Cutting Planes for RLT Relaxations of Mixed 0-1 Polynomial Programs

Franklin Djeumou Fomeni Konstantinos Kaparis
Adam N. Letchford*

December 2013

Abstract

The *Reformulation-Linearization Technique* (RLT), due to Sherali and Adams, can be used to construct hierarchies of linear programming relaxations of mixed 0-1 polynomial programs. As one moves up the hierarchy, the relaxations grow stronger, but the number of variables increases exponentially. We present a procedure that generates cutting planes at any given level of the hierarchy, by optimally weakening linear inequalities that are valid at any given higher level. Computational experiments, conducted on instances of the quadratic knapsack problem, indicate that the cutting planes can close a significant proportion of the integrality gap.

Keywords: polynomial optimisation, cutting planes, mixed-integer nonlinear programming, quadratic knapsack problem.

1 Introduction

The *Reformulation-Linearization Technique* (RLT), developed by Adams, Sherali and co-authors, is a general framework for constructing hierarchies of linear programming (LP) relaxations of various optimisation problems. Although first developed for 0-1 polynomial programs (PPs) [23], it was soon adapted to continuous PPs [27], and then extended to mixed 0-1 PPs [24]. Since then, it has been further extended and adapted, to cover a wide range of integer programming and global optimisation problems (see, e.g., [22, 25]).

As one moves up the levels of the RLT hierarchy, the LP relaxations grow stronger, but the number of variables increases exponentially. In practice, therefore, one can hope to solve the relaxations only at very low levels of

*Department of Management Science, Lancaster University Management School, Lancaster LA1 4YX, United Kingdom. E-mail: {F.DjeumouFomeni,K.Kaparis,A.N.Letchford}@lancaster.ac.uk

the hierarchy. In fact, even solving the relaxation at the first level can be a challenge [3]. This led us to develop the framework presented in this paper, which enables one to strengthen low-level relaxations by adding cutting planes, rather than additional variables.

Our framework is applicable to mixed 0-1 PPs with bounded continuous variables. It has three main components. First, we give procedures for constructing low-degree polynomial over- and under-estimators of polynomials of higher degree. Second, we show how to use those over- and under-estimators to convert any valid linear inequality for a given high-level RLT relaxation into an exponentially large collection of cutting planes for a given low-level RLT relaxation. Third, we present separation algorithms that, under certain conditions, enable one to find in polynomial time the most-violated cutting plane in a given collection.

In order to explore the potential of the new cutting planes, we present some computational results, using the *quadratic knapsack problem* (QKP) as an example. It turns out that, for the difficult *sparse* QKP instances, our cutting planes close around half of the gap between the upper bound from the standard first-level relaxation and the optimum.

The paper is organised as follows. In Section 2, we review the relevant literature. In Section 3, we present the new over- and under-estimators. In Section 4, the cutting planes are defined, and it is shown how to strengthen them using disjunctive arguments. In Section 5, we present the separation algorithms. The computational results are presented in Section 6. Finally, some concluding remarks are presented in Section 7.

2 Literature Review

In this section, we review the relevant literature. We review the RLT for 0-1 LPs in Subsection 2.1, extensions of the RLT to other problems in Subsection 2.2, and some strengthening procedures in Subsection 2.3.

2.1 The RLT for 0-1 LPs

The RLT was first introduced in Sherali & Adams [23], in the context of 0-1 LPs. Suppose we have a 0-1 LP of the form:

$$\min \quad c^T x \tag{1}$$

$$\text{s.t.} \quad Ax \le b \tag{2}$$

$$x \in \{0,1\}^n, \tag{3}$$

where $c \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{m \times n}$ and $b \in \mathbb{Q}^m$. The *continuous relaxation* of the instance is the problem obtained by replacing the constraints (3) with the weaker constraints $x \in [0,1]^n$.

Let $N = \{1, \ldots, n\}$. Given any integer $1 \leq k \leq n$, the level-k RLT relaxation is constructed in two phases. The first step (reformulation) involves the construction of a system of valid polynomial inequalities of degree $k+1$. The second phase (linearisation) involves the replacement of monomials of degree greater than 1 with new variables. The details of the reformulation phase are as follows:

- For any disjoint pair $S, T \subset N$, let $J(S,T)$ denote $\prod_{i \in S} x_i \prod_{i \in T}(1-x_i)$.
- For each disjoint pair $S, T \subset N$ satisfying $|S| + |T| = k+1$, place the inequality $J(S,T) \geq 0$ in the system.
- For each linear inequality in the system (2), say $\alpha^T x \leq \beta$, and each disjoint pair $S, T \subset N$ satisfying $|S| + |T| = k$, place the inequality

  $$J(S,T)(\beta - \alpha^T x) \geq 0$$

  in the system.

The details of the linearisation phase are as follows:

- Take each polynomial inequality in the system and expand the left-hand side so that it becomes a weighted sum of distinct monomials.
- Make each monomial multilinear using the identities $x_i^r = x_i$ for all $i \in N$ and all integers $2 \leq r \leq k+1$.
- For each $S \subseteq N$ with $2 \leq |S| \leq \min\{k+1, n\}$, let y_S be a new binary variable, representing the multilinear monomial $\prod_{i \in S} x_i$.
- Linearise each of the polynomial (now multilinear) inequalities in the system, by replacing each multilinear monomial of degree larger than one with the corresponding y variable.

Sherali and Adams showed that the RLT relaxation gets stronger as one moves up the levels of the hierarchy, and becomes optimal when $k = n$. On the other hand, the number of variables and constraints increases exponentially with increasing k. In practice, it is often the first level that is of most use for 0-1 LPs (see, e.g., [3, 26]).

2.2 Extensions of the RLT to more general problems

At the end of [23], Sherali and Adams extended the RLT to 0-1 PPs. The reformulation phase is the same, except that one multiplies each original inequality, whether linear or not, by the $J(S,T)$ terms. The linearisation phase is also the same, except that, if the objective function is non-linear, one must make it multilinear and then express it in terms of the x and y variables.

Note that, if the original 0-1 PP contains an inequality that involves a polynomial of degree d, then the level-k relaxation will contain y_S variables for sets S of cardinality up to $\min\{d+k, n\}$. Perhaps for this reason, this variant of the RLT has been applied mainly to 0-1 quadratic programs (e.g., [1, 4, 12, 14, 21]).

