Available online at www.sciencedirect.com
ScienceDirect
Discrete Applied Mathematics 155 (2007) 623–648
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

The quadratic knapsack problem—a survey

David Pisinger

Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen, Denmark

Received 14 May 2003; received in revised form 20 May 2005; accepted 6 August 2006
Available online 20 October 2006

Abstract

The binary *quadratic knapsack problem* maximizes a quadratic objective function subject to a linear capacity constraint. Due to its simple structure and challenging difficulty it has been studied intensively during the last two decades. The present paper gives a survey of upper bounds presented in the literature, and show the relative tightness of several of the bounds. Techniques for deriving the bounds include relaxation from upper planes, linearization, reformulation, Lagrangian relaxation, Lagrangian decomposition, and semidefinite programming. A short overview of heuristics, reduction techniques, branch-and-bound algorithms and approximation results is given, followed by an overview of valid inequalities for the quadratic knapsack polytope. The paper is concluded by an experimental study where the upper bounds presented are compared with respect to strength and computational effort.

Keywords: Quadratic knapsack problem; Upper bounds; Approximation algorithms; Semidefinite programming; Valid inequalities

1. Introduction

The binary *quadratic knapsack problem* (QKP) was introduced by Gallo et al. [18]. Formally it may be defined as follows: assume that n items are given where item j has a positive integer weight w_j. In addition we are given an $n \times n$ nonnegative integer matrix $P = \{p_{ij}\}$, where p_{jj} is the profit achieved if item j is selected and $p_{ij} + p_{ji}$ is a profit achieved if both items i and j are selected for $i < j$. The QKP calls for selecting an item subset whose overall weight does not exceed a given knapsack capacity c, so as to maximize the overall profit. For notational convenience, let $N := \{1, \ldots, n\}$ denote the item set. By introducing a binary variable x_j to indicate whether item j is selected, the problem may be formulated:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in N} \sum_{j \in N} p_{ij} x_i x_j \\ \text{subject to} \quad & \sum_{j \in N} w_j x_j \leqslant c, \\ & x_j \in \{0, 1\}, \quad j \in N. \end{aligned} \tag{1}$$

Without loss of generality we assume that $\max_{j \in N} w_j \leqslant c < \sum_{j \in N} w_j$ and that the profit matrix is symmetric, i.e., $p_{ij} = p_{ji}$ for all $j > i$. Notice, that if negative weights $w_j < 0$ are present, we may flip variable x_j to $1 - x_j$. If $w_j > c$

E-mail address: pisinger@diku.dk.

0166-218X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2006.08.007

we may fix $x_j = 0$, and if $w_j = c$ then we may decompose the problem. Hence, normally $0 \leqslant w_j < c$. If $p_{ij} \geqslant 0$ for all coefficients $i \neq j$ the QKP is denoted the *supermodular knapsack problem*. In the sequel we will rely on the stronger assumption that all coefficients $p_{ij} \geqslant 0$, $i, j \in N$ which is not made without loss of generality. Where upper bounds and other results are valid for a more general model, it will be stated in the text. QKP in all the above mentioned forms is $\mathcal{NP}$-hard in the strong sense, which can be seen by reduction from the clique problem.

One may give several graph-theoretic interpretations to QKP: given a complete undirected graph on node set *N*, where each node *j* has a profit p_{jj} and weight w_j and each edge (i, j) has a profit $p_{ij} + p_{ji}$, select a node subset $S \subseteq N$ whose overall weight does not exceed *c* so as to maximize the overall profit, given by the sum of the profits of the nodes in *S* and of the edges with both endpoints in *S*. It is then easy to see that QKP is also a generalization of the *clique problem*. This latter problem, in its recognition version, calls for checking whether, for a given positive integer *k*, a given undirected graph $G = (V, E)$ contains a complete subgraph on *k* nodes. A possible optimization version of clique is given by the so-called *dense subgraph problem*, in which one wants to select a node subset $S \subseteq V$ of cardinality $|S| = k$ such that the subgraph of *G* induced by *S* contains as many edges as possible. This problem can be modeled as (1) by setting $n := |V|$; $c := k$; $w_j := 1$ for $j \in N$; $p_{ij} := p_{ji} := 1$ if $(i, j) \in E$ and $p_{ij} := p_{ji} := 0$ otherwise, for $i, j \in N$. Note that in this case the knapsack constraint reduces to a cardinality constraint, and will be satisfied with equality by the optimal solution. Clearly, the answer to clique is positive if and only if the optimal solution of this QKP has value $k(k-1)$. The most famous optimization version of clique, called *max clique*, calls for an induced complete subgraph with a maximum number of nodes. This latter problem can be solved through a QKP algorithm by using binary search with *c* between 1 and *n*.

