Polynomial-size formulations and relaxations for the quadratic multiple knapsack problem

Polynomial-Size Formulations and Relaxations for the Quadratic Multiple Knapsack Problem

Laura Galli[1] Silvano Martello [2 3] Carlos Rey [2] Paolo Toth [2]

September 19, 2020

Abstract

The *Quadratic Multiple Knapsack Problem* generalizes, simultaneously, two well-known combinatorial optimization problems that have been intensively studied in the literature: the (single) Quadratic Knapsack Problem and the Multiple Knapsack Problem. The only exact algorithm for its solution uses a formulation based on an exponential-size number of variables, that is solved via a Branch-and-Price algorithm. This work studies polynomial-size formulations and upper bounds. We derive linear models from classical reformulations of 0-1 quadratic programs and analyze theoretical properties and dominances among them. We define surrogate and Lagrangian relaxations, and we compare the effectiveness of the Lagrangian relaxation when applied to a quadratic formulation and to a Level 1 reformulation linearization that leads to a decomposable structure. The proposed methods are evaluated through extensive computational experiments.

Keywords: Combinatorial Optimization; Quadratic Multiple Knapsack; Binary Quadratic Programming; Lagrangian Relaxation; Reformulation Linearization Technique.

1 Introduction

The (linear) *Multiple Knapsack Problem* (MKP) has been intensively studied in the last 40 years (see the relative chapters in the monographs by Martello and Toth [22] and Kellerer, Pferschy, and Pisinger [19]).

The MKP is defined on n *items* and m *knapsacks*. Each knapsack $k \in M = \{1, \ldots, m\}$ has a *capacity* C_k. Each item $i \in N = \{1, \ldots, n\}$ has a *profit* p_i and a *weight* w_i. The objective is to select m disjoint subsets of items to be assigned to the knapsacks so that the total weight assigned to each knapsack does not exceed its capacity and the total profit of

[1]Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy. E-mail: laura.galli@unipi.it

[2]DEI "Guglielmo Marconi", Università di Bologna, Viale Risorgimento 2, 40136 Bologna, Italy. E-mail: {silvano.martello,carlos.rey2,paolo.toth}@unibo.it

[3]Corresponding author

the selected items is maximized. By introducing nm binary variables x_{ik} ($i \in N, k \in M$) taking the value 1 if and only if item i is assigned to knapsack k, the problem is formally defined by the *0-1 Linear Program*

$$\max \quad \sum_{i=1}^{n} \sum_{k=1}^{m} p_i x_{ik} \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_i x_{ik} \leq C_k \qquad (k \in M) \tag{2}$$

$$\sum_{k=1}^{m} x_{ik} \leq 1 \qquad (i \in N) \tag{3}$$

$$x \in \{0,1\}^{n \times m}, \tag{4}$$

where (2) and (3) are the classical *capacity* and *cardinality constraints*, respectively. The problem is a generalization of the famous *0-1 Knapsack Problem* (KP), in which $m = 1$. While the KP is ordinary $\mathcal{NP}$-hard and admits pseudo-polynomial time dynamic programming algorithms, the MKP is known to be strongly $\mathcal{NP}$-hard, as it can be seen by transformation from 3-partition (see, e.g., [22]).

Although a problem with a similar flavor had been considered by Witzgall [35] in 1975, to the best of our knowledge, the first *quadratic* version of a knapsack problem was introduced by Gallo, Hammer, and Simeone [11] in 1980. In the (single) *Quadratic Knapsack Problem* (QKP) one is given a knapsack with capacity C and n items. Each item $i \in N$ has a profit p_i and a weight w_i. In addition, each pair of distinct items i, j gives a profit p_{ij} if both belong to the solution. (It is assumed that $p_{ji} = p_{ij}$.) The objective is to select a subset of items so that the total weight does not exceed the capacity, and the total profit (sum of the profits of the selected items and of their pairwise profits) is maximized. Formally,

$$\max \quad \sum_{i=1}^{n} p_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} p_{ij} x_i x_j \tag{5}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_i x_i \leq C \tag{6}$$

$$x \in \{0,1\}^n, \tag{7}$$

where x_i is a binary variable taking the value 1 if and only if item i is selected. We refer the reader to monograph [19] (Chapter 12) for an extensive treatment of the QKP until 2003, and to Billionnet and Soutif [3], Julstrom [18], Pisinger [25], Pisinger, Rasmussen, and Sandvik [26], Pulikanti and Singh [27], and Lalla-Ruiz, Segredo, and Voß [20] for later studies.

The *Quadratic Multiple Knapsack Problem* (QMKP), to which this paper is devoted, was first introduced by Hiley and Julstrom [17], and ideally combines the objective function of the QKP and the constraints of the MKP. We have n items and m knapsacks. Each knapsack $k \in M$ has a capacity $C_k \in \mathbb{Z}_+$, each item $i \in N$ has a profit $p_i \in \mathbb{Z}_+$ and a weight $w_i \in \mathbb{Z}_+$. Each pair of distinct items i, j produces a profit $p_{ij} \in \mathbb{Z}_+$ (with

$p_{ji} = p_{ij}$) if both are assigned to the same knapsack. The objective is to select m disjoint subsets of items to be assigned to the knapsacks, so that the total weight assigned to each knapsack does not exceed its capacity, and the total profit (sum of the profits of the selected items and of the pairwise profits of items assigned to the same knapsack) is maximized. Formally,

$$\max \quad \sum_{i=1}^{n}\sum_{k=1}^{m} p_i x_{ik} + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{m} p_{ij} x_{ik} x_{jk} \tag{8}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_i x_{ik} \leq C_k \qquad (k \in M) \tag{9}$$

$$\sum_{k=1}^{m} x_{ik} \leq 1 \qquad (i \in N) \tag{10}$$

$$x \in \{0,1\}^{n \times m}, \tag{11}$$

where x is defined as for the MKP. As the QKP is the special case of the QMKP arising when $m = 1$, the QMKP is strongly $\mathcal{NP}$-hard. In addition, all computational experiments reported so far in the literature indicate that it is extremely challenging to solve in practice.

Owing to its many practical applications, that range from project management to capital budgeting to product-distribution system design, as well as to its mathematical structure borrowing from well-studied combinatorial problems, the QMKP has received increasing attention in the literature over the last fifteen years. In their seminal work, Hiley and Julstrom [17] presented the first 60 benchmark instances and three heuristics. Their paper started a stream of research based on *meta-heuristic* techniques, that includes a genetic algorithm by Singh and Baghel [31], an artificial bee colony algorithm by Sundar and Singh [33], and a memetic algorithm by Soak and Lee [32]. More recently, Garcia-Martinez et al. [12, 13] presented a strategic oscillation algorithm and a Tabu-enhanced iterated greedy approach. Chen and Hao [6] and Chen et al. [7] used, respectively, an iterative response threshold search algorithm, and an evolutionary path relinking approach, for which recent variations have been proposed by Qin et al. [28] and Tlili et al. [34]. Despite this growing stream of research on heuristics, no exact method for the QMKP was proposed in the literature until the recent contribution by Bergman [2], who presented the first exact solution approach to the QMKP, that uses a formulation based on an exponential-size number of variables, solved via a Branch-and-Price algorithm.

While the literature has been so far concentrating on exponential-size formulations and meta-heuristic approaches, our contribution consists of investigating several polynomial-size formulations, aiming at devising the relaxations that produce good upper bounds in reasonable computing times. In particular, our goal is to compare the effectiveness of the Lagrangian relaxation when applied to the quadratic formulation (8)-(11) and to a Level 1 Reformulation Linearization, that leads to a decomposable structure. We present the results of computational experiments on a large set of benchmark instances.

The paper has the following structure. In Section 2, we derive several linear models for the QMKP, obtained from classical reformulations of 0-1 quadratic programs. Some theoretical properties and dominances among the resulting formulations are outlined. The surrogate relaxation of the quadratic model is discussed in Section 3. Section 4 is

concerned with the Lagrangian relaxation of the quadratic model (8)-(11) and of a linear reformulation leading to a set of independent, well-structured subproblems. Section 5 presents the computational results, and Section 6 contains some concluding remarks.

2 Linear Formulations

In this section we show how some *linear reformulations* for *0-1 Quadratic Programming* problems with linear constraints (01QP) can be specialized for the QMKP.

2.1 Classical Linear Formulations

In 1959 Fortet [8] proved that any integer-valued algebraic function can be transformed into a linear function by introducing auxiliary binary variables and linear *linking constraints*. In 1974 the idea was independently re-discovered and developed by Glover and Woolsey [15] for 01QP. A direct application to the QMKP would result in *4-index* variables, each representing the product $x_{ik}x_{j\ell}$ for $i, j \in N$ and $k, \ell \in M$. We can observe, however, that our objective function (8) only includes products involving the same knapsack index, so it is sufficient to introduce *3-index* binary variables $\hat{y}_{ijk}$, taking the value one if and only if items i and j are assigned to the same knapsack k:

$$\hat{y}_{ijk} = x_{ik}x_{jk} \text{ for } i \in N \setminus \{n\}, j \in N \ (j > i), k \in M. \tag{12}$$

The *Fortet-Glover-Woolsey* (FGW) formulation for the QMKP is

$$\begin{aligned}
\text{(FGW)} \quad \max \quad & \sum_{i=1}^{n}\sum_{k=1}^{m} p_i x_{ik} + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{m} p_{ij}\hat{y}_{ijk} && && (13)\\
\text{s.t.} \quad & \hat{y}_{ijk} \le x_{ik} && (i \in N \setminus \{n\}, j \in N(j > i), k \in M) && (14)\\
& \hat{y}_{ijk} \le x_{jk} && (i \in N \setminus \{n\}, j \in N(j > i), k \in M) && (15)\\
& \hat{y}_{ijk} \ge x_{ik} + x_{jk} - 1 && (i \in N \setminus \{n\}, j \in N(j > i), k \in M) && (16)\\
& \hat{y}_{ijk} \in \{0, 1\} && (i \in N \setminus \{n\}, j \in N(j > i), k \in M) && (17)\\
& (9), (10), (11).
\end{aligned}$$

Constraints (14) and (15) ensure that $\hat{y}_{ijk}$ takes the value 0 when at least one of the two associated variables is 0. Constraints (16) force $\hat{y}_{ijk}$ to take the value 1 when both associated variables are 1.