In Sherali & Tuncbilek [27], the RLT was adapted to *continuous* PPs with bounded variables. The approach is similar, but with two small differences. The first is that one must begin by scaling all variables so that they are bounded between zero and one. The second is that the identities of the form $x_i^r = x_i$ are no longer valid. As a result, the RLT relaxation contains y variables that represent general monomials, rather than only multilinear ones.

Unfortunately, when applied to PPs, the RLT hierachy no longer converges to the optimum as k approaches n. Sherali & Adams [24] showed however that the convergence result still holds if the RLT is applied to mixed 0-1 PPs, under the restrictions that (i) all continuous variables are bounded and (ii) the objective and constraint functions do not involve products of continuous variables.

The RLT has also been extended to many other problems in integer programming and global optimisation. See, e.g., [22, 25] for details.

2.3 Known ways to strengthen RLT relaxations

The following three ways to strengthen level-1 RLT relaxations have been proposed in the literature:

1. As well as multiplying linear inequalities by terms of the form x_i and $1 - x_i$ in the reformulation phase, one can also multiply pairs of linear inequalities together [17]. Specifically, given two inequalities of the form $\alpha^T x \le \beta$ and $\gamma^T x \le \delta$, one can generate the quadratic inequality $(\beta - \alpha^T x)(\delta - \gamma^T x) \ge 0$, which can then be linearised in the usual way.

2. Let X be the $n \times n$ symmetric matrix in which $X_{ii} = x_i$ for all i and $X_{ij} = y_{ij}$ for all $i \ne j$. Note that $X = xx^T$. Also define the augmented matrix
$$\hat{X} = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}.$$
Then one can strengthen the relaxation by adding the constraint that $\hat{X}$ must be positive semidefinite [17].

3. One can add any inequality that is valid for the so-called *Boolean quadric polytope*, which was defined by Padberg [18] as:
$$\text{conv}\left\{(x, y) \in \{0,1\}^{n+\binom{n}{2}} : \; y_{ij} = x_i x_j \; (\{i,j\} \subset N)\right\}.$$

> Many valid inequalities are known for this polytope; see the survey Deza & Laurent [7]. Such inequalities have been used, e.g., in [2, 4, 13, 14, 29].

The first idea can easily be extended to level-k relaxations with $k > 1$, by multiplying $k+1$ linear inequalities together (or indeed any collection of polynomial inequalities whose degrees sum to $k+1$). The second idea has been extended to higher levels by, e.g., Lasserre [16] and Parrilo [19].

3 Low-Degree Polynomial Estimators

In this section, we consider how to derive polynomial over- and under-estimators of a given low degree for monomials of a given higher degree. We remark that the estimators provided in this section differ from the ones usually used in global optimisation (see, e.g., [8, 28]), in that they are neither convex nor concave in general.

3.1 Best possible estimators for a special case

Consider a multilinear monomial of the form $\prod_{i \in S} x_i$, where $|S| \geq 2$. The following lemma presents a natural family of multilinear over- and under-estimators of degree $|S|-1$. (In this lemma, we use the convention that $\prod_{i \in \emptyset} x_i = 1$.)

Lemma 1 *Let $0 \leq x_i \leq 1$ for $i \in S$. For any $T \subseteq S$ with $|T|$ odd, we have:*

$$\prod_{i \in S} x_i \leq \sum_{W \subset T} (-1)^{|W|} \prod_{i \in (S \setminus T) \cup W} x_i. \tag{4}$$

(Here, W is permitted to be empty.) Similarly, for any $T \subseteq S$ with $|T|$ even, we have:

$$\prod_{i \in S} x_i \geq \sum_{W \subset T} (-1)^{|W|-1} \prod_{i \in (S \setminus T) \cup W} x_i. \tag{5}$$

(Here, both T and W are permitted to be empty.)

Proof. Since all variables are in $[0,1]$, we have

$$\prod_{i \in S \setminus T} x_i \prod_{i \in T} (1 - x_i) \geq 0.$$

Expanding the left-hand side we obtain:

$$\sum_{W \subseteq T} (-1)^{|W|} \prod_{i \in (S \setminus T) \cup W} x_i \geq 0,$$

or, equivalently,

$$(-1)^{|T|} \prod_{i \in S} x_i + \sum_{W \subset T} (-1)^{|W|} \prod_{i \in (S \setminus T) \cup W} x_i \geq 0.$$

The result follows. □

We illustrate this lemma with a simple example:

Example 1: Suppose that we seek quadratic estimators for the cubic monomial $x_1x_2x_3$. Accordingly, we set $S = \{1, 2, 3\}$ in Lemma 1. By then setting T to each of $\{1\}$, $\{2\}$, $\{3\}$ and $\{1, 2, 3\}$ in turn, we obtain the over-estimators x_2x_3, x_1x_3, x_1x_2 and

$$1 - x_1 - x_2 - x_3 + x_1x_2 + x_1x_3 + x_2x_3,$$

respectively. By setting T to each of $\emptyset$, $\{1, 2\}$, $\{1, 3\}$ and $\{2, 3\}$ in turn, we obtain the under-estimators 0, $-x_3 + x_1x_3 + x_2x_3$, $-x_2 + x_1x_2 + x_2x_3$ and $-x_1 + x_1x_2 + x_1x_3$, respectively. So, we obtain four distinct over-estimators and four distinct under-estimators for the given monomial. □

In general, Lemma 1 yields 2^{d-1} distinct over-estimators and 2^{d-1} distinct under-estimators for a multilinear monomial of degree d. The following theorem shows that these estimators are, in a sense, best possible.

Theorem 1 *For any $n > 2$, consider the following set:*

$$\left\{ x \in \{0, 1\}^{2^n - 1} : \; x_S = \prod_{i \in S} x_i \; (\forall S \subseteq N : |S| > 1) \right\}.$$

The convex hull of this set is completely described by the inequalities

$$\sum_{W \subseteq T} (-1)^{|W|} x_{(N \setminus T) \cup W} \geq 0 \qquad (T \subseteq N), \tag{6}$$

where, as before, both T and W are permitted to be empty.