QKP is a generalization of the 0–1 *knapsack problem* (KP) which arises when $p_{ij} = 0$ for all $i \neq j$. Solution techniques for KP are considered in e.g. Martello and Toth [31], Pisinger and Toth [39] and Kellerer et al. [48]. Not surprising, several upper bounds for QKP rely on some kind of relaxation to KP. QKP is also a constrained version of the *Quadratic 0–1 Programming Problem* (QP) which is defined as (1) without the capacity constraint. Since the set of solutions for QKP is a subset of QP, all valid inequalities for QP are also valid for QKP and hence they can be used to tighten bounds for the QKP.

A special version of QKP appears when restricting the problem to a *diagonal* profit matrix *P*, such that $p_{ij} = 0$ for $i \neq j$. The *integer diagonal* QKP, where variables may take on any integer value between a lower and an upper bound, is considered by Bretthauer et al. [7]. Brucker [8] presented an $O(n)$ algorithm for solving the LP-relaxation of this problem.

Although QKP has not been studied as intensively as the related *quadratic assignment problem* (see e.g. [11] for a survey) numerous papers dealing with the problem have been presented during the last years. Gallo et al. [18] invented the QKP and presented a family of upper bounds based on *upper planes*, which are linear functions of the binary variables satisfying that their value is not smaller than the QKP objective function over the set of feasible QKP solutions. Johnson et al. [29] considered the graph version of the QKP. After linearization of the objective function, the model is solved by a branch-and-cut system in which tree inequalities and star inequalities are used to tighten the formulation. Billionnet and Calmels [5] used a classical linearization technique of the objective function to obtain an ILP formulation. As the linearized model may grow quite large, a delayed formulation method is used in a branch-and-cut manner. Lagrangian relaxation approaches are described by Chaillou et al. [12]. Relaxing the capacity constraint, a quadratic optimization problem appears which is solvable in polynomial time through a maximum flow problem. Michelon and Veilleux [32] used a Lagrangian decomposition technique to split the problem into a quadratic 0–1 optimization problem and a KP. The quadratic optimization problem has some nice properties, which makes it solvable by use of the techniques introduced by Chaillou et al. Hammer and Rader [24] used the upper bound by Chaillou et al. in their computational study, but improved the algorithm by using order relations to fix variables inside a branch-and-bound algorithm. Hammer and Rader also presented an improved heuristic based on the best linear approximation of QKP. Helmberg et al. [27] consider a more general version of the problem where *P* may have negative entries. Several upper bounds are presented based on a cascade of semidefinite programming relaxations. To strengthen the formulation a number of valid inequalities are derived based on the ordinary KP polyhedron, as well as specific inequalities for the QKP polyhedron. Caprara et al. [9] used Lagrangian relaxation of the symmetry constraint $x_i x_j = x_j x_i$ to reach a reformulation of the problem through subgradient optimization. Using the reformulated problem, upper bounds tighter than those presented by Gallo, Hammer, Simeone can be derived in $O(n)$ expected time inside a branch-and-bound algorithm. Billionnet et al. [6] presented a bound based on the partitioning of *N* into *m* disjoint classes. Using Lagrangian decomposition, the problem can be split into *m* independent subproblems, which are easier to solve. Rader and Woeginger [40] finally presented a fully polynomial approximation scheme for a QKP defined on an edge series–parallel graph. They also proved that QKP with positive and negative profits does not have any polynomial time approximation algorithm with fixed approximation ratio.

As one might expect, due to its generality, QKP has a wide spectrum of applications. Witzgall [46] presented a problem which arises in telecommunications when a number of sites for satellite stations have to be selected, such that the global traffic between these stations is maximized and a budget constraint is respected. This problem appears to be a QKP. Similar models arise when considering the location of airports, railway stations or freight handling terminals [42]. Johnson et al. [29] mention a compiler design problem which may be formulated as a QKP, as described in [27]. Dijkhuizen and Faigle [14] and Park et al. [35] consider the weighted maximum b-clique problem. If all edge weights are nonnegative this problem is the special case of QKP arising when $w_j = 1$ for $j \in N$ and $b = c$. Ferreira et al. [16] consider a problem in VLSI design where large graphs need to be decomposed into smaller graphs of tractable size. The corresponding optimization problem for a single subgraph can be recognized as a QKP in the minimization form. Finally, QKP appears as the pricing problem when solving a graph partitioning problem through column generation as described in Johnson et al. [29].