We next show that an equivalent formulation can be obtained by removing constraints (16) and (17):

Lemma 1 *The optimal solution to the LP relaxation of* FGW *does not change if constraints* (16) *are removed.*

Proof. Let $(x^*, \hat{y}^*)$ be an optimal solution to the LP relaxation of FGW *without inequalities* (16). The second term of the objective function maximizes a linear function of $\hat{y}$ with coefficients $p_{ij} \ge 0$. It follows that *every variable* $\hat{y}^*_{ijk}$ will take the largest possible value, and hence, from (14)-(15), $\hat{y}^*_{ijk} = \min\{x^*_{ik}, x^*_{jk}\}$. Since $x^*_{ik} \le 1 \ \forall i \in N$ and $k \in M$, we have $\min\{x^*_{ik}, x^*_{jk}\} \ge x^*_{ik} + x^*_{jk} - 1$. □

Corollary 1 *If the LP relaxation of FGW satisfies* (11)*, constraints* (17) *are automatically satisfied.*

Proof. Assume that the optimal solution $(x^*, \hat{y}^*)$ to the LP relaxation of FGW satisfies $x^*_{ik} \in \{0,1\}\ \forall i \in N$ and $k \in M$. From the proof of Lemma 1 we have $\hat{y}^*_{ijk} = \min\{x^*_{ik}, x^*_{jk}\}$, and hence $\hat{y}^*_{ijk} \in \{0,1\}$. □

Proposition 1 *Constraints* (16) *and* (17) *are redundant for* FGW *and for its LP relaxation.*

Proof. Immediate from Lemma 1 and Corollary 1. □

Model FGW has $O(n^2m)$ variables and constraints. A more compact, $O(nm)$, linear model for 01QP was proposed in 1975 by Glover [14], who introduced, for each original variable x_{ik}, a new continuous variable z_{ik} to represent its contribution to the objective function. For the QMKP, let us define, for each $i \in N$ and $k \in M$,

$$g_{ik}(x) = \begin{cases} p_i + \sum_{j=i+1}^{n} p_{ij} x_{jk} & \text{if } i \in N \setminus \{n\}; \\ p_n & \text{if } i = n. \end{cases} \tag{18}$$

The contribution of x_{ik} to the objective function is then

$$z_{ik} = g_{ik}(x) x_{ik} \quad (i \in N, k \in M), \tag{19}$$

and observe that z_{ik} will always take integer values as the profits are assumed to be integer. The resulting *Glover* model (GLOV) for the QMKP is

$$\text{(GLOV)} \quad \max \quad \sum_{i=1}^{n} \sum_{k=1}^{m} z_{ik} \tag{20}$$

$$\text{s.t.} \quad L_i x_{ik} \leq z_{ik} \leq U_i x_{ik} \qquad (i \in N, k \in M) \tag{21}$$

$$g_{ik}(x) - U_i(1 - x_{ik}) \leq z_{ik} \leq g_{ik}(x) - L_i(1 - x_{ik}) \quad (i \in N, k \in M) \tag{22}$$

$$(9), (10), (11),$$

where $g_{ik}(x)$ is defined in (18), while $L_i = p_i + \sum_{j=i+1}^{n} \min\{0, p_{ij}\}$, $U_i = p_i + \sum_{j=i+1}^{n} \max\{0, p_{ij}\}$ (for $i \in N \setminus \{n\}$), and $L_n = U_n = p_n$ are the smallest and largest values, respectively, that g_{ik} (and hence z_{ik}) can take. Note that, as we assume the pairwise profits p_{ij} to be non-negative, these values can be simplified to $L_i = p_i$, $U_i = p_i + \sum_{j=i+1}^{n} p_{ij}$ (for $i \in N$). Constraints (21) and (22) link variables x_{ik} and z_{ik}: constraints (21) impose $z_{ik} = 0$ when $x_{ik} = 0$, while constraints (22) impose $z_{ik} = g_{ik}(x)$ when $x_{ik} = 1$. (Note the similarity with the effect of (14)-(15) and (16), respectively.)

GLOV is indeed more compact than FGW, but, as proved by Furini and Traversi [10], its LP relaxation is weaker.

2.2 Reformulation Linearization Technique

In 1986 Adams and Sherali [1] strengthened FGW by proposing a new linearization method for 01QP. The idea was later extended to general 0-1 problems in Sherali and Adams [30]. The method, known as the *Reformulation Linearization Technique* (RLT), provides different *Levels* of representation with an increasingly stronger LP bound.

Let $\bar{n}$ denote the number of original binary variables appearing in each constraint. New quadratic constraints are added to the original formulation, to strengthen the resulting LP relaxation. At *Level 1*,

(i) each equality constraint results into $\bar{n}$ quadratic constraints obtained by multiplying it by each original binary variable;

(ii) each inequality constraint results into $2\bar{n}$ quadratic constraints obtained by multiplying it by each original binary variable and by its complement.

All the resulting quadratic constraints are then linearized by introducing auxiliary binary variables to represent the products of the original ones together with appropriate linking constraints. Higher levels are rarely used as the problem size increases so sharply that the bound computation becomes impractical.

In order to adapt the RLT to the QMKP, let us define binary variables y_{ijk} similarly to variables $\hat{y}_{ijk}$ of Section 2.1, but by considering *all* ordered pairs (i, j) with $i \neq j$, i.e.,

$$y_{ijk} = x_{ik}x_{jk} \text{ for } i \in N, j \in N \setminus \{i\}, k \in M. \tag{23}$$

A *Level 1 RLT model* (RLT1) for the QMKP is then

$$\text{(RLT1)} \quad \max \quad \sum_{i=1}^{n}\sum_{k=1}^{m} p_i x_{ik} + \frac{1}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{k=1}^{m} p_{ij} y_{ijk} \tag{24}$$

$$\text{s.t.} \quad y_{ijk} \leq x_{ik} \qquad (i \in N, j \in N\setminus\{i\}, k \in M) \tag{25}$$

$$y_{ijk} = y_{jik} \qquad (i \in N\setminus\{n\}, j \in N (j > i), k \in M) \tag{26}$$

$$y_{ijk} \geq x_{ik} + x_{jk} - 1 \qquad (i \in N, j \in N\setminus\{i\}, k \in M) \tag{27}$$

$$\sum_{j \in N\setminus\{i\}} w_j y_{ijk} \leq (C_k - w_i)\, x_{ik} \qquad (i \in N, k \in M) \tag{28}$$

$$\sum_{j \in N\setminus\{i\}} w_j (x_{jk} - y_{ijk}) \leq C_k (1 - x_{ik}) \qquad (i \in N, k \in M) \tag{29}$$

$$y_{ijk} \in \{0, 1\} \qquad (i \in N, j \in N\setminus\{i\}, k \in M) \tag{30}$$

$$(9), (10), (11).$$

The objective function (24) maps (13) in the new variable space. Constraints (25) and (27) are equivalent to (14) and (16), respectively, while (15) are implied by (25) and (26) (the latter known as *symmetry constraints*). Constraints (28) and (29) are the *RLT constraints* derived from capacity constraints (9).

We next show that if we drop the RLT constraints from RLT1, the LP relaxation of the resulting model is equivalent to the LP relaxation of FGW.

Proposition 2 *The polyhedra associated with the LP relaxation of* RLT1 *without the* RLT *constraints* (28)-(29)*, and the LP relaxation of* FGW *are isomorphic under the linear transformation*

$$\hat{y}_{ijk} = y_{ijk} = y_{jik} \quad \forall\, i, j \in N\,(j > i),\, k \in M$$

(*with* x *unchanged*).

Proof. Inequalities (25) and (26) imply $y_{ijk} \leq x_{jk}\ \forall\, i, j \in N\,(i \neq j),\, k \in M$. By observing the different j-indexing in the two objective functions, it easily follows that the two solutions produce the same value. □

If, besides removing the RLT constraints, we also remove inequalities (27), the resulting LP bound is still as strong as the one produced by the LP relaxation of FGW:

Corollary 2 *The LP relaxation of* RLT1 without constraints (27)-(29) *is equivalent to the LP relaxation of* FGW.

Proof. According to Proposition 2, the polyhedra associated with the LP relaxations of the two models are isomorphic. Lemma 1 guarantees that inequalities (27) can be removed without changing the optimal value. □

Note that RLT1 does not include the RLT constraints obtained from cardinality constraints (10). The reason for this comes from our choice of having *3-index* variables. Indeed, by applying RLT to (10), we would obtain products involving different knapsacks, for which an additional index would be needed. On the one hand, this choice makes the LP relaxation of the resulting model weaker, but, on the other hand,

(i) it produces a more compact model, of size $O(n^2 m)$ (instead of $O(n^2 m^2)$), which lends itself to a much faster computation of the resulting LP bound;

(ii) RLT1 can be effectively decomposed, as shown in the next section.

2.2.1 A decomposable Level 1 RLT model

In this section we show how, starting from RLT1, we can construct a new linear reformulation that is amenable to a *decomposable Lagrangian relaxation* (to be examined in Section 4.2) that: (i) provides a stronger bound than the one given by its continuous relaxation, and (ii) can be computed with reasonable computational effort.

Point (i) obviously requires that the relaxed model does not have the integrality property (see, e.g., Nemhauser and Wolsey [23]). An effective way to pursue point (ii) is to obtain a "decomposable" Lagrangian problem, leading to a set of independent, well-structured subproblems. We generalize the approach presented by Caprara et al. [5] for the single QKP. The same approach was later applied by Pisinger [24] to the p-dispersion problem, then generalized by Caprara [4] to 0-1 quadratic problems with linear constraints, and recently adopted by Guignard [16] for a generalization of the quadratic assignment problem. Recall that the coefficients of the quadratic terms of the objective function

(i.e., the pairwise profits p_{ij}) are assumed to be non-negative, as it normally holds for the QMKP instances considered in the literature.

Let y_{ijk} be defined as in (23). A *Decomposable Level 1 RLT model* (DRLT1) for the QMKP can be obtained from RLT1 by eliminating constraints (27) and (29), i.e.,

$$\text{(DRLT1)} \quad \max \quad \sum_{i=1}^{n}\sum_{k=1}^{m} p_i x_{ik} + \tfrac{1}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{k=1}^{m} p_{ij} y_{ijk} \tag{24}$$

$$\text{s.t.} \quad y_{ijk} \leq x_{ik} \qquad (i \in N, j \in N \setminus \{i\}, k \in M) \tag{25}$$

$$y_{ijk} = y_{jik} \qquad (i \in N\setminus\{n\}, j \in N (j > i), k \in M) \tag{26}$$

$$\sum_{j \in N\setminus\{i\}} w_j y_{ijk} \leq (C_k - w_i)\, x_{ik} \qquad (i \in N, k \in M) \tag{28}$$

$$y_{ijk} \in \{0,1\} \qquad (i \in N, j \in N\setminus\{i\}, k \in M) \tag{30}$$

$$(9), (10), (11).$$

Note that the effect of RLT1 constraints (29) was purely to strengthen the continuous relaxation of the model. Moreover, as formally proved in Proposition 1 (also see [5] and [4]), constraints (27) are redundant when the coefficients of the quadratic term are non-negative. Therefore, DRLT1 is a valid (linear) reformulation for the QMKP.

The continuous relaxation of DRLT1 is weaker than that of RLT1 but *stronger* than that of FGW. In addition, it has the advantage that *dualizing* constraints (26) results in a *decomposable* Lagrangian relaxed problem, that does not have the integrality property, as we will show in Section 4.2.

Proposition 3 *The LP relaxation of* DRLT1 *is stronger than the LP relaxation of* FGW.