Proof. The 2^n points in the set are affinely independent, and therefore each of them is a vertex of the convex hull. Since the convex hull is a polytope with 2^n vertices lying in a space of dimension $2^n - 1$, it must be a simplex. As a result, it must have 2^n facets. Moreover, by construction, each of the 2^n inequalities (6) is satisfied at equality at all but one of the 2^n vertices. To see this, note that, for a given T, the inequality is satisfied at equality at a vertex x^* if and only if:

$$\prod_{i \in N \setminus T} x_i^* \prod_{i \in T} (1 - x_i^*) = 0,$$

or, equivalently,

$$\bigvee_{i \in N \setminus T} (x_i^* = 0) \ \vee \ \bigvee_{i \in T} (x_i^* = 1).$$

Equivalently, the inequality is *not* satisfied at equality if and only if $x_i^* = 1$ for all $i \in N \setminus T$ and $x_i^* = 0$ for all $i \in T$. Therefore each of the inequalities (6) defines a facet. □

Remark: If one has a monomial that is not multilinear, one can still use Lemma 1 to derive estimators of one lower degree, simply by identifying relevant variables, and then eliminating duplicates. This is illustrated in the following example.

Example 2: Suppose that we seek quadratic estimators for the cubic monomial $x_1^2 x_2$. We simply take the eight estimators given in Example 1, identify x_3 with x_1, and then eliminate duplicates. This yields three distinct over-estimators, namely x_1x_2, x_1^2 and $1-2x_1-x_2+2x_1x_2+x_1^2$, and three distinct under-estimators, namely 0, $-x_1+x_1^2+x_1x_2$ and $-x_2+2x_1x_2$. □

3.2 Estimators obtained via recursion

By applying Lemma 1 recursively, one can obtain polynomial estimators of any given degree for monomials of any given higher degree. One must then be careful, however, not only to eliminate duplicates, but also to eliminate estimators that are dominated by others. This is illustrated in the following example.

Example 3: Suppose that we seek *linear* estimators for the cubic monomial $x_1x_2x_3$. In Example 1, we found four quadratic over-estimators and four quadratic under-estimators. Applying Lemma 1 with sets S of cardinality 2, each quadratic term appearing in any of the four quadratic over-estimators, say x_ix_j, can itself be replaced with either x_i or x_j. Enumerating all combinations yields ten distinct linear over-estimators, namely, x_1, x_2, x_3, 1, and $1 + x_i - x_j$ for all $\{i, j\} \subset \{1, 2, 3\}$. Of these, only the first three are non-dominated.

In an analogous way, any quadratic term x_ix_j appearing in any of the four quadratic under-estimators can be replaced with either 0 or $x_i + x_j - 1$. Enumerating all combinations yields eight distinct linear under-estimators, namely, 0, $-x_1$, $-x_2$, $-x_3$, $x_1 - 1$, $x_2 - 1$, $x_3 - 1$ and $x_1 + x_2 + x_3 - 2$. Of these, only the first and last are non-dominated.

So we obtain three non-dominated linear over-estimators and two non-dominated linear under-estimators for the given monomial. □

Clearly, the time taken to enumerate the estimators that can be derived by a recursive application of Lemma 1, and then eliminate duplicate and dominated estimators, grows very rapidly with the degree of the monomial under consideration. This is not a serious drawback, however, since we will be using these estimators to derive cutting planes for RLT relaxations (see the next section), and, in practice, the RLT is useful only for mixed 0-1 PPs involving polynomials of very low degree.

3.3 Estimators obtained via polyhedral computations

Another way to compute estimators of degree d for a monomial of degree larger than $d+1$ is to use a polyhedral computation package, such as `PORTA` [6] or `cdd` [9]. For example, if one feeds into these packages the 8 vectors in the following set:

$$\left\{x \in \{0,1\}^4 : x_{123} = x_1 x_2 x_3\right\},$$

one finds that the convex hull of the set is defined by the inequalities $x_{123} \geq 0$, $x_{123} \leq x_1 + x_2 + x_3 - 2$, and $x_{123} \leq x_i$ for $i = 1, 2, 3$. This implies that the five non-dominated estimators presented in Example 3 are best possible.

In a similar way, to enumerate the estimators of degree 2 or less for the quartic monomial $x_1x_2x_3x_4$, it suffices to compute the convex hull of the set

$$\left\{x \in \{0,1\}^{11} :\ x_{ij} = x_i x_j \ (1 \leq i < j \leq 4),\ x_{1234} = x_1 x_2 x_3 x_4\right\}.$$

This yields 16 over-estimators, namely:

- x_1x_2 and five other similar expression obtained by permutation;
- $1 - x_1 - x_2 - x_3 + x_1x_2 + x_1x_3 + x_2x_3$ and three similar expressions;
- $x_1 - x_1x_2 - x_1x_3 - x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$ and three similar expressions;
- $3 - 2\sum_{i=1}^4 x_i + \sum_{1 \leq i < j \leq 4} x_i x_j$
- and $(1 - \sum_{i=1}^4 x_i + \sum_{1 \leq i < j \leq 4} x_i x_j)/3$.

It also yields 23 under-estimators, namely:

- $-2x_1 + x_1x_2 + x_1x_3 + x_1x_4$ are three similar expressions;
- $-x_1 - x_2 + x_1x_2 + x_1x_3 + x_2x_4$ and eleven similar expressions;
- $-x_1 - x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 - x_3x_4$ and five similar expressions;
- and zero.

We remark that the last of the over-estimators cannot be derived using the recursive argument presented in the previous subsection.

Unfortunately, the time taken to perform these computations increases very rapidly with the degree of the given monomial. Moreover, the number of estimators increases extremely rapidly. For example, we found 694 best-possible quadratic estimators for the quintic monomial $x_1x_2x_3x_4x_5$.

4 The Cutting Planes

In this section, we show how to use the estimators presented in the last section to generate cutting planes for RLT relaxations of mixed 0-1 polynomial programs. We assume that the given mixed 0-1 PP instance takes the following form:

$$\inf \quad p^0(x) \tag{7}$$
$$\text{s.t.} \quad p^j(x) \geq 0 \quad (j = 1, \ldots, m) \tag{8}$$
$$x_i \in \{0, 1\} \quad (i \in B) \tag{9}$$
$$x_i \in [0, 1] \quad (i \in N \setminus B), \tag{10}$$

where B denotes the set of binary variables. We let M denote $\{1, \ldots, m\}$, and, for each $j \in M$, we let d_j denote the degree of the polynomial $p^j(x)$.

4.1 The main concept

Suppose we are given an RLT relaxation of the mixed 0-1 PP (7)–(10), or indeed any other linear or semidefinite relaxation, that includes all of the x variables, together with the y variables for all monomials that are multilinear in the binary variables, and have degree between 2 and d, for some integer $d \geq 2$. Also let $d' > d$ be a given integer parameter. In order to derive a *single* cutting plane that involves only the x variables and the existing y variables, we can proceed as follows:

1. Let S, T be any disjoint subsets of N, and M' be any subset of M, such that $|S| + |T| + \sum_{j \in M'} d_j = d'$.

2. Construct the valid polynomial inequality:

   $$J(S, T) \prod_{j \in M'} p^j(x) \geq 0,$$

   which is of order d' by construction.