The paper is organized as follows. In the following section we will give a survey of the most important upper bounds for QKP. The bounds are presented according to the relaxation techniques used, starting from upper planes and followed by linearization, Lagrangian relaxation, reformulation, Lagrangian decomposition and finally semidefinite programming. Section 3 shows how the size of a QKP instance may be reduced by fixing some variables at their optimal values. Section 4 describes a number of heuristics for the solution of QKP, including primal algorithms, dual algorithms, and algorithms based on reformulation. Next, Section 5 presents a few results on approximation of QKP, Section 6 is devoted to polyhedral results, and Section 7 deals with exact algorithms based on branch-and-bound. An experimental study, comparing the tightness and computational effort of the bounds is presented in Section 8. The paper is concluded in Section 9 with a discussion of future challenges.

2. Upper bounds

Numerous upper bounds have been presented for the QKP during the last two decades. The bounds are based on a variety of techniques including: linearization, Lagrangian relaxation, derivation of upper planes, semidefinite relaxations and reformulation techniques. In the following we will give a detailed survey of the most important bounds.

The following table gives an overview of the upper bounds presented in the succeeding sections.

Paper	Technique	Bounds
Gallo, Hammer, Simeone [18]	Upper planes	$U^1_{\mathrm{GHS}}, U^2_{\mathrm{GHS}}, U^3_{\mathrm{GHS}}, U^4_{\mathrm{GHS}}$
Chaillou, Hansen, Mahieu [12]	Lagrangian relaxation	U_{CHM}
Billionnet and Calmels [5] Johnson, Mehrotra, Nemhauser [29]	Linearization, cutting plane method	$U^1_{\mathrm{BC}}, U^2_{\mathrm{BC}}$
Caprara, Pisinger, Toth [9]	Lagrangian relaxation, reformulation	$U^1_{\mathrm{CPT}}, U^2_{\mathrm{CPT}}, \hat{U}^2_{\mathrm{CPT}}$
Michelon and Veuilleux [32]	Lagrangian decomposition	$U^1_{\mathrm{MV}}, U^2_{\mathrm{MV}}, \hat{U}^2_{\mathrm{MV}}$
Billionnet, Faye, Soutif [6]	Lagrangian decomposition	$U^1_{\mathrm{BFS}}, U^2_{\mathrm{BFS}}, \hat{U}^2_{\mathrm{BFS}}$
Helmberg, Rendl, Weismantel [27,28]	Semidefinite programming	$U^0_{\mathrm{HRW}}, U^1_{\mathrm{HRW}}, U^2_{\mathrm{HRW}}, U^3_{\mathrm{HRW}}, U^4_{\mathrm{HRW}}$

2.1. Gallo, Hammer, Simeone

Gallo et al. [18] presented the first bounds for QKP using the concept of *upper plane*. An upper plane (or *linear majorization function*) is a linear function g satisfying $g(x) \geqslant \sum_{i\in N}\sum_{j\in N} p_{ij}x_i x_j$ for any feasible solution x of (1). The actual upper planes proposed in [18] are of the form $\sum_{j\in N} \pi_j x_j$, for $j \in N$ leading to the following relaxed optimization problem

$$\begin{aligned}
\text{maximize} \quad & \sum_{j \in N} \pi_j x_j \\
\text{subject to} \quad & \sum_{j \in N} w_j x_j \leqslant c, \\
& x_j \in \{0, 1\}, \quad j \in N.
\end{aligned} \tag{2}$$

The problem is recognized as an ordinary KP since we have a linear objective function $g(\overline{x})$ subject to a linear capacity constraint. Gallo, Hammer, Simeone consider four different expressions of π_j, leading to the bounds U^1_{GHS}, U^2_{GHS}, U^3_{GHS} and U^4_{GHS}. The bound U^1_{GHS} is obtained by using the upper plane

$$\pi^1_j := \sum_{i \in N} p_{ij}, \tag{3}$$

which obviously is valid since $\sum_{j\in N}\sum_{i\in N} p_{ij}\overline{x}_i\overline{x}_j \leqslant \sum_{j\in N}(\sum_{i\in N} p_{ij})\overline{x}_j \leqslant \sum_{j\in N}\pi^1_j\overline{x}_j$.