Proof. From Corollary 2, the LP relaxation of FGW is as strong as the LP relaxation of DRLT1 *without constraints* (28). Therefore, it is enough to show an example where inequalities (28) improve the LP bound. Consider an instance consisting of a single knapsack of capacity $C = 8$, and three items with $w_1 = 2$, $w_2 = 8$, $w_3 = 5$, $p_1 = 1$, $p_2 = 3$, $p_3 = 1$, and pairwise profits $p_{12} = 4$, $p_{13} = 2$, $p_{23} = 2$. The optimal solution of the LP relaxation of DRLT1 is $\bar{x}_1 = \bar{x}_3 = 1$, $\bar{x}_2 = 0.125$, $\bar{y}_{13} = \bar{y}_{31} = 1$ (all other $\bar{y}$ being 0) and has value 4.375. The optimal solution of the LP relaxation of FGW is instead $\bar{x}_1 = \bar{x}_2 = \bar{x}_3 = 0.5\bar{3}$, $\bar{y}_{12} = \bar{y}_{13} = \bar{y}_{23} = 0.5\bar{3}$ and has value $6.9\bar{3}$. □

Model DRLT1 can be improved by means of the following considerations:

(i) variables y_{ijk} for which $p_{ij} = 0$ can always be set to zero;

(ii) variables y_{ijk} for which $w_i + w_j > C_k$ must take the value zero;.

(iii) due to Corollary 1, constraints (30) can be relaxed in a continuous way.

By defining

$$S_{ik} = \{j \in N \setminus \{i\} \,:\, p_{ij} > 0 \text{ and } w_i + w_j \leq C_k\} \; (i \in N, k \in M); \tag{31}$$

$$T_{ik} = \{j \in N \setminus \{i\} \,:\, w_i + w_j > C_k\} \; (i \in N, k \in M); \tag{32}$$

$$R_{ik} = \{j \in N \,:\, j > i, p_{ij} > 0, \text{ and } w_i + w_j \leq C_k\} \; (i \in N \setminus \{n\}, k \in M), \tag{33}$$

we get the *Modified Decomposable Level 1 RLT model* (MDRLT1)

$$\text{(MDRLT1)} \quad \max \quad \sum_{i=1}^{n}\sum_{k=1}^{m} p_i x_{ik} + \tfrac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{m}\sum_{j\in S_{ik}} p_{ij} y_{ijk} \tag{34}$$

$$\text{s.t.} \quad y_{ijk} \le x_{ik} \qquad (i \in N, k \in M, j \in S_{ik}) \tag{35}$$

$$y_{ijk} = y_{jik} \qquad (i \in N \setminus \{n\}, k \in M, j \in R_{ik}) \tag{36}$$

$$\sum_{j\in S_{ik}} w_j y_{ijk} \le (C_k - w_i)\, x_{ik} \qquad (i \in N, k \in M) \tag{37}$$

$$y_{ijk} \ge 0 \qquad (i \in N, k \in M, j \in S_{ik}) \tag{38}$$

$$x_{ik} + x_{jk} \le 1 \qquad (i \in N, k \in M, j \in T_{ik}) \tag{39}$$

$$(9), (10), (11).$$

3 Surrogate relaxation of the quadratic model

A classical relaxation technique for the (linear) MKP is obtained by surrogating the capacity constraints (9) with multipliers $\pi_k \ge 0$ ($k \in M$). Its popularity comes from the fact that, as proved by Martello and Toth [21], the optimal choice for the surrogate multipliers is to have them all equal to any positive number. We next show that such property carries through to the quadratic case.

For the QMKP, the surrogate relaxation of the quadratic model (8)-(11) is:

$$S(\pi) = \max \quad \sum_{i=1}^{n}\sum_{k=1}^{m} p_i x_{ik} + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{m} p_{ij} x_{ik} x_{jk} \tag{40}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} \pi_k \sum_{i=1}^{n} w_i x_{ik} \le \sum_{k=1}^{m} \pi_k C_k \tag{41}$$

$$\sum_{k=1}^{m} x_{ik} \le 1 \qquad (i \in N) \tag{42}$$

$$x \in \{0,1\}^{n\times m}. \tag{43}$$

Lemma 2 *There always exists an optimal solution to* (40)-(43) *that assigns all the selected items to the knapsack with smallest surrogate multiplier.*

Proof. Let $k^* = \arg\min\{\pi_k : k \in M\}$ and let x be a feasible solution to (40)-(43). Another feasible solution $\bar{x}$, not worse than x, can be obtained by setting $\bar{x}_{ik} = 0$ and $\bar{x}_{ik^*} = 1$ for each $i \in N$ such that $x_{ik} = 1$ and $k \ne k^*$. □

Proposition 4 *The optimal vector of multipliers for* (40)-(43) *is* $\pi_k = \bar{\pi}$ (*where* $\bar{\pi}$ *is any positive constant*) *for all* $k \in M$.

Proof. Using Lemma 2, (40)-(43) is equivalent to the (single) QKP

$$\begin{aligned} S(\pi) = \max \quad & \sum_{i=1}^{n} p_i x_{ik^*} + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} p_{ij} x_{ik^*} x_{jk^*} \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i x_{ik^*} \leq \left\lfloor \sum_{k=1}^{m} \frac{\pi_k}{\pi_{k^*}} C_k \right\rfloor \\ & x_{ik^*} \in \{0,1\} \qquad (i \in N). \end{aligned} \tag{44}$$

where k^* is the (knapsack) index corresponding to the smallest surrogate multiplier. Since $\left\lfloor \sum_{k=1}^{m} \frac{\pi_k}{\pi_{k^*}} C_k \right\rfloor \geq \sum_{k=1}^{m} C_k$, the choice $\pi_k = \bar{\pi}$ (any positive constant) for all $k \in M$ produces the minimum capacity and hence the minimum value of $S(\pi)$. □

4 Decomposable Lagrangian relaxations

In this section we study the Lagrangian relaxation when applied to the quadratic formulation (8)-(11) of Section 1, and to the DRLT1 formulation of Section 2.2.1.

4.1 Relaxing the Quadratic Model

A classical relaxation of the MKP is obtained by relaxing in a Lagrangian fashion the cardinality constraints (10) with multipliers $\lambda_i \geq 0$ $(i \in N)$. For the QMKP, such relaxation becomes:

$$\begin{aligned} L^Q(\lambda) = & \sum_{i=1}^{n} \lambda_i + \max \sum_{i=1}^{n} \sum_{k=1}^{m} (p_i - \lambda_i) x_{ik} + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{m} p_{ij} x_{ik} x_{jk} \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i x_{ik} \leq C_k \qquad (k \in M) \\ & x \in \{0,1\}^{n \times m}. \end{aligned} \tag{45}$$

As the objective function does not contain terms involving items assigned to different knapsacks, the problem decomposes into m independent QKPs (one for each knapsack $k \in M$).

It is worth mentioning that, if the knapsack set M is partitioned into t subsets $M_1, \ldots, M_t$, such that all knapsacks in M_h $(h = 1, \ldots, t)$ have the same capacity $\overline{C}_h$, the optimal solution to the above Lagrangian relaxation can be obtained by solving t independent QKPs. Indeed, for each subset M_h, it is enough to solve one single QKP and to sum up the optimal values. Such situation occurs, e.g., in the benchmark instances by Bergman [2], where all knapsacks have the same capacity.

In order to solve the *Lagrangian dual* problem, i.e., to find the best possible set of multipliers, λ^*, in our computational experiments we adopted the *proximal bundle method*, as implemented by Frangioni [9]. The corresponding software is freely available at `https://gitlab.com/frangio68/ndosolver_fioracle_project` (as a part of the `NDOSolver/FiOracle` suite of `C++` solvers for NonDifferentiable Optimization, developed by the Department of Computer Science of the University of Pisa).

4.2 Relaxing DRLT1

A different Lagrangian relaxation can be obtained by the DRLT1 model introduced in Section 2.2.1. Let us dualize the symmetry equations (26) with multipliers $\lambda_{ijk} \lessgtr 0$. We get:

$$L^R(\lambda) = \max \quad \sum_{i=1}^{n}\sum_{k=1}^{m} p_i x_{ik} + \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{k=1}^{m}(\tfrac{1}{2}\, p_{ij} + \lambda_{ijk}) y_{ijk} \tag{46}$$

$$\text{s.t.} \quad y_{ijk} \leq x_{ik} \qquad (i \in N, j \in N \setminus \{i\}, k \in M) \tag{47}$$

$$\sum_{j \in N\setminus\{i\}} w_j y_{ijk} \leq (C_k - w_i)\; x_{ik} \;\; (i \in N, k \in M) \tag{48}$$

$$y_{ijk} \in \{0,1\} \qquad (i \in N, j \in N \setminus \{i\}, k \in M) \tag{49}$$

$$(9), (10), (11).$$

Since the multipliers λ_{ijk} for the symmetry constraints (26) are only defined for $j > i$, we assume, for notational convenience, that $\lambda_{jik} = -\lambda_{ijk}$ in (46).

The main reason for relaxing (26) is that the resulting model has a *decomposable* structure. Observe indeed that constraints (47) allow a variable y_{ijk} to be 1 only if x_{ik} is 1. Moreover, for each pair i, k $(i \in N, k \in M)$, variables y_{ijk} $(j \in N \setminus \{i\})$ only appear in capacity constraints (48) and in the objective function. Hence, if all x_{ik} variables are fixed, the relaxed problem consists of nm independent sub-problems, one for each pair i, k. More precisely, the relaxed problem decomposes into $nm + 1$ sub-problems, that can be cascaded as follows:

(i) first we solve nm (linear) KPs, one for each pair i, k $(i \in N, k \in M)$, of the form:

$$\begin{aligned} \max \quad & \sum_{\substack{j=1\\ j\neq i}}^{n} (\tfrac{1}{2}\, p_{ij} + \lambda_{ijk}) y_{ijk} \\ \text{s.t.} \quad & \sum_{j \in N\setminus\{i\}} w_j y_{ijk} \leq (C_k - w_i) x_{ik} \\ & y_{ijk} \in \{0,1\} \qquad\qquad (j \in N \setminus \{i\}) \\ & x_{ik} \in \{0,1\}. \end{aligned} \tag{50}$$

having only one x_{ik} variable and its associated $n-1$ auxiliary variables y_{ijk} $(j \in N \setminus \{i\})$ subject to a single capacity constraint (50) associated with the pair (i, k). We denote by v_{ik} the optimal solution value when $x_{ik} = 1$, while the optimal solution value is clearly 0 when $x_{ik} = 0$.

(ii) then we solve a unique (linear) *pseudo*-MKP with all the original x_{ik} variables subject to constraints (9)-(11):

$$\begin{aligned} \max \quad & \sum_{i=1}^{n}\sum_{k=1}^{m} \tilde{p}_{ik} x_{ik} \\ \text{s.t.} \quad & (9), (10), (11), \end{aligned} \tag{51}$$

where $\tilde{p}_{ik} = p_i + v_{ik}$ $(i \in N, k \in M)$.