3. Expand the left-hand side so that it becomes a weighted sum of distinct monomials.

4. Simplify each monomial, if possible, using the identities $x_i^r = x_i$ for all $i \in B$ and all integers $2 \leq r \leq d'$.

5. Convert the resulting polynomial inequality into a valid polynomial inequality of degree d, by replacing each monomial of degree greater than d with an estimator of degree d, derived using the procedures presented in the previous section. (Choose an over-estimator if the monomial has positive weight, and an under-estimator if it has negative weight.)

6. Expand the left-hand side of the resulting degree-d inequality so that it becomes a weighted sum of distinct monomials once more.

7. Replace each monomial of degree larger than one with the corresponding y variable.

Using this approach, one can easily generate exponentially-large families of cutting planes for mixed 0-1 PPs. A detailed example is given in the next subsection.

4.2 Example: the (s,t) inequalities

In this subsection, we show that our method yields several non-trivial, and exponentially large, families of cutting planes, even when $d = 2$ and $d' = 3$. These cutting planes can be applied to level-1 RLT relaxations of pure or mixed 0-1 linear or quadratic programs. (See Section 6 for an application to the QKP.)

We will need the following notation: for any $S \subset N$ and any $\alpha \in \mathbb{Q}^n$, let $\alpha(S)$ denote $\sum_{i \in S} \alpha_i$, S^+ denote $\{i \in S : \alpha_i > 0\}$, and S^- denote $\{i \in S : \alpha_i < 0\}$. The cutting planes are presented in the following three theorems.

Theorem 2 *Suppose that $p^j(x)$ is linear for some $j \in M$, and takes the form $\beta - \alpha^T x$. (That is, the corresponding constraint can be written as $\alpha^T x \leq \beta$.) For any pair $\{s,t\} \subset B$, and any disjoint subsets $S, T, W \subset N \setminus \{s,t\}$, let R denote $N \setminus (\{s,t\} \cup S \cup T \cup W)$. Then the following '$(s,t)$' inequalities are valid for all possible s, t, S, T and W:*

$$\sum_{i \in S \cup W} \alpha_i y_{is} + \sum_{i \in T \cup W} \alpha_i y_{it} - \sum_{i \in W} \alpha_i x_i \leq -\alpha(W^-) + \alpha(S^+ \cup W^-) x_s$$

$$+ \alpha(T^+ \cup W^-) x_t + \big(\beta - \alpha(\{s,t\} \cup S^+ \cup T^+ \cup W^- \cup R^-)\big) y_{st}. \qquad (11)$$

Proof. We start by constructing the following valid cubic inequality:

$$(\beta - \alpha^T x) x_s x_t \geq 0.$$

Since x_s and x_t are binary, we have $x_s x_t = x_s^2 x_t = x_s x_t^2$, and the cubic inequality can be re-written as:

$$\sum_{i \in N \setminus \{s,t\}} \alpha_i x_i x_s x_t \leq (\beta - \alpha_s - \alpha_t) x_s x_t. \qquad (12)$$

Now, using the estimators given in Example 1 in Subsection 3.1, we can convert this cubic inequality into a weaker quadratic inequality, by replacing $x_i x_s x_t$ with

- $x_ix_s + x_sx_t - x_s$ when $i \in S^+$,
- $x_ix_t + x_sx_t - x_t$ when $i \in T^+$,
- $x_ix_s + x_ix_t - x_i$ when $i \in W^+$,
- 0 when $i \in R^+$,
- x_ix_s when $i \in S^-$,
- x_ix_t when $i \in T^-$,
- $1 - x_i - x_s - x_t + x_ix_s + x_ix_t + x_sx_t$ when $i \in W^-$,
- x_sx_t when $i \in R^-$.

Doing this, re-arranging, and then replacing x_ix_j with y_{ij} everywhere yields the inequality (11). □

Theorem 3 *Let α, β, s, t, S, T, W and R be defined as in Theorem 2. Then the following 'mixed (s,t)' inequalities are valid for all possible s, t, S, T and W:*

$$\sum_{i \in W} \alpha_i x_i + \sum_{i \in T \cup R} \alpha_i y_{is} - \sum_{i \in T \cup W} \alpha_i y_{it} \leq \alpha(W^+) + \big(\beta - \alpha(\{s\} \cup S^- \cup W^+)\big) x_s - \alpha(W^+ \cup T^-) x_t + \big(\alpha(\{s\} \cup S^- \cup T^- \cup W^+ \cup R^+) - \beta\big) y_{st}. \quad (13)$$

Proof. We construct the following valid cubic inequality:

$$(\beta - \alpha^T x) x_s (1 - x_t) \geq 0,$$

and re-write it as:

$$\sum_{i \in N \setminus \{s,t\}} \alpha_i (x_ix_s - x_ix_sx_t) \leq (\beta - \alpha_s)(x_s - x_sx_t). \quad (14)$$

The rest of the proof is similar to that of Theorem 2. □

Theorem 4 *Let α, β, s, t, S, T, W and R be defined as in Theorem 2. Then the following 'reverse (s,t)' inequalities are valid for all possible s, t, S, T and W:*

$$\sum_{i \in S \cup T \cup R} \alpha_i x_i - \sum_{i \in T \cup R} \alpha_i y_{is} - \sum_{i \in S \cup R} \alpha_i y_{it} \leq \beta - \alpha(W^-) + \big(\alpha(S^+ \cup W^-) - \beta\big) x_s + \big(\alpha(T^+ \cup W^-) - \beta\big) x_t + \big(\beta - \alpha(S^+ \cup T^+ \cup W^- \cup R^-)\big) y_{st}. \quad (15)$$

Proof. We construct the following valid cubic inequality

$$(\beta - \alpha^T x)(1 - x_s)(1 - x_t) \geq 0,$$

and re-write it as:

$$\sum_{i \in N \setminus \{s,t\}} \alpha_i (x_i - x_i x_s - x_i x_t + x_i x_s x_t) \leq \beta(1 - x_s - x_t + x_s x_t). \qquad (16)$$

The rest of the proof is similar to that of Theorem 2. □

Note that, in each of the three theorems, there is an exponentially-large number of ways of selecting S, T and W for a given pair (s, t) and a given linear constraint $\alpha^T x \leq \beta$.