Bound U^2_{GHS} is based on the upper plane

$$\pi^2_j := \max\left\{\sum_{i\in N} p_{ij}\overline{x}_i : \sum_{i\in N}\overline{x}_i \leqslant k, \overline{x}_i \in \{0,1\} \text{ for } i \in N\right\}, \tag{4}$$

where k is the maximum cardinality of a feasible QKP solution which may be obtained by sorting the items according to nondecreasing weights and setting

$$k = \min\left\{h : \sum_{j=1}^{h} w_j > c\right\} - 1. \tag{5}$$

In other words π^2_j is the sum of the k biggest profits among $\{p_{1j}, \ldots, p_{nj}\}$. We notice that the upper plane is valid since $\sum_{j\in N}(\sum_{i\in N} p_{ij}\overline{x}_i)\overline{x}_j \leqslant \sum_{j\in N}\pi^2_j\overline{x}_j$ for any feasible solution $\overline{x}$.

The next bound, U^3_{GHS}, is obtained by using the upper plane

$$\pi^3_j := \max\left\{\sum_{i\in N} p_{ij}\overline{x}_i : \sum_{i\in N} w_i\overline{x}_i \leqslant c, 0 \leqslant \overline{x}_i \leqslant 1 \text{ for } i \in N\right\}, \tag{6}$$

which appears by noting that $\sum_{i\in N} p_{ij}x_i$ subject to the constraint $\sum_{i\in N} w_i x_i \leqslant c$, $x_i \in \{0,1\}$ does not exceed π^3_j.

Finally U^4_{GHS} is derived by setting

$$\pi^4_j := \max\left\{\sum_{i\in N} p_{ij}\overline{x}_i : \sum_{i\in N} w_i\overline{x}_i \leqslant c, \overline{x}_i \in \{0,1\} \text{ for } i \in N\right\}. \tag{7}$$

The validity of this bound is checked as above.

Due to the solution of the 0–1 KP (2) none of the bounds U^1_{GHS}, U^2_{GHS}, U^3_{GHS}, U^4_{GHS} have polynomial time bounds. However, if we solve the continuous relaxation of (2) we obtain the weaker bounds U^{1*}_{GHS}, U^{2*}_{GHS}, U^{3*}_{GHS}, U^{4*}_{GHS}. Of these bounds U^{1*}_{GHS}, U^{2*}_{GHS}, U^{3*}_{GHS} can be obtained in $O(n^2)$ time, while it is $\mathcal{NP}$-hard to derive U^{4*}_{GHS}. Caprara et al. [9] noticed that coefficients π^2_j, π^3_j and π^4_j can be improved by forcing $\overline{x}_j = 1$ in the computation of π_j. We will denote these bounds by $\overline{U}^2_{\mathrm{GHS}}$, $\overline{U}^3_{\mathrm{GHS}}$ and $\overline{U}^4_{\mathrm{GHS}}$.

It can easily be verified that $U^4_{\mathrm{GHS}} \leqslant U^3_{\mathrm{GHS}}$ and $U^4_{\mathrm{GHS}} \leqslant U^2_{\mathrm{GHS}}$. Moreover $U^3_{\mathrm{GHS}} \leqslant U^1_{\mathrm{GHS}}$ and $U^2_{\mathrm{GHS}} \leqslant U^1_{\mathrm{GHS}}$. Hence U^4_{GHS} is the tightest of the bounds. No dominance exists between U^2_{GHS} and U^3_{GHS} as can be seen by constructing the following two instances: Instance 1 has $n = 3$, $c = 3$, $w = (1, 2, 2)$, and p given by $p_{23} = p_{33} = 2$ and other values are $p_{ij} = 0$ for $i, j \in \{1, 2, 3\}$. Instance 2 has $n = 2$, $c = 3$, $w = (2, 2)$, and p given by $p_{ij} = 2$ for all $i, j \in \{1, 2\}$. In instance 1 we find $U^2_{\mathrm{GHS}} = 4$ and $U^3_{\mathrm{GHS}} = 3$ hence $U^2_{\mathrm{GHS}} \nleqslant U^3_{\mathrm{GHS}}$. The opposite situation appears in instance 2, where we find $U^2_{\mathrm{GHS}} = 2$ and $U^3_{\mathrm{GHS}} = 3$ hence $U^3_{\mathrm{GHS}} \nleqslant U^2_{\mathrm{GHS}}$.