Observe that, as it is known that the polyhedron of the 0-1 knapsack problem KP is not integral (see, e.g., Nemhauser and Wolsey [23]), our Lagrangian problem does not have the integrality property. Therefore, the Lagrangian bound, corresponding to the optimal dual multipliers λ^*, is not dominated by the standard continuous relaxation of DRLT1.

In this case too we performed our computational experiments by solving the Lagrangian dual problem by means of the proximal bundle method, as implemented by Frangioni [9].

5 Computational experiments

The formulations and the relaxations introduced in the previous sections were implemented in `C++` language. In the present section, we report the outcome of computational experiments aimed at evaluating the quality of the upper bounds produced by the polynomial-size models and the relaxations we have introduced. All the experiments were performed on a single thread of an AMD Ryzen 7 2700X Eight-Core Processor running at 3.7 GHz with 64 GB RAM. In order to evaluate our models and relaxations, we used benchmark instances adopted by Bergman [2], for most of which his Branch-and-Price algorithm could find the optimal solution (available online, see below). For the sake of completeness, in the next section we describe the way in which the instances were generated. The solution of our mathematical models was obtained using different codes:

- the general purpose solver CPLEX 12.10;
- the open source `C` code `quadknap`, that implements the algorithm for the QKP developed by Caprara et al. [5] and is available at the home page of D. Pisinger, `http://hjemmesider.diku.dk/~pisinger/codes.html` . This code works with integer parameters and non-negative pairwise profits p_{ij}: in Section 5.2 we detail how we handled this feature to solve Lagrangian subproblems;
- the open source Fortran code `MT1R`, that implements a variant of the KP algorithm MT1 by Martello and Toth [22] (adapted to non-integer parameters), available at the home page of S. Martello, `http://www.or.deis.unibo.it/knapsack.html` .

5.1 Benchmark instances

Bergman [2] presented two sets of random instances, called `HJ` and `SS`, based on the generation schemes proposed, respectively, by Hiley and Julstrom [17] for the QMKP and by Sarac and Sipahioglu [29] for a generalization of the problem. However, as reported by Chen and Hao [6], the known optimality gap for even the easiest of the `HJ` instances ($n = 100$) is enormous, so smaller instances were generated by Bergman [2] (with $n \in \{20, 25, 30, 35\}$) to test his exact approach. For our experiments, we considered the `HJ` instances, both because they have been specifically designed for the QMKP and because all the involved pairwise profits are non-negative.

All the instances can be downloaded from the INFORMS page as a zipfile at the address `https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.2018.0840/suppl_`

file/ijoc.2018.0840-instances.sm2.zip. The knapsacks have a common integer capacity C, with n ranging in $\{20, 25, 30, 35\}$ and m in $\{3, 5, 10\}$. Three different values $d \in (0, 1]$ were used for the density of the non-zero quadratic terms: $d \in \{0.25, 0.50, 0.75\}$. For each triple (n, m, d), 5 random instances were produced as follows. Linear profits p_i were generated as uniformly random integers from $[0, 100]$. For every pair $i, j \in N$, quadratic profits p_{ij} were set with probability d to a random integer value uniformly drawn from $[0, 100]$, and to 0 with probability $1 - d$.

The weights w_i were generated as uniformly random integers from $[1, 50]$, while the capacities C were all set to $\lfloor 0.8 \sum_{i \in N} w_i/m \rfloor$. In total, 180 instances were thus generated.

In addition, in order to analyze how the most promising reformulations and relaxations scale for larger values of n, we generated new HJ instances, using the instance generator provided by Bergman [2], that is available for download. The generator produces instances according to the scheme described above. In this case, we considered instances with n ranging in $\{40, 45, 50, 55, 60\}$, m in $\{3, 5, 10\}$, and $d \in \{0.25, 0.50, 0.75\}$.

5.2 Experiments

We first evaluate the polynomial-size formulations discussed in Sections 1 and 2, for what concerns both their performance on the computation of the optimal solution and the quality of the LP relaxation of the linear ones. Table 1 reports on the different formulations of the QMKP, when solved through CPLEX, with one hour time limit. The six groups, of three columns each, refer to the models we have obtained for the QMKP:

- CPLEX-QF: 0-1 quadratic formulation (Section 1);
- CPLEX-FGW: 0-1 linear formulation by Fortet, Glover, and Woolsey (Section 2);
- CPLEX-GLOV: mixed-integer linear formulation by Glover (Section 2);
- CPLEX-RLT1: Level 1 reformulation linearization by Sherali and Adams (Section 2.2);
- CPLEX-DRLT1: decomposable Level 1 reformulation (Section 2.2.1).
- CPLEX-MDRLT1: modified decomposable Level 1 reformulation (Section 2.2.1).

Each line refers to a triple (n, m, d). For each formulation, the three entries in the table report (over the corresponding 5 instances),

- %gap = average percentage optimality gap of the best solution value z obtained by CPLEX within one CPU hour with respect to its best found upper bound u, computed as $100\,(u - z)/z$. In parentheses #, total number of instances solved to proven optimality;
- nodes = average number of nodes of the CPLEX branch-decision tree;
- t(s) = average CPU time expressed in seconds.

instance			CPLEX-QF			CPLEX-FGW			CPLEX-GLOV			CPLEX-RLT1			CPLEX-DRLT1			CPLEX-MDRLT1		
n	m	d	%gap (#)	nodes	t(s)	%gap (#)	nodes	t(s)	%gap (#)	nodes	t(s)	%gap (#)	nodes	t(s)	%gap (#)	nodes	t(s)	%gap (#)	nodes	t(s)
20	3	0.25	0.00 (5)	712	0.2	0.00 (5)	692	1.7	0.00 (5)	2091	0.8	0.00 (5)	209	4.8	0.00 (5)	411	0.2	0.00 (5)	448	0.2
20	5	0.25	0.00 (5)	1282	1.1	0.00 (5)	1698	9.5	0.00 (5)	4188	3.3	0.00 (5)	73	4.1	0.00 (5)	380	0.7	0.00 (5)	1877	2.5
20	10	0.25	0.00 (5)	58	0.2	0.00 (5)	43	1.2	0.00 (5)	127	0.2	0.00 (5)	3	0.4	0.00 (5)	12	0.0	0.00 (5)	682	0.3
20	3	0.50	0.00 (5)	6762	4.7	0.00 (5)	12958	42.5	0.00 (5)	9893	5.9	0.00 (5)	413	6.0	0.00 (5)	1062	1.6	0.00 (5)	1475	1.8
20	5	0.50	0.00 (5)	9263	13.5	0.00 (5)	43682	253.8	0.00 (5)	33478	25.6	0.00 (5)	123	4.9	0.00 (5)	453	1.4	0.00 (5)	1936	4.4
20	10	0.50	0.00 (5)	112	0.5	0.00 (5)	80	2.2	0.00 (5)	440	0.4	0.00 (5)	5	1.6	0.00 (5)	15	0.3	0.00 (5)	217	0.3
20	3	0.75	0.00 (5)	23199	25.4	0.00 (5)	32420	88.2	0.00 (5)	55320	28.2	0.00 (5)	895	10.0	0.00 (5)	2333	6.1	0.00 (5)	4230	5.3
20	5	0.75	0.00 (5)	76629	157.5	0.00 (5)	30042	127.2	0.00 (5)	242127	233.9	0.00 (5)	138	4.7	0.00 (5)	661	3.2	0.00 (5)	1175	3.8
20	10	0.75	0.00 (5)	229	1.1	0.00 (5)	199	3.3	0.00 (5)	1995	2.5	0.00 (5)	9	1.7	0.00 (5)	10	0.5	0.00 (5)	47	0.2
Avg(#tot)			0.00 (45)	13138	22.7	0.00 (45)	13535	58.8	0.00 (45)	38851	33.4	0.00 (45)	208	4.2	0.00 (45)	593	1.6	0.00 (45)	1343	2.1
25	3	0.25	0.00 (5)	2898	1.5	0.00 (5)	3183	18.2	0.00 (5)	5416	2.4	0.00 (5)	1328	32.7	0.00 (5)	2805	1.9	0.00 (5)	2346	1.5
25	5	0.25	0.00 (5)	5243	6.4	1.12 (4)	69566	1028.8	0.00 (5)	13960	14.1	0.00 (5)	790	44.4	0.00 (5)	2727	6.7	0.00 (5)	7103	12.0
25	10	0.25	0.00 (5)	900	2.0	0.00 (5)	812	14.7	0.00 (5)	2938	4.9	0.00 (5)	78	8.0	0.00 (5)	190	0.8	0.00 (5)	558918	641.9
25	3	0.50	0.00 (5)	39882	37.8	0.00 (5)	62716	320.6	0.00 (5)	161340	168.8	0.00 (5)	3480	84.9	0.00 (5)	10403	26.6	0.00 (5)	8912	19.2
25	5	0.50	0.00 (5)	286271	813.6	24.71 (1)	210054	3540.8	0.00 (5)	1966676	2676.9	0.00 (5)	1486	47.7	0.00 (5)	4907	39.3	0.00 (5)	14344	65.1
25	10	0.50	0.00 (5)	1776	6.9	0.00 (5)	41219	509.4	0.00 (5)	17082	39.6	0.00 (5)	35	8.9	0.00 (5)	83	1.7	0.00 (5)	4956	17.9
25	3	0.75	0.00 (5)	553050	1157.6	4.07 (3)	343601	2256.5	5.39 (2)	1671387	2430.5	0.00 (5)	5507	154.8	0.00 (5)	24292	163.4	0.00 (5)	18290	60.1
25	5	0.75	14.16 (2)	735365	3117.2	47.11 (0)	253386	3600.0	31.83 (0)	1417607	3600.0	0.00 (5)	4148	167.9	0.00 (5)	24286	295.2	0.00 (5)	32127	201.2
25	10	0.75	0.00 (5)	10690	56.1	10.05 (1)	189176	2907.7	1.78 (4)	323526	965.3	0.00 (5)	63	9.9	0.00 (5)	279	4.1	0.00 (5)	11074	61.4
Avg(#tot)			1.57 (42)	181786	577.7	9.67 (29)	130412	1577.4	4.33 (36)	619992	1100.3	0.00 (45)	1879	62.1	0.00 (45)	7775	60.0	0.00 (45)	73119	120.0
30	3	0.25	0.00 (5)	6935	5.6	0.00 (5)	12680	116.0	0.00 (5)	45366	39.0	0.00 (5)	2546	165.4	0.00 (5)	6301	6.5	0.00 (5)	5057	5.4
30	5	0.25	0.00 (5)	78649	162.3	16.36 (1)	142000	2988.1	0.00 (5)	548528	991.7	0.59 (4)	11848	1046.0	0.00 (5)	17744	69.6	0.00 (5)	33281	90.7
30	10	0.25	1.14 (4)	162474	745.5	9.03 (3)	39804	1637.6	0.00 (5)	67015	223.2	0.00 (5)	242	29.9	0.00 (5)	675	4.9	0.63 (4)	225539	1226.4
30	3	0.50	0.00 (5)	599912	1409.6	7.20 (2)	341303	3134.3	1.71 (4)	1706468	2267.9	0.00 (5)	12057	577.8	0.00 (5)	126844	464.4	0.00 (5)	73760	254.5
30	5	0.50	21.42 (0)	785381	3600.0	69.75 (0)	158100	3600.0	26.73 (0)	1294407	3600.0	3.63 (1)	32286	2949.1	0.00 (5)	70244	845.2	0.00 (5)	95814	776.9
30	10	0.50	20.04 (1)	360085	3013.4	50.21 (0)	80619	3600.0	15.28 (0)	711346	3600.0	0.00 (5)	612	64.2	0.00 (5)	1620	35.4	0.00 (5)	11446	185.5
30	3	0.75	20.45 (0)	1043048	3600.0	31.10 (0)	425252	3600.0	22.80 (0)	1845088	3600.0	0.00 (5)	14281	789.1	1.17 (3)	127112	2061.4	0.00 (5)	135415	816.1
30	5	0.75	65.62 (0)	419412	3600.0	84.78 (0)	207738	3600.0	51.48 (0)	922215	3600.0	0.00 (5)	10027	884.7	0.00 (5)	46904	1038.0	0.00 (5)	56323	657.6
30	10	0.75	52.40 (1)	204383	3050.9	77.71 (0)	73858	3600.0	61.36 (0)	622603	3600.0	0.00 (5)	230	39.4	0.00 (5)	7222	196.4	0.00 (5)	4525	148.5
Avg(#tot)			20.12 (21)	406698	2131.9	38.46 (11)	164595	2875.1	19.93 (19)	862560	2391.3	0.47 (40)	9348	727.3	0.13 (43)	44963	524.7	0.07 (44)	71240	462.4
35	3	0.25	0.00 (5)	49383	48.2	0.17 (4)	82912	1322.0	0.00 (5)	195385	282.0	0.00 (5)	8045	815.9	0.00 (5)	41148	51.4	0.00 (5)	30827	41.4
35	5	0.25	1.50 (4)	638998	2124.0	21.63 (0)	104089	3600.0	7.80 (0)	1181369	3600.0	0.00 (5)	12511	2029.2	0.39 (4)	206468	1344.8	0.00 (5)	241679	1109.0
35	10	0.25	4.64 (3)	223228	1771.7	42.40 (0)	39541	3600.0	8.83 (2)	456966	2882.6	0.00 (5)	970	201.3	0.00 (5)	6786	96.8	1.63 (3)	116274	2015.3
35	3	0.50	10.02 (1)	1641493	3255.5	26.09 (0)	224323	3600.0	15.65 (0)	1239528	3600.0	2.45 (2)	29491	2816.6	5.19 (1)	542010	3358.3	2.86 (2)	489435	2924.5
35	5	0.50	49.95 (0)	523292	3600.0	78.72 (0)	111527	3600.0	46.77 (0)	703265	3600.0	10.61 (0)	21284	3600.0	8.96 (0)	168929	3600.0	7.90 (0)	229339	3600.0
35	10	0.50	32.74 (0)	211563	3600.0	105.18 (0)	35657	3600.0	65.90 (0)	379695	3600.0	0.00 (5)	2946	713.2	1.59 (2)	46683	2283.3	0.67 (4)	47960	2161.0
35	3	0.75	30.51 (0)	927451	3600.0	40.86 (0)	237571	3600.0	25.66 (0)	1128057	3600.0	0.00 (4)	21997	1980.4	5.06 (1)	187245	3081.6	1.97 (2)	197177	2437.0
35	5	0.75	86.50 (0)	277265	3600.0	108.75 (0)	111032	3600.0	60.77 (0)	625095	3600.0	4.90 (0)	15068	3600.0	6.24 (0)	98240	3600.0	5.19 (0)	139215	3600.0
35	10	0.75	135.39 (0)	112848	3600.0	155.78 (0)	34545	3600.0	119.48 (0)	319078	3600.0	1.43 (3)	7874	2011.1	5.33 (1)	29577	2965.4	1.90 (2)	43900	3126.7
Avg(#tot)			39.03 (13)	459759	2799.9	64.40 (4)	92953	3346.9	38.99 (7)	637548	3151.6	2.15 (29)	12274	1974.2	3.64 (19)	142480	2264.6	2.46 (23)	167329	2335.0
Ov.Avg(#tot)			15.18(121)	278337	1383.1	28.13 (89)	104391	1964.6	15.81(107)	553363	1669.2	0.66(159)	6197	692.0	0.94(152)	50196	712.7	0.63(157)	79087	729.9

