLIB_10042	125	5	7619	2-1000	47135-50209	47649.47	11420	26658	31755.62	247	45715	–	0.04	0.11
QPLIB_10043	150	10	10513	1-1000	54036-58799	571.05	15703	36729	148.83	297	63121	–	0.13	0.01
QPLIB_10044	150	6	10549	1-999	50-57547	1326.93	15852	36950	495.15	291	63295	–	0.08	2.26
QPLIB_10045	150	10	10091	1-1000	54036-58799	–	15202	35384	–	285	60547	–	0.12	0.02
QPLIB_10046	150	6	9875	1-999	50-57547	350.25	14894	34644	159.58	287	59253	–	0.08	0.14
QPLIB_10047	150	10	11168	1-1000	54036-58799	–	16811	39147	–	299	67009	–	0.13	0.01
QPLIB_10048	150	5	11018	2-1000	55564-60822	–	16475	38511	–	297	66109	–	0.07	127.98
QPLIB_10049	150	10	11018	1-1000	54036-58799	–	16528	38564	–	297	66109	–	0.13	0.01
QPLIB_10050	150	5	10878	2-1000	55564-60822	–	16287	38043	–	295	65269	–	0.07	0.53
QPLIB_10051	150	10	11015	1-1000	54036-58799	–	16438	38468	–	297	66091	–	0.13	4.45
QPLIB_10052	150	6	11165	1-999	50-57547	–	16798	39128	–	299	66991	–	0.08	2.78
QPLIB_10053	150	10	11019	1-1000	54036-58799	–	16634	38672	–	297	66115	–	0.12	0.01
QPLIB_10054	175	11	13398	1-1000	116-22886	519.02	20162	46958	142.53	339	80389	–	0.22	41.78
QPLIB_10055	175	5	13918	2-1000	66009-70533	575.32	20924	48760	177.60	347	83511	–	0.09	1.22
QPLIB_10056	175	5	14535	2-1000	66009-70533	–	21912	50982	–	341	87211	–	0.09	0.20
QPLIB_10057	200	11	15056	1-1000	66-77923	427.72	22601	52713	85.84	371	90347	–	0.27	0.01
QPLIB_10058	200	11	17491	1-1000	100-54119	1128.88	26231	61213	275.03	395	105337	–	0.28	248.14
QPLIB_10059	200	10	14135	1-1000	45545-53253	–	21165	49435	–	341	84811	–	0.24	0.02
QPLIB_10060	200	10	17474	1-1000	45545-53253	933.20	26114	61062	318.57	389	104896	–	0.29	0.09
QPLIB_10061	200	5	17892	2-1000	73590-80431	–	26912	62696	–	381	107353	–	0.13	0.84
QPLIB_10062	200	10	18162	1-1000	45545-53253	–	27215	63539	–	383	108973	–	0.26	0.05
QPLIB_10063	200	5	19413	2-1000	73590-80431	5803.32	29189	68015	2459.25	395	116479	–	0.14	0.08
QPLIB_10064	200	11	19272	1-1000	66-77923	–	28885	67429	–	393	115633	–	0.30	0.01
QPLIB_10065	200	11	19302	1-1000	100-50591	–	28925	67529	–	393	115813	–	0.32	2.07
QPLIB_10066	200	11	19301	1-1000	100-54119	–	28747	67349	–	393	115807	–	0.29	0.40
QPLIB_10067	200	5	19642	2-1000	73590-80431	–	29462	68746	–	397	117853	–	0.14	0.34
QPLIB_10068	200	11	19879	1-1000	100-54119	–	29898	69656	–	399	119275	–	0.30	1.14
QPLIB_10069	200	10	19264	1-1000	45545-53253	1183.56	28787	67315	580.68	405	117012	–	0.26	0.01
QPLIB_10070	200	11	19478	1-1000	100-54119	–	29215	68171	–	395	116869	–	0.33	30.93
QPLIB_10071	200	11	19698	1-1000	66-77923	–	29552	68948	–	399	118189	–	0.33	0.02
QPLIB_10072	75	10	2049	1-1000	23837-30414	7300.00	3104	7202	4618.37	133	12295	–	0.03	0.00
QPLIB_10073	75	6	1928	1-999	37-10480	9100.78	2889	6745	1675.62	133	11697	–	–	0.02
QPLIB_10074	75	10	2774	1-1000	23837-30414	13759.47	4141	9689	8382.24	149	16645	–	0.03	0.00

Table 14: QPLIB results: (Linear) Inequality-constrained BQPs (time limit 48 hours), 2nd part.