Gallo, Hammer, Simeone experimentally showed that the upper bound U^{3*}_{GHS} gives the best trade-off between tightness and computational effort in a branch-and-bound algorithm.

2.2. Chaillou, Hansen, Mahieu

Chaillou et al. [12] Lagrangian relaxed the capacity constraint in QKP using multiplier $\lambda \geqslant 0$, getting the problem

$$\begin{aligned} &\text{maximize} \quad \sum_{i\in N}\sum_{j\in N} p_{ij}x_ix_j - \lambda\left(\sum_{j\in N} w_jx_j - c\right)\\ &\text{subject to} \quad x_j \in \{0,1\}, \quad j\in N. \end{aligned} \tag{8}$$

By setting $\tilde{p}_{ij} = p_{ij}$ if $i \neq j$ and $\tilde{p}_{ij} = p_{ij} - \lambda w_j$ if $i = j$, the relaxed problem can be reformulated as a QP denoted $L^c(\mathrm{QKP}, \lambda)$ on the form

$$\begin{aligned} &\text{maximize} \quad \sum_{i\in N}\sum_{j\in N} \tilde{p}_{ij}x_ix_j + \lambda c\\ &\text{subject to} \quad x_j \in \{0,1\}, \quad j\in N. \end{aligned} \tag{9}$$

Picard and Ratliff [36] showed that the latter problem can be solved in polynomial time, since the matrix $\{\tilde{p_{ij}}\}$ has nonnegative off diagonal elements. Chaillou et al. [12] presented a simple algorithm based on the solution of a *maximum flow problem* in a network. The network $\mathscr{N} = (V, E, \overline{c})$ is defined with a vertex set $V = \{s, 1, \ldots, n, t\}$ where s is the source and t is the sink. The edge set E is defined on $V \times V$ having the capacities

$$\begin{aligned} \overline{c}_{si}(\lambda) &= \max\left(0, \sum_{j\in N} p_{ij} - \lambda w_i\right),\\ \overline{c}_{ij}(\lambda) &= p_{ij},\\ \overline{c}_{it}(\lambda) &= \max\left(0, \lambda w_i - \sum_{j\in N} p_{ij}\right). \end{aligned} \tag{10}$$

Assume that the value of the maximum flow from s to t in $\mathscr{N}$ is $\Psi(\mathscr{N})$. Then Chaillou, Hansen, Mahieu showed that the corresponding solution to (8) is given by

$$z(L^c(\mathrm{QKP}, \lambda)) = \lambda c + \sum_{i\in N} \overline{c}_{si}(\lambda) - \Psi(\mathscr{N}). \tag{11}$$

The Lagrangian dual problem corresponding to (8) is

$$\min_{\lambda \geqslant 0} L^c(\mathrm{QKP}, \lambda). \tag{12}$$

Let U_{CHM} denote the bound corresponding to optimal solution of (12). Chaillou, Hansen, Mahieu proved that the Lagrangian dual is a piecewise linear function of λ with at most n linear segments. This observation can be used to construct a simple binary search algorithm for determining the optimal choice λ^* of Lagrangian multiplier. Since there are at most n linear segments, no more than $O(n)$ iterations are needed, each iteration solving a maximum flow problem on a graph with $n + 2$ vertices, and $2n + n^2$ edges. Gallo et al. [17] further improved the complexity of solving the Lagrangian dual problem by taking advantage of the similarity of successive maximum flow problems.

As shown by Geoffrion [20], the bound obtained from the Lagrangian dual problem is equivalent to the bound obtained through continuous relaxation, hence $U_{\mathrm{CHM}} = U_{\mathrm{CONT}}$, where the latter bound corresponds to the continuous relaxation of QKP.