Table 1: CPLEX solution of the polynomial formulations within one hour. Average percentage optimality gap, number of instances solved to proven optimality, average number of nodes and CPU time over five instances. Time limit: 1 hour.

The average values of %gap, nodes, t(s) and the total value of # for each value of n (45 instances) are also reported, as well as the overall values over the 180 instances.

The table shows that the direct use of the models to provide solutions to the QMKP through a general purpose solver like CPLEX (we also tried Gurobi 9, with similar results) can only be effective for small size instances. We observe that the quadratic formulation QF and the two linear formulations FGW and GLOV obtain worse results than those obtained by the three Level 1 RLT formulations.

For $n \leq 25$, RLT1, DRLT1, and MDRLT1 could solve all 90 instances to optimality. DRLT1 turned out to be the fastest method, although it requires a higher number of CPLEX decision nodes than RLT1. For $n = 30$, the same models could solve, respectively, 40, 43, and 44 instances out of 45, with CPU times of few hundred seconds. The models look instead inadequate for instances with $n = 35$.

As previously mentioned, all the considered instances but two were efficiently solved to optimality by Bergman's [2] Branch-and-Price algorithm, referred to as BBP in the following. (He used Gurobi 7.5.1 with one hour time limit on a computer similar to ours, namely an Intel Core i7-4770 running at 3.40 GHz with 32 GB RAM.) Although a direct comparison between the CPLEX solution of polynomial-size models and a specialized Branch-and-Price algorithm may be questionable, we can observe that the Level 1 RLT models appear to perform better for $n = 20$ and $n = 25$, while BBP is more effective for $n = 30$, and much better for $n = 35$. More specifically,

- for $n = 20$, the three Level 1 reformulations and BBP solved all instances, with DRLT1 and MDRLT1 taking smaller times (on average, 1.6 and 2.1 seconds, respectively, versus 4.2 seconds of RLT1 and 3.6 seconds of BBP);
- for $n = 25$, RLT1 and DRLT1 solved all 45 instances (with average times 62.1 and 60.0 seconds, respectively) while BBP solved one instance less with average time 95.8 seconds. MDRLT1 solved all instances, but required a much higher, anomalous, time;
- for $n = 30$, MDRLT1 solved 44 instances with an average time of 462.4 seconds, while BBP solved all 45 instances with an average time of 151.2 seconds;
- for $n = 35$, RLT1 solved 29 instances with an average time of 1974.2 seconds, while BBP solved 44 instances with an average time of 455.2 seconds.

Table 2 examines the quality of the upper bounds computed through the LP relaxations of the linear models considered in Table 1. The five groups, of two columns each, refer to:

- LP-FGW: LP relaxation of FGW (Section 2.1);
- LP-GLOV: LP relaxation of GLOV (Section 2.1);
- LP-RLT1: LP relaxation of RLT1 (Section 2.2);
- LP-DRLT1: LP relaxation of DRLT1 (Section 2.2.1);
- LP-MDRLT1: LP relaxation of MDRLT1 (Section 2.2.1);

instance			LP-FGW		LP-GLOV		LP-RLT1		LP-DRLT1		LP-MDRLT1	
n	m	d	%gap	t(s)	%gap	t(s)	%gap	t(s)	%gap	t(s)	%gap	t(s)
20	3	0.25	37.75	0.01	39.21	0.00	29.83	0.01	29.83	0.00	29.83	0.00
20	5	0.25	64.50	0.01	66.31	0.00	29.47	0.01	29.48	0.00	29.30	0.00
20	10	0.25	163.12	0.01	166.21	0.00	19.31	0.02	19.32	0.01	11.94	0.00
20	3	0.50	68.66	0.01	70.94	0.00	31.59	0.01	31.63	0.00	31.63	0.00
20	5	0.50	126.35	0.01	129.41	0.00	29.31	0.01	29.32	0.01	28.05	0.00
20	10	0.50	265.74	0.01	270.76	0.01	24.72	0.02	24.72	0.01	16.72	0.00
20	3	0.75	90.80	0.01	95.75	0.00	28.76	0.01	28.77	0.01	28.77	0.00
20	5	0.75	156.59	0.01	163.41	0.00	24.26	0.02	24.26	0.01	23.67	0.01
20	10	0.75	310.87	0.01	322.10	0.01	19.36	0.02	19.37	0.01	12.06	0.01
Avg			142.71	0.01	147.12	0.00	26.29	0.01	26.30	0.01	23.55	0.00
25	3	0.25	41.19	0.01	42.97	0.00	36.28	0.02	36.28	0.00	36.28	0.00
25	5	0.25	65.55	0.01	67.69	0.00	33.27	0.02	33.27	0.01	33.27	0.00
25	10	0.25	138.13	0.02	141.35	0.01	24.82	0.04	24.85	0.01	19.86	0.01
25	3	0.50	78.55	0.01	81.90	0.00	40.42	0.02	40.49	0.01	40.49	0.01
25	5	0.50	132.64	0.01	137.10	0.01	33.33	0.03	33.34	0.01	33.34	0.01
25	10	0.50	263.46	0.02	270.41	0.01	27.69	0.04	27.74	0.02	20.17	0.01
25	3	0.75	93.43	0.01	97.98	0.00	34.57	0.02	34.59	0.01	34.59	0.01
25	5	0.75	163.96	0.01	170.19	0.01	30.60	0.03	30.60	0.01	30.60	0.01
25	10	0.75	310.82	0.02	320.75	0.01	23.95	0.05	23.98	0.02	20.03	0.01
Avg			143.08	0.01	147.81	0.00	31.66	0.03	31.68	0.01	29.85	0.01
30	3	0.25	43.53	0.01	45.56	0.00	39.37	0.02	39.37	0.01	39.37	0.00
30	5	0.25	69.84	0.02	72.27	0.00	39.87	0.03	39.91	0.01	39.91	0.01
30	10	0.25	135.77	0.03	139.28	0.01	28.45	0.07	28.53	0.02	24.19	0.01
30	3	0.50	77.83	0.01	82.43	0.00	47.04	0.03	47.10	0.01	47.10	0.01
30	5	0.50	125.58	0.02	131.49	0.01	37.88	0.04	37.88	0.01	37.88	0.01
30	10	0.50	250.12	0.03	259.62	0.01	29.97	0.07	30.01	0.02	26.12	0.02
30	3	0.75	104.83	0.02	110.95	0.00	36.65	0.04	36.65	0.01	36.65	0.01
30	5	0.75	175.09	0.02	183.45	0.01	28.54	0.05	28.54	0.02	28.54	0.01
30	10	0.75	348.85	0.04	362.68	0.01	26.20	0.08	26.20	0.03	20.71	0.02
Avg			147.94	0.02	154.19	0.01	34.89	0.05	34.91	0.02	33.39	0.01
35	3	0.25	50.01	0.02	53.32	0.00	43.19	0.04	43.19	0.01	43.19	0.01
35	5	0.25	77.38	0.02	81.28	0.01	44.89	0.05	44.90	0.02	44.90	0.01
35	10	0.25	137.54	0.04	142.89	0.01	31.79	0.09	31.79	0.03	29.81	0.01
35	3	0.50	82.83	0.02	87.35	0.00	49.75	0.04	49.79	0.02	49.79	0.01
35	5	0.50	135.69	0.03	141.55	0.01	42.82	0.05	42.85	0.02	42.85	0.01
35	10	0.50	248.47	0.05	257.14	0.01	30.51	0.09	30.51	0.03	28.35	0.02
35	3	0.75	101.86	0.02	108.36	0.01	36.15	0.06	36.16	0.02	36.16	0.02
35	5	0.75	178.30	0.03	187.29	0.01	32.10	0.07	32.10	0.02	32.10	0.02
35	10	0.75	345.25	0.05	360.17	0.01	24.95	0.11	24.95	0.04	23.37	0.04
Avg			150.81	0.03	157.70	0.01	37.35	0.07	37.36	0.02	36.72	0.02
Ov.Avg			146.14	0.02	151.71	0.01	32.55	0.04	32.56	0.01	30.88	0.01