We remark that the particular case of the mixed (s, t) inequalities obtained when $S = T = R = \emptyset$ and $\alpha_i = 1$ for all i was previously given in [15], in the context of the 'edge-weighted clique' problem.

4.3 Cut strengthening

When binary variables are present, it is sometimes possible to strengthen the inequalities obtained via our approach, using a disjunctive argument. We illustrate this with the three families of (s, t) inequalities presented in the previous subsection.

Theorem 5 *Let s, t, S, T, W, R, α and β be defined as in Theorem 2. Also let*

$$U_s = \min\left\{\alpha(S^+), \beta - \alpha(\{s\} \cup T^- \cup W^- \cup R^-)\right\}$$

and

$$U_t = \min\left\{\alpha(T^+), \beta - \alpha(\{t\} \cup S^- \cup W^- \cup R^-)\right\}.$$

Then the following 'strong (s, t)' inequality is valid for Q:

$$\sum_{i \in S \cup W} \alpha_i y_{is} + \sum_{i \in T \cup W} \alpha_i y_{it} - \sum_{i \in W} \alpha_i x_i \leq -\alpha(W^-) + (U_s + \alpha(W^-))x_s$$

$$+ (U_t + \alpha(W^-))x_t + \left(\beta - U_s - U_t - \alpha(\{s,t\} \cup W^- \cup R^-)\right) y_{st}. \qquad (17)$$

Proof. Since $\{s, t\} \subset B$, every feasible vector x satisfies the following four-term disjunction:

$$(x_s = x_t = 0) \vee (x_s = 1 \wedge x_t = 0) \vee (x_s = 0 \wedge x_t = 1) \vee (x_s = x_t = 1). \quad (18)$$

Let LHS denote the left-hand side of the inequality (17). Consider each of the four terms of the disjunction (18), and note that:

1. If $x_s = x_t = 0$, then LHS reduces to $-\sum_{i \in W} \alpha_i x_i$, and therefore cannot exceed $-\alpha(W^-)$.

2. If $x_s = 1$ but $x_t = 0$, then LHS reduces to $\sum_{i \in S} \alpha_i x_i$, and therefore cannot exceed U_s.

3. If $x_s = 0$ but $x_t = 1$, then LHS reduces to $\sum_{i \in T} \alpha_i x_i$, and therefore cannot exceed U_t.

4. If $x_s = x_t = 1$, then LHS reduces to $\sum_{i \in S \cup T \cup W} \alpha_i x_i$, and therefore cannot exceed $\beta - \alpha(\{s,t\} \cup R^-)$.

Note also that the variable y_{st} takes the value 1 if and only if $x_s = x_t = 1$, i.e., only if the fourth term of the disjunction is satisfied. Therefore, LHS cannot exceed the right-hand side of (17) in any of the four cases. □

Theorem 6 *Let s, t, S, T, W, R, α and β be defined as in Theorem 2. Also let*

$$U''_{\emptyset} = \min\left\{\alpha(W^+), \beta - \alpha(S^- \cup T^- \cup R^-)\right\}$$

and

$$U''_{st} = \min\left\{\alpha(R^+), \beta - \alpha(\{s,t\} \cup T^- \cup W^- \cup S^-)\right\}.$$

Then the following 'strong mixed (s,t)' inequality is valid for Q:

$$\sum_{i \in W} \alpha_i x_i + \sum_{i \in T \cup R} \alpha_i y_{is} - \sum_{i \in T \cup W} \alpha_i y_{it} \leq U''_{\emptyset} + \left(\beta - \alpha(\{s\} \cup S^-) - U''_{\emptyset}\right) x_s$$

$$- (U''_{\emptyset} + \alpha(T^-))x_t + (U''_{st} + U''_{\emptyset} - \beta + \alpha(\{s\} \cup S^- \cup T^-)y_{st}. \tag{19}$$

Proof. The proof is similar to that of Theorem 5, except that:

1. If $x_s = x_t = 0$, then LHS $\leq U''_{\emptyset}$.
2. If $x_s = 1$ but $x_t = 0$, then LHS $\leq \beta - \alpha_s - \alpha(S^-)$.
3. If $x_s = 0$ but $x_t = 1$, then LHS $\leq -\alpha(T^-)$.
4. If $x_s = x_t = 1$, then LHS $\leq U''_{st}$. □

Theorem 7 *Let s, t, S, T, W, R, α and β be defined as in Theorem 2. Also let*

$$U'_s = \min\left\{\alpha(S^+), \beta - \alpha(\{s\} \cup T^- \cup W^- \cup R^-)\right\}$$

and

$$U'_t = \min\left\{\alpha(T^+), \beta - \alpha(\{t\} \cup S^- \cup W^- \cup R^-)\right\}.$$

Then the following 'strong reverse (s,t)' inequality is valid for Q:

$$\sum_{i \in S \cup T \cup R} \alpha_i x_i - \sum_{i \in T \cup R} \alpha_i y_{is} - \sum_{i \in S \cup R} \alpha_i y_{it} \leq (\beta - \alpha(W^-)) + (U'_s - \beta + \alpha(W^-))x_s$$

$$+ (U'_t - \beta + \alpha(W^-))x_t + (\beta - U'_s - U'_t - \alpha(R^- \cup W^-))y_{st}. \tag{20}$$

Proof. The proof is similar to that of Theorem 5, except that:

1. If $x_s = x_t = 0$, then LHS $\leq \beta - \alpha(W^-)$.
2. If $x_s = 1$ but $x_t = 0$, then LHS $\leq U'_s$.
3. If $x_s = 0$ but $x_t = 1$, then LHS $\leq U'_t$.
4. If $x_s = x_t = 1$, then LHS $\leq -\alpha(R^-)$. □

5 Separation

For a given family of cutting planes, a *separation algorithm* is a procedure that takes a fractional LP solution (x^*, y^*) as input, and outputs a violated inequality in that family, if one exists [11]. In this section, we present separation algorithms for the cutting planes presented in the previous section.

5.1 Separation for the unstrengthened inequalities

First, we give a positive result for the case in which d and d' are fixed.

Theorem 8 *Let d and d' be fixed. Suppose that a collection of degree-d estimators has already been computed and stored for the degree-d' monomial $\prod_{i=1}^{d'} x_i$ (using, e.g., the methods described in Section 3). Then the separation problem for the cutting planes presented in Subsection 4.1 can be solved in $\mathcal{O}(n^{d'}(n+m)^{d'})$ time.*

Proof. To begin with, take each of the degree-d estimators in the collection, and convert it into a linear estimator, by substituting monomials of degree greater than 1 with the corresponding y variables. This can be done in constant time for fixed d and d'.