							SLC	SLR			IL					GRB
Instance	$\|N\|$	$\|K_E\|$	$\|K_I\|$	$\|P\|$	LHS	RHS	Solve [s]	Ineq	NZ+	Solve [s]	$\|Q\|$	$\|M\|$	NZ+	Solve [s]	Get [s]	Solve [s]
QPLIB_3587	240	30	16	3648	1-209	1-233	1357.53	3648	10944	1007.97	3648	912	8208	418.58	0.05	5816.12
QPLIB_3614	210	30	14	2793	1-233	1-302	750.88	2793	8379	519.03	2793	798	6384	45.07	0.04	8919.05
QPLIB_3709	600	30	20	165300	1-98	1-129	–	165300	495900	–	174000	17400	365400	148374.37	1.16	–
QPLIB_3865	525	35	15	124950	1-98	1-246	–	124950	374850	–	133875	17850	285600	–	0.76	–

Table 15: QPLIB results: Linearly constrained (equations and inequalities) BQPs (time limit 48 hours).

References

[1] Warren P. Adams and Terri A. Johnson. Improved linear programming-based lower bounds for the quadratic assignment problem. In P. M. Pardalos and H. Wolkowicz, editors, *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS Series on Discrete Mathematics and Computer Science*, pages 43–75. AMS, Providence, RI, USA, 1994.

[2] Warren P. Adams and Hanif D. Sherali. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, volume 31 of *Nonconvex Optimization and its Applications*. Springer, 1999.

[3] Alain Billionnet and Frédéric Calmels. Linear programming for the 0–1 quadratic knapsack problem. *European Journal of Operational Research*, 92(2):310–325, 1996.

[4] Alain Billionnet and Sourour Elloumi. Best reduction of the quadratic semi-assignment problem. *Discrete Applied Mathematics*, 109(3):197–213, 2001.

[5] Alain Billionnet and Éric Soutif. An exact method based on lagrangian decomposition for the 0–1 quadratic knapsack problem. *European Journal of Operational Research*, 157(3):565–575, 2004.

[6] 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.

[7] T. Davidović, L. Liberti, N. Maculan, and N. Mladenović. Towards the optimal solution of the multiprocessor scheduling problem with communication delays. In P. Baptiste, G. Kendall, A. Munier-Kordon, and F. Sourd, editors, *Proc. 3rd Multidisciplinary Int. Conf. on Scheduling: Theory and Application*, pages 128–135. École Polytechnique, Paris, France, 2007.

[8] A. M. Frieze and J. Yadegar. On the quadratic assignment problem. *Discrete Appl. Math.*, 5(1):89–98, 1983.

[9] Fabio Furini, Emiliano Traversi, Pietro Belotti, Antonio Frangioni, Ambros Gleixner, Nick Gould, Leo Liberti, Andrea Lodi, Ruth Misener, Hans Mittelmann, Nikolaos Sahinidis, Stefan Vigerske, , and Angelika Wiegele. QPLIB: A library of quadratic programming instances. *Mathematical Programming Computation*, 11:237–265, 2019.

[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] C. Helmberg, F. Rendl, and R. Weismantel. A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, 4(2):197–215, Jun 2000.

[12] Lena Hupp, Laura Klein, and Frauke Liers. An exact solution method for quadratic matching: The one-quadratic-term technique and generalisations. *Discrete Optimization*, 18:193–216, 2015.

[13] Tjalling C. Koopmans and Martin Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25(1):53–76, 1957.

[14] L. Liberti. Compact linearization for binary quadratic problems. *4OR*, 5(3):231–245, 2007.

[15] Sven Mallach. Improved mixed-integer programming models for the multiprocessor scheduling problem with communication delays. *Journal of Combinatorial Optimization*, Nov 2017.

[16] Sven Mallach. Compact linearization for binary quadratic problems subject to assignment constraints. *4OR*, 16:295–309, Sep 2018.

[17] Sven Mallach. A natural quadratic approach to the generalized graph layering problem. In Daniel Archambault and Csaba D. Tóth, editors, *Graph Drawing and Network Visualization*, pages 532–544, Cham, 2019. Springer.

[18] Sven Mallach. Inductive linearization for binary quadratic programs with linear constraints. *4OR*, Oct 2020.

[19] Borzou Rostami, André Chassein, Michael Hopf, Davide Frey, Christoph Buchheim, Federico Malucelli, and Marc Goerigk. The quadratic shortest path problem: complexity, approximability, and solution methods. *European Journal of Operational Research*, 268(2):473–485, 2018.