Table 2: Upper bounds computed through LP relaxation of the linear formulations. Average percentage optimality gap and CPU time over 5 instances. Time limit: 1 hour.

The LP relaxations were solved through CPLEX. Each line refers to a triple (n, m, d). For each formulation, the two entries in the table report the values (over the corresponding 5 instances) of:

- `%gap` = average percentage optimality gap of the upper bound u obtained within one CPU hour with respect to the best known solution value z, computed as $100\,(u - z)/z$. The value of z is optimal for 179 instances out of 180: 178 were provided by Bergman [2], one more was found by the Level 1 RLT models (see the comments on Table 1);
- `t(s)` = average CPU time expressed in seconds.

We have seen in Table 1 that the linear models (DRLT1 in particular) can provide good solutions for instances of limited size. Table 2 shows that the CPU times for computing

their LP relaxations are very small, but the quality of the upper bounds they provide is poor, especially for what concerns GLOV and FGW. The performances of RLT1 and DRLT1 are very similar to each other. Although the continuous relaxation of DRLT1 is weaker than that of RLT1 (as observed in Section 2.2.1), the quality of the bounds they produce is practically the same, while DRLT1 is faster. The best performance was obtained by MDRLT1. In particular: (i) for $m = 10$, MDRLT1 produced the smallest percentage gaps, thanks to the addition of constraints (39); (ii) MDRLT1 was slightly faster than DRLT1, probably due to the use of sets R_{ik} and S_{ik}.

In any case, the results of Table 2 indicate that the LP relaxations are inadequate to be embedded in an enumerative approach. We next show that much better results can be obtained from Lagrangian relaxations.

In Table 3 we analyze the quality of the upper bounds obtained by the surrogate and

instance			Srg CPLEX		Srg Qknap		Lgr QP CPLEX		S-Lgr QP CPLEX		S-Lgr QP Qknap		S-Lgr QPL CPLEX		D-Lgr DRLT1 CPLEX		D-Lgr DRLT1 MT1R	
n	m	d	%gap	t(s)	%gap	t(s)	%gap	t(s)	%gap	t(s)	%gap	t(s)	%gap	t(s)	%gap	t(s)	%gap	t(s)
20	3	25	34.45	0.01	34.45	0.00	0.24	1.37	0.24	0.71	0.24	0.38	0.24	0.64	26.04	5.29	26.04	1.58
20	5	25	59.80	0.01	59.80	0.00	0.09	2.11	0.09	1.12	0.09	0.45	0.09	1.05	18.90	13.09	18.90	2.46
20	10	25	156.18	0.01	156.18	0.00	0.00	1.78	0.00	0.81	0.00	0.43	0.00	0.74	3.89	4.35	3.89	1.50
20	3	50	62.53	0.03	62.53	0.00	0.44	5.80	0.44	1.73	0.44	0.45	0.44	1.81	25.24	33.06	25.24	6.67
20	5	50	119.77	0.02	119.77	0.00	0.02	13.90	0.02	1.73	0.02	0.46	0.02	1.77	18.18	34.55	18.18	7.27
20	10	50	253.45	0.02	253.45	0.00	0.47	2.43	0.47	1.24	0.47	0.40	0.47	0.90	8.74	16.02	8.74	2.61
20	3	75	85.14	0.04	85.14	0.00	1.07	7.20	1.07	3.14	1.07	0.49	1.07	2.86	22.51	48.52	22.51	9.16
20	5	75	149.76	0.02	149.76	0.00	0.52	10.44	0.52	2.09	0.52	0.45	0.52	1.87	14.42	52.50	14.42	9.87
20	10	75	297.79	0.04	297.79	0.00	0.00	3.68	0.00	1.39	0.00	0.44	0.00	0.91	5.31	21.50	5.31	3.94
Avg			135.43	0.02	135.43	0.00	0.32	5.41	0.32	1.55	0.32	0.44	0.32	1.39	15.91	25.43	15.91	5.01
25	3	25	38.84	0.01	38.84	0.00	0.41	5.16	0.41	1.89	0.41	0.63	0.41	1.86	34.23	7.80	34.23	3.20
25	5	25	63.01	0.01	63.01	0.00	0.24	7.91	0.24	1.62	0.24	0.73	0.24	1.68	23.59	35.99	23.59	7.24
25	10	25	134.29	0.01	134.29	0.00	0.19	4.49	0.19	1.42	0.19	0.66	0.19	1.15	10.66	20.27	10.66	4.00
25	3	50	74.39	0.03	74.39	0.00	0.45	12.67	0.45	6.58	0.45	0.81	0.45	7.18	35.72	52.57	35.71	14.37
25	5	50	127.88	0.02	127.88	0.00	0.31	12.38	0.31	3.51	0.31	0.71	0.31	3.60	25.06	85.20	25.08	20.13
25	10	50	256.45	0.02	256.45	0.00	0.00	11.24	0.00	1.75	0.00	0.73	0.00	1.23	10.98	60.98	10.99	17.12
25	3	75	87.19	0.07	87.19	0.00	0.46	54.57	0.46	42.07	0.46	0.88	0.46	21.49	30.31	80.56	30.31	22.51
25	5	75	155.37	0.06	155.37	0.00	0.86	22.97	0.86	7.86	0.86	0.70	0.86	6.98	22.93	122.87	22.93	31.31
25	10	75	298.84	0.05	298.84	0.00	0.28	42.17	0.28	1.93	0.28	0.78	0.28	1.61	11.07	91.47	11.12	31.12
Avg			137.36	0.03	137.36	0.00	0.36	19.29	0.36	7.63	0.36	0.74	0.36	5.20	22.73	61.97	22.74	16.78
30	3	25	41.91	0.03	41.91	0.00	0.19	12.16	0.19	4.13	0.19	1.23	0.19	4.50	37.40	16.90	37.40	7.46
30	5	25	68.10	0.04	68.10	0.00	0.20	20.86	0.20	3.31	0.20	1.01	0.20	3.49	33.05	55.19	33.04	15.61
30	10	25	133.05	0.04	133.05	0.00	0.16	20.25	0.16	1.75	0.16	1.01	0.16	1.53	14.09	65.79	14.10	17.79
30	3	50	73.90	0.04	73.90	0.01	0.73	64.11	0.73	50.42	0.74	1.75	0.73	39.86	43.55	83.00	43.56	30.48
30	5	50	120.64	0.05	120.64	0.00	0.04	27.11	0.04	11.32	0.04	1.17	0.04	10.93	31.42	163.19	31.38	60.97
30	10	50	242.59	0.05	242.59	0.00	0.05	47.13	0.05	3.28	0.06	1.05	0.05	2.70	17.37	179.19	17.30	87.92
30	3	75	98.86	0.12	98.86	0.01	1.07	496.64	1.07	456.88	1.11	1.96	1.07	138.98	33.45	143.83	33.44	53.25
30	5	75	167.51	0.08	167.51	0.00	0.25	74.82	0.25	58.34	0.26	1.05	0.25	17.79	23.53	223.65	23.53	79.20
30	10	75	337.06	0.09	337.06	0.01	0.23	32.14	0.23	4.56	0.23	0.84	0.23	3.09	13.53	221.99	13.56	104.73
Avg			142.63	0.06	142.63	0.00	0.32	88.36	0.32	66.00	0.33	1.23	0.32	24.76	27.49	128.08	27.48	50.82
35	3	25	47.94	0.09	47.94	0.01	0.51	20.84	0.51	10.30	0.51	3.30	0.51	11.41	41.86	32.58	41.86	17.53
35	5	25	75.42	0.06	75.42	0.01	0.23	23.08	0.23	7.07	0.24	1.40	0.23	8.05	39.50	87.92	39.50	31.58
35	10	25	135.16	0.03	135.16	0.01	0.16	54.57	0.16	2.86	0.16	1.11	0.16	2.93	18.14	144.62	18.13	57.52
35	3	50	81.15	0.04	81.15	0.01	0.49	377.56	0.49	342.35	0.49	4.61	0.49	261.28	46.65	143.11	46.66	67.96
35	5	50	133.57	0.04	133.57	0.01	0.36	149.04	0.36	119.59	0.36	1.59	0.36	63.66	37.17	266.52	37.16	125.57
35	10	50	245.30	0.04	245.30	0.01	0.15	47.62	0.15	6.74	0.15	1.02	0.15	5.73	21.11	331.66	21.10	167.77
35	3	75	98.13	0.16	98.13	0.01	0.54	2192.90	0.50	2023.31	0.50	3.97	0.50	381.58	33.66	268.22	33.66	137.15
35	5	75	173.40	0.14	173.40	0.01	0.30	792.43	0.30	689.64	0.31	1.93	0.30	134.80	28.10	401.31	28.07	182.57
35	10	75	335.82	0.13	335.82	0.01	0.36	103.01	0.36	21.78	0.36	1.25	0.36	13.03	17.41	495.15	17.24	293.53
Avg			147.32	0.08	147.32	0.01	0.35	417.89	0.34	358.18	0.34	2.24	0.34	98.05	31.51	241.23	31.49	120.13
Ov.Avg			140.68	0.05	140.68	0.00	0.34	132.74	0.33	108.34	0.34	1.16	0.33	32.35	24.41	114.18	24.40	48.18

Table 3: Upper bounds computed through surrogate and Lagrangian relaxations. Average percentage gap and CPU time over 5 instances. Time limit: 1 hour.