Now, the number of possible candidates for the triple (S, T, M') (see Subsection 4.1) is maximised when all of the constraints (8) are linear, in which case, it achieves the value:

$$\sum_{k=0}^{d'} 2^k \binom{n}{k} \binom{m}{d'-k}.$$

This is $\mathcal{O}((n+m)^{d'})$ when d' is fixed. Each of the resulting $\mathcal{O}((n+m)^{d'})$ degree-d' inequalities can be expressed as a weighted sum of $\mathcal{O}(n^{d'})$ monomials in $\mathcal{O}(n^{d'})$ time. For each of the resulting simplified inequalities, we can obtain a most-violated cutting plane (if any exists) in $\mathcal{O}(n^{d'})$ time as follows. Consider each monomial in turn. If it has a positive coefficient on the left-hand side of the inequality, then replace it with the linear over-estimator that has the smallest value at (x^*, y^*). If it has a negative left-hand side coefficient, replace it with the linear under-estimator that has the largest value at (x^*, y^*). □

For the particular case of the (s,t) inequalities, we have:

Theorem 9 *The separation problems for the (s,t) inequalities (11) can be solved in $\mathcal{O}(mn|B|^2)$ time.*

Proof. There are m choices for the linear inequality $\alpha^T x \leq \beta$, and $\mathcal{O}(|B|^2)$ choices for the pair $\{s,t\}$. Once the inequality and the pair $\{s,t\}$ have been fixed, it suffices to assign each $i \in N \setminus \{s,t\}$ to one of the sets S, T, W or R, in the way which maximises the violation at (x^*, y^*). This can be done in $\mathcal{O}(n)$ time. □

The separation problems for the mixed (s,t) and reverse (s,t) inequalities can be solved in $\mathcal{O}(mn|B|^2)$ time in a similar way.

5.2 Separation for the strengthened inequalities

It turns out that the separation problems for the strengthened inequalities presented in Subsection 4.3 can also be solved in polynomial time. The following theorem establishes this for the strong (s,t) inequalities.

Theorem 10 *The separation problem for the strong (s,t) inequalities (17) can be solved in $\mathcal{O}(mn|B|^2)$ time.*

Proof. Suppose that α, β, s and t are fixed. From the proof of Theorem 5, one sees that the inequality (17) remains valid regardless of whether one sets U_s to either $\alpha(S^+)$ or $\beta - \alpha(\{s\} \cup T^- \cup W^- \cup R^-)$, and regardless of whether one sets U_t to either $\alpha(T^+)$ or $\beta - \alpha(\{t\} \cup S^- \cup W^- \cup R^-)$. Accordingly, it suffices to devise an efficient separation algorithm for each of the four possible cases.

If we set U_s to $\alpha(S^+)$ and U_t to $\alpha(T^+)$, then the inequality (17) reduces to the original (non-strengthened) (s,t) inequality (11), for which we provided a separation algorithm in the Subsection 5.1.

Now suppose that we set U_s to $\beta - \alpha(\{s\} \cup T^- \cup W^- \cup R^-)$, but set U_t to $\alpha(T^+)$, as before. The inequality (17) can then be written as:

$$\sum_{i \in S} \alpha_i y_{is} + \sum_{i \in T^+} \alpha_i (y_{it} + y_{st} - x_t) + \sum_{i \in T^-} \alpha_i (x_s + y_{it} - y_{st}) + \sum_{i \in W^+} \alpha_i (y_{is} + y_{it} - x_i)$$

$$+ \sum_{i \in W^-} \alpha_i (1 + y_{is} + y_{it} - x_i - x_t) + \sum_{i \in R^-} \alpha_i x_s \ \leq \ (\beta - \alpha_s) x_s - \alpha_t y_{st}.$$

Now, the right-hand side is a constant for fixed α, β, s and t. We can therefore maximise the violation in $\mathcal{O}(n)$ time as follows. Consider each $i \in N \setminus \{s,t\}$ in turn. If $\alpha_i > 0$, place i into S, T or W according to which of the following quantities is largest: y^*_{is}, $y^*_{it} + y^*_{st} - x^*_t$ or $y^*_{is} + y^*_{it} - x^*_i$. If $\alpha_i < 0$, place i into S, T, W or R according to which of the following quantities is smallest: y^*_{is}, $x^*_s + y^*_{it} - y^*_{st}$, $1 + y^*_{is} + y^*_{it} - x^*_i - x^*_t$ or x^*_s.

The case in which we set U_s to $\alpha(S^+)$ and U_t to $\beta - \alpha(\{t\} \cup S^- \cup W^- \cup R^-)$ can be treated similarly, just by switching the roles of s and t.

Finally, if we set U_s to $\beta - \alpha(\{s\} \cup T^- \cup W^- \cup R^-)$ and U_t to $\beta - \alpha(\{t\} \cup S^- \cup W^- \cup R^-)$, then the inequality (17) can be written as:

$$\sum_{i \in S^+} \alpha_i y_{is} + \sum_{i \in S^-} \alpha_i (x_t + y_{is} - y_{st}) + \sum_{i \in T^+} \alpha_i y_{it} + \sum_{i \in T^-} \alpha_i (x_s + y_{it} - y_{st})$$

$$+ \sum_{i \in W^+} \alpha_i (y_{is} + y_{it} - x_i) + \sum_{i \in W^-} \alpha_i (1 - x_i + y_{is} + y_{it} - y_{st}) + \sum_{i \in R^-} \alpha_i (x_s + x_t - y_{st})$$
$$\leq (\beta - \alpha_s) x_s + (\beta - \alpha_t) x_t - \beta y_{st}.$$

To maximise the violation in this case, one must do the following for each $i \in N \setminus \{s, t\}$. If $\alpha_i > 0$, place i into S, T or W according to which of the following quantities is largest: y^*_{is}, y^*_{it} or $y^*_{is} + y^*_{it} - x^*_i$. If $\alpha_i < 0$, place i into S, T, W or R according to which of the following quantities is smallest: $x^*_t + y^*_{is} - y^*_{st}$, $x^*_s + y^*_{it} - y^*_{st}$, $1 - x^*_i + y^*_{is} + y^*_{it} - y^*_{st}$ or $x^*_s + x^*_t - y^*_{st}$. □

One can also solve the separation problems for the strong mixed (s, t) inequalities (19) and strong reverse (s, t) inequalities (20) in $\mathcal{O}(mn|B|^2)$ time, in a similar way. We omit the details for brevity.