Lagrangian relaxations studied in Sections 3-4. For the surrogate relaxations, the optimal multipliers are known (see Proposition 4). For the Lagrangian relaxations, the search of the best multipliers was always performed via the proximal bundle method [9]. The columns provide information on the different ways we solved the relaxed subproblems (either CPLEX, or Quadknap [5], or MT1R [22]). We also consider both the case where separability due to equal capacities is exploited (see Section 4.1) and where it is not. The eight groups, of two columns each, refer to:

- `Srg CPLEX`: surrogate bound $S(\pi)$ (Section 3) solved through CPLEX;
- `Srg Qknap`: surrogate bound $S(\pi)$ (Section 3) solved through `quadknap` [5];
- `Lgr QP CPLEX`: Lagrangian bound $L^Q(\lambda)$ (Section 4.1) solved through CPLEX;
- `S-Lgr QP CPLEX`: Lagrangian bound $L^Q(\lambda)$ exploiting separability, with the single QKP solved through CPLEX;
- `S-Lgr QP Qknap`: Lagrangian bound $L^Q(\lambda)$ exploiting separability, with the single QKP solved through `quadknap` [5]. Note that `quadknap` works with non-negative pairwise profits p_{ij} (which holds in our formulation) and integer coefficients. Since our Lagrangian linear profits $p_i - \lambda_i$ ($i \in N$) can assume non-integer values, we multiplied all profits by 100, rounded each resulting value a to $\lceil a \rceil$, and correspondingly divided the solution value by 100 (thus obtaining a valid upper bound on the optimal QKP solution);
- `S-Lgr QPL CPLEX`: Lagrangian bound $L^Q(\lambda)$ exploiting separability, with the single QKP linearized through the MDRLT1 formulation with $m = 1$ and solved through CPLEX;
- `D-Lgr DRLT1 CPLEX`: Lagrangian bound $L^R(\lambda)$ exploiting the decomposable structure (see (i)-(ii) of Section 4.2) with single KPs solved through CPLEX;
- `D-Lgr DRLT1 MT1R`: Lagrangian bound $L^R(\lambda)$ exploiting the decomposable structure (see (i)-(ii) of Section 4.2) with single KPs solved through `MT1R` [22].

Preliminary computational experiments showed that the exact solution of the linear pseudo-MKP (point (ii) in Section 4.2), performed at each iteration of the bundle procedure, takes a large computing time, so we replaced it with its LP relaxation (solved through CPLEX). The entries in the table are the same as for Table 2. The results indicate that:

- despite using optimal multipliers, the surrogate relaxation is very weak: it takes very short CPU times, but the upper bounds are extremely loose;
- all Lagrangian relaxations provide much better bounds, although $L^R(\lambda)$ is considerably weaker than $L^Q(\lambda)$ (with `D-Lgr DRLT1 MT1R` requiring much smaller CPU times than `D-Lgr DRLT1 CPLEX`);
- all versions of $L^Q(\lambda)$ are by far the best approaches:

– they produce an average gap of 0.34%, with individual gaps rarely exceeding 1%;

– their gaps are identical, with the only exception of `S-Lgr QP Qknap` due to the non-optimal solution of the Lagrangian subproblems imposed by `quadknap`, which can result in a suboptimal Lagrangian dual (for 7 instances out of 180, its value is higher by one unit);

– `Lgr QP CPLEX`, which does not exploit separability, has the highest CPU times;

– the second highest times are those of `S-Lgr QP CPLEX`, which solves the single QKP Lagrangian subproblem through CPLEX on the standard quadratic formulation. By linearizing the QKP through the MDRLT1 formulation, `S-Lgr QPL CPLEX` reduces the computational effort by two thirds;

– the best approach is by far `S-Lgr QP Qknap`, which directly solves the QKP through `quadknap`. It provides very tight upper bounds in short CPU times, with a much smaller growth rate with respect to n than that of the other $L^Q(\lambda)$ methods;

– by comparing the obtained upper bounds with the optimal solution values, it turns out that they are frequently identical: it happens for 94 instances out of 180 (92 instances for `S-Lgr QP Qknap`).

Finally, Table 4 shows how the two best performing reformulations `RLT1` and `MDRLT1`, as well as three of our relaxations scale for larger values of n, up to 60. The five groups, of two columns each, refer to:

- `CPLEX-RLT1`: Level 1 reformulation linearization by Sherali and Adams (Section 2.2);
- `CPLEX-MDRLT1`: modified decomposable Level 1 reformulation (Section 2.2.1);
- `LP-MDRLT1`: LP relaxation of MDRLT1 (Section 2.2.1);
- `S-Lgr QP Qknap`: Lagrangian bound $L^Q(\lambda)$ exploiting separability, with the single QKP solved through `quadknap` [5];
- `D-Lgr DRLT1 MT1R`: Lagrangian bound $L^R(\lambda)$ exploiting the decomposable structure (see (i)-(ii) of Section 4.2) with single KPs solved through `MT1R` [22].

Since for most of these instances the optimal value is unknown, we ran CPLEX, both for `RLT1` and `MDRLT1`, with a time limit of three hours (instead of one our, as for all other runs): the percentage gaps of the relaxations were thus computed with respect to the best solution value z obtained by the two CPLEX executions. The `%gap` and `t(s)` values have the same meaning as in the previous tables. For `CPLEX-RLT1` and `CPLEX-MDRLT1`, two additional columns provide

- `%gapL` = average percentage optimality gap of the lower bound L obtained within three hours with respect to z, computed as $100\,(z - L)/L$;

- %gapU = average percentage optimality gap of the upper bound U obtained within three hours with respect to z, computed as $100\,(U - z)/z$.

The table shows that the computing time for the three relaxations was generally below one hour, apart from a few cases that took slightly longer for the Lagrangian relaxation S-Lgr QP Qknap. The Lagrangian relaxation of the quadratic model still provides fairly good bounds, if compared to the others. Yet, the computing time of the bundle procedure

instance			CPLEX-RLT1				CPLEX-MDRLT1				LP-MDRLT1		S-Lgr QP Qknap		D-Lgr DRLT1 MT1R	
n	m	d	%gap	%gapL	%gapU	t(s)	%gap	%gapL	%gapU	t(s)	%gap	t(s)	%gap	t(s)	%gap	t(s)
40	3	25	0.00	0.00	0.00	2476.2	0.00	0.00	0.00	191.6	45.26	0.01	0.11	11.47	44.23	34.24
40	5	25	4.72	0.27	4.43	9577.2	2.65	0.12	2.53	8368.1	48.91	0.01	0.55	1.81	44.14	71.77
40	10	25	5.49	0.29	5.18	9308.6	0.00	0.00	0.00	2010.2	35.23	0.02	0.43	0.59	25.22	156.48
40	3	50	2.18	0.01	2.17	9154.0	8.32	0.04	8.28	10559.8	54.21	0.02	1.05	25.30	51.79	129.77
40	5	50	11.12	2.24	8.67	10793.2	11.12	0.00	11.11	10800.0	46.99	0.02	1.03	3.39	42.73	247.10
40	10	50	9.55	0.22	9.31	10792.9	6.60	0.04	6.56	9335.4	32.12	0.03	0.83	0.71	24.86	428.60
40	3	75	1.71	0.05	1.65	9265.2	8.81	0.20	8.59	10795.0	40.76	0.03	0.45	26.12	38.88	233.18
40	5	75	11.13	0.66	10.41	10793.2	10.82	0.29	10.49	10794.2	37.36	0.03	1.76	3.79	33.84	310.38
40	10	75	8.23	1.31	6.84	10792.8	6.96	0.00	6.96	10794.8	25.19	0.05	0.70	0.67	19.53	434.95
Avg			6.01	0.56	5.41	9217.0	6.14	0.08	6.06	8183.2	40.67	0.02	0.77	8.21	36.14	227.39
45	3	25	3.91	0.62	3.25	8889.6	0.00	0.00	0.00	1548.8	50.35	0.01	0.23	83.92	49.53	61.97
45	5	25	11.78	1.89	9.69	10792.8	12.27	0.00	12.27	10800.0	55.36	0.01	0.68	10.12	51.30	138.89
45	10	25	14.03	2.39	11.35	10792.7	10.27	0.00	10.28	10800.0	38.67	0.02	0.45	0.91	29.74	342.53
45	3	50	7.26	0.81	6.39	10117.3	16.17	0.00	16.17	10800.0	56.59	0.03	1.25	151.94	54.57	236.79
45	5	50	17.64	2.77	14.45	10792.9	19.00	0.00	19.00	10800.0	50.84	0.03	1.59	12.01	47.34	413.74
45	10	50	15.30	2.62	12.37	10792.5	11.23	0.00	11.23	10800.0	35.22	0.04	1.31	1.27	29.29	701.21
45	3	75	4.91	0.02	4.89	9354.8	12.09	0.13	11.95	10800.0	42.57	0.04	0.98	125.74	41.00	395.29
45	5	75	15.42	2.63	12.43	10792.7	12.93	0.00	12.94	10800.0	38.83	0.04	1.84	12.48	36.21	533.15
45	10	75	13.81	2.02	11.55	10792.3	11.16	0.00	11.16	10800.0	30.06	0.06	1.72	1.48	24.64	854.63
Avg			11.56	1.75	9.60	10346.4	11.68	0.01	11.67	9772.1	44.28	0.03	1.12	44.43	40.40	408.69
50	3	25	10.05	1.98	7.91	10792.3	3.91	0.00	3.91	7977.8	53.29	0.01	0.57	702.10	52.61	172.38
50	5	25	18.44	2.56	15.52	10792.1	21.03	0.00	21.03	10800.0	62.60	0.02	2.00	44.53	59.12	411.04
50	10	25	19.87	2.02	17.51	10792.2	15.51	0.00	15.51	10800.0	46.75	0.03	1.48	2.61	37.67	977.58
50	3	50	13.94	1.81	11.92	10792.5	24.23	0.00	24.23	10800.0	59.71	0.04	1.24	1459.54	58.13	390.89
50	5	50	24.33	2.65	21.16	10792.4	28.86	0.12	28.70	10800.0	56.69	0.04	3.09	58.10	53.61	610.32
50	10	50	21.84	4.36	16.73	10792.3	17.82	0.00	17.82	10800.0	37.30	0.05	1.98	2.66	31.87	919.93
50	3	75	10.47	2.40	7.88	10792.4	17.20	0.00	17.20	10800.0	44.24	0.05	0.84	1134.35	42.95	873.14
50	5	75	16.60	2.52	13.70	10792.5	19.73	0.00	19.73	10800.0	42.59	0.06	2.59	76.35	40.38	1455.24
50	10	75	17.85	2.14	15.41	10792.6	17.10	0.35	16.69	10800.0	35.77	0.08	3.94	4.32	31.11	2317.03
Avg			17.04	2.49	14.19	10792.4	18.38	0.05	18.31	10486.4	48.77	0.04	1.97	387.17	45.27	903.06
55	3	25	12.91	2.13	10.54	10796.4	9.29	0.08	9.21	10800.0	57.31	0.01	0.79	2527.69	57.13	169.40
55	5	25	23.26	3.66	18.89	10795.7	28.92	0.00	28.92	10800.0	69.31	0.02	1.37	183.96	66.30	389.81
55	10	25	26.92	3.03	23.21	10793.7	23.66	0.00	23.66	10800.0	51.07	0.03	2.52	3.27	43.72	793.73
55	3	50	19.81	2.15	17.29	10796.4	35.11	0.11	34.96	10800.0	67.15	0.05	3.88	3659.88	65.70	656.78
55	5	50	31.03	4.54	25.34	10795.6	36.68	0.00	36.68	10800.0	62.47	0.05	3.52	334.96	59.98	1019.73
55	10	50	27.17	2.87	23.61	10793.6	25.52	0.26	25.20	10800.0	44.25	0.06	4.91	6.31	39.04	1422.40
55	3	75	13.53	2.55	10.70	10796.3	23.18	0.00	23.17	10800.0	46.56	0.07	1.15	3570.32	45.46	1049.39
55	5	75	16.38	1.67	14.44	10795.6	23.28	0.07	23.19	10800.0	42.76	0.07	2.26	207.63	40.89	1237.25
55	10	75	20.57	1.64	18.61	10793.5	21.29	0.23	21.02	10800.0	36.90	0.08	5.35	8.54	33.14	1581.45
Avg			21.29	2.69	18.07	10795.2	25.21	0.08	25.11	10800.0	53.09	0.05	2.86	1166.95	50.15	924.44
60	3	25	18.19	3.64	14.04	10795.7	18.67	0.00	18.67	10800.0	62.73	0.02	3.43	3682.17	62.54	415.61
60	5	25	33.09	7.33	23.99	10794.7	38.30	0.00	38.30	10800.0	76.75	0.03	1.73	1791.48	74.11	1056.09
60	10	25	38.17	7.80	28.16	10793.7	31.99	0.00	31.99	10800.0	58.47	0.05	3.29	12.37	51.91	2284.78
60	3	50	23.04	4.10	18.20	10795.8	37.20	0.00	37.20	10800.0	67.25	0.06	5.93	3841.32	66.00	1033.00
60	5	50	33.26	4.59	27.42	10794.7	42.16	0.28	41.76	10800.0	65.77	0.06	4.01	1259.54	63.47	1798.34
60	10	50	32.69	3.60	28.17	10793.5	31.13	0.01	31.11	10800.0	51.20	0.07	6.92	23.76	46.63	1795.76
60	3	75	15.06	3.60	11.09	10795.6	25.10	0.00	25.10	10800.0	46.46	0.10	2.94	3735.64	45.55	1990.74
60	5	75	20.42	3.16	16.80	10794.9	27.21	0.07	27.12	10800.0	44.87	0.09	3.48	1211.01	43.47	2031.27
60	10	75	20.93	1.34	19.32	10793.3	23.82	0.31	23.44	10800.0	38.25	0.13	6.66	19.91	35.09	2214.12
Avg			26.10	4.35	20.80	10794.7	30.62	0.07	30.52	10800.0	56.86	0.07	4.26	1730.80	54.31	1624.41
Ov.Avg			16.40	2.37	13.61	10389.1	18.41	0.06	18.33	10008.3	48.73	0.04	2.20	667.51	45.25	817.60

Table 4: CPLEX solution and upper bounds for larger instances. Average percentage gap and CPU time over 5 instances. Time limit: 1 hour (3 hours for CPLEX runs).

may be rather large for $n > 50$. The bounds provided by `D-Lgr DRLT1 MT1R` are pretty close to those found by `LP-MDRLT1` despite a considerably larger computational effort required by the bundle method. Considering the size of gaps for the LP relaxation, it comes as no surprise that `CPLEX` cannot find an optimal solution for most of the instances. The `%gapL` values confirm that the MDRLT1 reformulation leads to a better CPLEX performance.

We finally mention that it turned out to be impossible to execute the experiments with even larger instances, as already for $n = 60$ `CPLEX` required more than 30 GB of memory to solve an instance. Observe that, in any case, the `%gap` values in the table are still likely to overestimate the real gaps.

6 Conclusions

Over the last 15 years, the quadratic multiple knapsack problem has received increasing attention from the literature, dealing almost exclusively with meta-heuristics. Although in 2019 Bergman [2] presented the first (and only) specialized Branch-and-Price algorithm, the problem has never been studied from a broader mathematical perspective. We attempted to fill this gap, by focusing on classical reformulations and relaxations and analyzing their properties, in order to gain insight into the strengths and weaknesses of such methods. Currently, Bergman's algorithm can solve instances up to 10 knapsacks and 35 items. Yet, the original benchmark instances considered in the literature (for heuristic solutions) are one order of magnitude larger, involving up to 30 knapsacks and 300 items. We believe our results have implications for the development of future exact algorithms capable of tackling larger instances. Indeed, in an enumerative algorithm, a trade-off must be made between the quality of the upper bound and the time taken to compute it. Our results suggest that, among the different possible approaches, the most promising is the one based on the Lagrangian relaxation of the cardinality constraints of the 0-1 quadratic model, both in terms of bound quality and CPU time. In particular, the convergence of the proximal bundle method to solve the Lagrangian dual problem appears to be very fast. Another interesting observation is that the adoption of non-optimal solutions of the Lagrangian subproblems speeds up the computation of each bundle iteration without deteriorating the bounds significantly. This turns out to be true for both Lagrangian relaxations we have considered. Our experiments also show that the use of specialized methods to solve the subproblems can be crucial to reduce the computing time, and should always be preferred, when possible, to general purpose MIP solvers.

Acknowledgments

This research was supported by the Air Force Office of Scientific Research under Grant no. FA8655-20-1-7012 and by National Agency for Research and Development (ANID)/ Scholarship Program/Doctorado Becas Chile/2018-72190600. We thank Antonio Frangioni and David Pisinger for the support in the use of their codes.

References

[1] W.P. Adams and H.D. Sherali. A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32(10):1274–1290, 1986.

[2] D. Bergman. An exact algorithm for the quadratic multiknapsack problem with an application to event seating. *INFORMS Journal on Computing*, 31(3):477–492, 2019.

[3] A. Billionnet and E. 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.

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

[5] A. Caprara, D. Pisinger, and P. Toth. Exact solution of the quadratic knapsack problem. *INFORMS Journal on Computing*, 11(2):125–137, 1999.

[6] Y. Chen and J.K. Hao. Iterated responsive threshold search for the quadratic multiple knapsack problem. *Annals of Operations Research*, 226(1):101–131, 2015.

[7] Y. Chen, J.K. Hao, and F. Glover. An evolutionary path relinking approach for the quadratic multiple knapsack problem. *Knowledge-Based Systems*, 92:23–34, 2016.

[8] R. Fortet. L'algèbre de Boole et ses applications en recherche opérationnelle. *Cahiers du Centre d'Études de Recherche Opérationnelle*, 1(4):5–36, 1959.

[9] A. Frangioni. Generalized Bundle Methods. *SIAM Journal on Optimization*, 13(1):117–156, 2002.

[10] F. Furini and E. Traversi. Theoretical and computational study of several linearisation techniques for binary quadratic problems. *Annals of Operations Research*, 279(1-2):387–411, 2019.

[11] G. Gallo, P.L. Hammer, and B. Simeone. Quadratic knapsack problems. *Mathematical Programming*, pages 132–149, 1980.

[12] C. García-Martínez, F. Glover, F.J. Rodriguez, M. Lozano, and R. Martí. Strategic oscillation for the quadratic multiple knapsack problem. *Computational Optimization and Applications*, 58(1):161–185, 2014.

[13] C. García-Martínez, F.J. Rodriguez, and M. Lozano. Tabu-enhanced iterated greedy algorithm: a case study in the quadratic multiple knapsack problem. *European Journal of Operational Research*, 232(3):454–463, 2014.

[14] F. Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.

[15] F. Glover and E. Woolsey. Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations Research*, 22(1):180–182, 1974.

[16] M. Guignard. Strong RLT1 bounds from decomposable Lagrangean relaxation for some quadratic 0–1 optimization problems with linear constraints. *Annals of Operations Research*, pages 173–200, 2020.

[17] A. Hiley and B.A. Julstrom. The quadratic multiple knapsack problem and three heuristic approaches to it. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 547–552, 2006.

[18] B.A. Julstrom. Greedy, genetic, and greedy genetic algorithms for the quadratic knapsack problem. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, pages 607–614, 2005.

[19] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer Nature Book Archives Millennium. Springer, 2004.

[20] E. Lalla-Ruiz, E. Segredo, and S. Voß. A cooperative learning approach for the quadratic knapsack problem. In *International Conference on Learning and Intelligent Optimization*, pages 31–35, 2018.

[21] S. Martello and P. Toth. A bound and bound algorithm for the zero-one multiple knapsack problem. *Discrete Applied Mathematics*, 3(4):275–288, 1981.

[22] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.

[23] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*. John Wiley & Sons, 1998.

[24] D. Pisinger. Upper bounds and exact algorithms for p-dispersion problems. *Computers & Operations Research*, 33(5):1380–1398, 2006.

[25] D. Pisinger. The quadratic knapsack problem - a survey. *Discrete Applied Mathematics*, 155(5):623–648, 2007.

[26] W.D. Pisinger, A.B. Rasmussen, and R. Sandvik. Solution of large quadratic knapsack problems through aggressive reduction. *INFORMS Journal on Computing*, 19(2):280–290, 2007.

[27] S. Pulikanti and A. Singh. An artificial bee colony algorithm for the quadratic knapsack problem. In *International Conference on Neural Information Processing*, pages 196–205. Springer, 2009.

[28] J. Qin, X. Xu, Q. Wu, and T.C.E. Cheng. Hybridization of tabu search with feasible and infeasible local searches for the quadratic multiple knapsack problem. *Computers & Operations Research*, 66:199–214, 2016.

[29] T. Saraç and A. Sipahioglu. Generalized quadratic multiple knapsack problem and two solution approaches. *Computers & Operations Research*, 43(1):78–89, 2014.

[30] H.D. Sherali and W.P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[31] A. Singh and A.S. Baghel. A new grouping genetic algorithm for the quadratic multiple knapsack problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 210–218. Springer, 2007.

[32] SM. Soak and SW. Lee. A memetic algorithm for the quadratic multiple container packing problem. *Applied Intelligence*, 36(1):119–135, 2012.

[33] S. Sundar and A. Singh. A swarm intelligence approach to the quadratic multiple knapsack problem. In *International Conference on Neural Information Processing*, pages 626–633. Springer, 2010.

[34] T. Tlili, H. Yahyaoui, and S. Krichen. An iterated variable neighborhood descent hyperheuristic for the quadratic multiple knapsack problem. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2015*, pages 245–251. Springer, 2016.

[35] C. Witzgall. Mathematical methods of site selection for electronic message systems (EMS). *NASA STI/Recon Technical Report N*, 76, 1975.