6 Computational Experiments

In order to explore the potential of the new cutting planes, we applied our method to the *quadratic knapsack problem* (QKP). The QKP, introduced by Gallo *et al.* [10], takes the form:

$$\max \left\{ x^T Q x : \; w^T x \leq c, \; x \in \{0,1\}^n \right\},$$

where $Q \in \mathbb{Z}_+^{n \times n}$ is a matrix of profits, $w \in \mathbb{Z}_+^n$ is a vector of *weights* and $c \in \mathbb{Z}_+$ is the *knapsack capacity.* The QKP is strongly $\mathcal{NP}$-hard. A good survey on the QKP is given by Pisinger [20].

To create the QKP instances, we followed the scheme proposed in [10]. Each weight w_i is uniformly distributed between 1 and 50. The capacity c is uniformly distributed between 50 and $\sum_{i \in N} w_i$. Then, for a given choice of a *density* parameter Δ%, each profit term q_{ij} is set to zero with probability (100 - Δ)%, and uniformly distributed between 1 and 100 with probability Δ%. It is known (see, e.g., Caprara *et al.* [5]) that the QKP tends to increase in difficulty as the density decreases. We created 5 such instances for each combination of $n \in \{10, 20, \ldots, 100\}$ and $\Delta \in \{100\%, 75\%, 50\%, 25\%\}$, making 200 instances in total.

We then coded an LP-based cutting-plane algorithm, in the C programming language and compiled with *gcc*, to compute various LP relaxations. (This algorithm calls on functions from the IBM CPLEX Callable Library to solve the LPs.) For each instance, we solved the following four relaxations:

- The standard (first-level) RLT relaxation.
- The RLT relaxation augmented with the following *triangle* inequalities, which were shown to define facets of the Boolean quadric polytope by Padberg [18]:

$$\begin{aligned} -x_k - y_{ij} + y_{ik} + y_{jk} &\leq 0 \quad (\forall i, j, k \in N) \\ x_i + x_j + x_k - y_{ij} - y_{ik} - y_{jk} &\leq 1 \quad (\forall i, j, k \in N). \end{aligned}$$

- The RLT relaxation augmented with the three kinds of strengthened (s, t) inequalities (17)–(20).
- The RLT relaxation augmented with both triangle and strengthened (s, t) inequalities.

We denote these relaxations by 'RLT', 'RLT+T', 'RLT+ST' and 'All', respectively.

Each instance was also solved to proven optimality, using the code described in [5]. (This was kindly given to us by the late Alberto Caprara.) We then computed, for each instance and each relaxation, the *percentage integrality gap*, which is the gap between the upper bound and the optimum, expressed as a percentage of the optimum.

Table 1 shows, for each value of Δ and n, and each of the four relaxations, the percentage integrality gap and the computational time (in seconds), averaged over the 5 instances concerned. We see that both the 'RLT+T' and 'RLT+ST' gaps are significantly smaller than the 'RLT' gap, indicating that both families of cutting planes are useful. This is especially true for the more difficult sparse instances, where the initial gap is typically reduced by 50% or more. Moreover, the 'RLT+ST' gap is invariably smaller than the 'RLT+T' gap, indicating that the (s, t) inequalities are more effective than the triangle inequalities. As further evidence of this claim, note that the 'All' gaps are very close to the 'RLT+ST' gaps, which means that the addition of triangle inequalities to the 'RLT+ST' relaxation is of little use.

On the other hand, we see that running times are significantly longer when (s, t) inequalities are included. This is because the separation algorithm for (s, t) inequalities typically found a very large number of violated inequalities in the early iterations. In our implementation, all violated inequalities were added to the LP relaxation. We imagine that more sophisticated cut selection strategies might lead to improved running times. Note also that times for the 'All' relaxation are typically smaller than for the 'RLT+ST' relaxation. The explanation for this is that the (s, t) separation algorithm was called only when no violated triangle inequalities could be found. This strategy seemed to improve overall convergence.

Instances		RLT		RLT + T		RLT + ST		All	
Δ	n	Gap	Time	Gap	Time	Gap	Time	Gap	Time
100	10	6.33	0.020	5.35	0.037	4.01	0.132	4.02	0.077
	20	3.76	0.025	3.28	0.057	2.95	0.404	2.95	0.243
	30	3.44	0.051	3.26	0.096	2.79	1.543	2.79	1.130
	40	1.30	0.238	1.25	0.235	1.21	1.989	1.21	1.569
	50	1.38	0.254	1.34	0.432	1.28	11.237	1.26	19.675
	60	0.79	0.367	0.79	0.394	0.75	5.308	0.75	3.173
	70	0.63	0.714	0.63	0.753	0.60	10.827	0.60	7.562
	80	0.38	1.308	0.38	1.494	0.36	11.539	0.36	10.433
	90	0.36	2.217	0.36	2.321	0.34	12.892	0.34	11.430
	100	0.17	3.214	0.17	3.453	0.16	15.980	0.16	16.821
75	10	4.61	0.015	3.88	0.044	3.27	0.073	3.27	0.037
	20	3.31	0.042	3.18	0.052	2.23	0.307	2.23	0.129
	30	0.66	0.086	0.57	0.219	0.55	1.864	0.55	1.117
	40	1.72	0.190	1.49	0.365	1.41	6.275	1.41	2.855
	50	0.65	0.545	0.63	0.957	0.60	5.529	0.60	3.086
	60	0.79	0.367	0.79	0.394	0.75	5.308	0.75	3.173
	70	0.92	0.936	0.90	1.677	0.77	18.190	0.77	9.341
	80	0.61	3.591	0.54	7.296	0.46	26.265	0.46	49.404
	90	0.78	4.464	0.68	17.420	0.60	417.182	0.60	286.721
	100	0.17	7.786	0.17	14.618	0.14	49.509	0.14	152.177
50	10	7.60	0.029	6.75	0.032	4.28	0.223	4.27	0.142
	20	3.31	0.042	3.18	0.052	2.23	0.307	2.23	0.129
	30	2.29	0.130	1.12	0.309	1.03	3.910	1.03	1.360
	40	1.84	0.249	1.68	0.806	1.55	13.776	1.55	10.298
	50	1.03	0.501	0.44	1.589	0.41	12.664	0.41	2.459
	60	2.31	0.854	0.56	1.951	0.45	130.791	0.45	54.400
	70	2.22	1.893	0.46	3.974	0.38	225.525	0.38	79.566
	80	2.09	3.743	0.38	9.489	0.27	724.311	0.27	112.722
	90	0.73	5.522	0.28	13.023	0.22	327.551	0.22	144.888
	100	0.77	8.722	0.24	39.562	0.21	543.732	0.20	456.292
25	10	8.76	0.058	8.29	0.025	1.28	0.112	1.28	0.196
	20	2.08	0.047	1.19	0.067	0.85	1.076	0.85	0.974
	30	1.65	0.159	1.19	0.250	1.04	5.147	1.04	4.194
	40	4.46	0.214	2.28	0.545	1.85	7.564	1.85	6.132
	50	3.64	0.510	1.25	1.039	0.92	14.102	0.92	9.888
	60	3.36	0.551	1.47	1.130	0.97	68.095	0.97	31.644
	70	0.38	1.270	0.34	1.359	0.27	129.260	0.27	124.606
	80	1.03	2.037	0.43	4.837	0.36	481.676	0.35	178.357
	90	1.75	4.591	1.72	14.756	0.72	160.507	0.22	116.680
	100	0.69	3.757	0.23	9.358	0.17	423.107	0.16	173.461

Table 1: Percentage integrality gaps for QKP instances

7 Concluding Remarks

We have introduced new techniques for constructing low-degree polynomial over- and under-estimators for monomials of higher degree, and shown how they can be used to generate cutting planes for RLT relaxations of mixed 0-1 polynomial programs. The computational results, though restricted to the level-1 RLT relaxation of the QKP, indicate that the cutting planes can be effective at reducing the integrality gap.

A natural topic for future research is the efficient incorporation of the cutting planes into an exact algorithm for mixed 0-1 polynomial programs of low degree. Another is the question of whether some estimators are 'better' than others, in the sense of being more likely to lead to violated cuts. If that were the case, then one could reduce the time taken by the separation algorithms by eliminating some estimators from consideration.

References

[1] W.P. Adams, M. Guignard, P.M. Hahn & W.L. Hightower (2007) A level-2 reformulation-linearization technique bound for the quadratic assignment problem. *Eur. J. Oper. Res.*, 180, 983–996.

[2] F. Barahona, M. Jünger & G. Reinelt (1989) Experiments in quadratic 0-1 programming. *Math. Program.*, 44, 127–137.

[3] P. Bonami & M. Minoux (2005) Using rank-1 lift-and-project closures to generate cuts for 01 MIPs, a computational investigation. *Discr. Opt.*, 2, 288–307.

[4] A. Billionnet & F. Calmels (1996) Linear programming for the 0-1 quadratic knapsack problem. *Eur. J. Oper. Res.*, 92, 310–325.

[5] A. Caprara, D. Pisinger & P. Toth (1998) Exact solution of the quadratic knapsack problem. *INFORMS J. Comput.*, 11, 125–137.

[6] T. Christof & A. Loebl, `PORTA` (polyhedron representation transformation algorithm). Software package, available for download at `http://www.iwr.uni-heidelberg.de/groups/comopt/software`

[7] M.M. Deza & M. Laurent (1997) *Geometry of Cuts and Metrics.* Berlin: Springer.

[8] C.A. Floudas (1999) *Deterministic Global Optimization: Theory, Algorithms and Applications.* Dordrecht: Kluwer.

[9] K. Fukuda, `cdd` (C double description program). Software package, available for download at:
`http://www.cs.mcgill.ca/~fukuda/soft/cdd_home`

[10] G. Gallo, P.L. Hammer & B. Simeone (1980) Quadratic knapsack problems. *Math. Program. Stud.*, 12, 132–149.

[11] M. Grötschel, L. Lovász & A.J. Schrijver (1988) *Geometric Algorithms and Combinatorial Optimization.* Berlin: Springer.

[12] P.M. Hahn, Y.-R. Zhu, M. Guignard, W.L. Hightower & M.J. Saltzman (2012) A level-3 reformulation-linearization technique-based bound for the quadratic assignment problem. *INFORMS J. Comput.*, 24, 202–209.

[13] C. Helmberg & F. Rendl (1998) Solving quadratic (0,1)-programs by semidefinite programs and cutting planes. *Math. Program.*, 82, 291–315.

[14] C. Helmberg, F. Rendl & R. Weismantel (2000) A semidefinite programming approach to the quadratic knapsack problem. *J. Comb. Optim.*, 4, 197–215.

[15] M. Hunting, U. Faigle & W. Kern (2001) A Lagrangean relaxation approach to the edge-weighted clique problem. *Eur. J. Oper. Res.*, 131, 119–131.

[16] J.B. Lasserre (2001) Global optimization with polynomials and the problem of moments. *SIAM J. Opt.*, 11, 796-817.

[17] L. Lovász & A.J. Schrijver (1991) Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1, 166–190.

[18] M.W. Padberg (1989) The Boolean quadric polytope: some characteristics, facets and relatives. *Math. Program.*, 45, 139–172.

[19] P.A. Parrilo (2003) Semidefinite programming relaxations for semialgebraic problems. *Math. Prog.*, 96, 293-320.

[20] D. Pisinger (2007) The quadratic knapsack problem – a survey. *Discr. Appl. Math.*, 155, 623-648.

[21] H. Saito, T. Fujie, T. Matsui & S. Matuura (2009) A study of the quadratic semi-assignment polytope. *Discr. Opt.*, 6, 37–50.

[22] H.D. Sherali (2007) RLT: A unified approach for discrete and continuous nonconvex optimization. *Ann. Oper. Res.*, 149, 185-193.

[23] H.D. Sherali & W. Adams (1990) A hierarchy of relaxations between the continuous and convex hull representations for 0-1 programming problems. *SIAM J. Discr. Math.*, 3, 411–430.

[24] H.D. Sherali & W.P. Adams (1994) A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discr. Appl. Math.*, 52, 83–106.

[25] H.D. Sherali & W.P. Adams (1998) *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems.* Dordrecht: Kluwer.

[26] H.D. Sherali & Y. Lee (1996) Tighter representations for set partitioning problems. *Discr. Appl. Math.*, 68, 153–167.

[27] H.D. Sherali & C.H. Tuncbilek (1992) A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. *J. Glob. Optim.*, 2, 101–112.

[28] M. Tawarmalani & N.V. Sahinidis (2003) *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming.* Dortrecht: Kluwer.

[29] Y. Yajima & T. Fujie (1998) A polyhedral approach for nonconvex quadratic programming problems with box constraints. *J. Glob. Optim.*, 13, 151–170.