2.3. Billionnet, Calmels

An obvious approach for deriving an upper bound for QKP is to linearize the quadratic term, and then solve the LP-relaxation of the problem. Billionnet and Calmels [5] presented a bound based on this principle, by introducing variables y_{ij} for $i<j$ which attain the value 1 if and only if $x_i=1$ and $x_j=1$. We may formulate this equivalence by the constraints

$$y_{ij} \leqslant x_i, \quad y_{ij} \leqslant x_j, \quad x_i + x_j \leqslant 1 + y_{ij} \tag{13}$$

which leads to the following ILP model

$$\begin{aligned} \text{maximize} \quad & \sum_{i,j\in N, i<j} 2p_{ij}y_{ij} + \sum_{j\in N} p_{jj}x_j \\ \text{subject to} \quad & \sum_{j\in N} w_j x_j \leqslant c \\ & y_{ij} \leqslant x_i, \ \ i,j\in N, \ \ i<j, \\ & y_{ij} \leqslant x_j, \ \ i,j\in N, \ \ i<j, \\ & x_i + x_j \leqslant 1 + y_{ij}, \ \ i,j\in N, \ \ i<j, \\ & x_j, \ \ y_{ij} \in \{0,1\}, \ \ i,j\in N, \ \ i<j. \end{aligned} \tag{14}$$

Since $p_{ij} \geqslant 0$ the constraints $x_i+x_j-1 \leqslant y_{ij}$ is not strictly necessary. Relaxing the integrality constraints to $0 \leqslant x_i \leqslant 1$ and $y_{ij} \geqslant 0$ we get the bound U^1_{BC}, which according to Billionnet, Calmels is of moderate quality. Note that the formulation (14) is identical to the formulation used by Johnson et al. [29] for the graph version of the QKP.

In order to tighten the above formulation a number of additional constraints are added to the model. By multiplying the capacity constraint $\sum_{i\in N} w_i x_i \leqslant c$ with x_j for each $j\in N$ and replacing x_j^2 by x_j we obtain n new constraints

$$\sum_{i\in N\setminus\{j\}} w_i x_i x_j \leqslant (c-w_j)x_j, \quad j\in N. \tag{15}$$

Although these constraints are redundant for the IP-formulation, they may tighten the LP-relaxation of (14). Such constraints are also used in Helmberg et al. [27], and Caprara et al. [9], and they may be seen as application of a general procedure proposed by Adams and Sherali [1] and further studied by Lovász and Schrijver [30] and Balas et al. [3].

Further, for each three indices $i \neq j \neq k$ we derive a *Chvatal–Gomory cut* on the form $x_i+x_j+x_k-y_{ij}-y_{jk}-y_{ki} \leqslant 1$ by adding together three of the constraints $x_i+x_j \leqslant 1+y_{ij}$ in (14), dividing by two, and rounding down. This leads to the following formulation

$$\begin{aligned} \text{maximize} \quad & \sum_{i,j\in N, i<j} 2p_{ij}y_{ij} + \sum_{j\in N} p_{jj}x_j, \\ \text{subject to} \quad & \sum_{j\in N} w_j x_j \leqslant c, && (16) \\ & y_{ij} \leqslant x_i, \quad i,j\in N, i<j, && (17) \\ & y_{ij} \leqslant x_j, \quad i,j\in N, i<j, && (18) \\ & x_i + x_j \leqslant 1 + y_{ij}, \quad i,j\in N, \ \ i<j, && (19) \\ & 0 \leqslant x_j \leqslant 1, \quad j\in N, && (20) \\ & y_{ij} \geqslant 0, \quad i,j\in N, \ \ i<j, && (21) \\ & \sum_{i\in N\setminus\{j\}} w_i y_{ij} \leqslant (c-w_i)x_j, \quad j\in N, && (22) \\ & x_i + x_j + x_k - y_{ij} - y_{jk} - y_{ki} \leqslant 1, \quad i<j<k. && (23) \end{aligned}$$

Solving the model gives the bound U^2_{BC} which according to Billionnet, Calmels is quite tight. The model has $\mathrm{O}(n^2)$ variables and $\mathrm{O}(n^3)$ constraints. This means that it is quite time consuming to derive the bound for large-sized instances. Billionnet, Calmels hence propose to solve the model defined on (16), (20), (21) and (22) only. This model has $\mathrm{O}(n)$ constraints, hence being easy to solve. Then, a search is performed which finds all constraints among (17)–(19) and (23) which are not satisfied. These constraints are added to the model, and the process is repeated progressively.

2.4. Caprara, Pisinger, Toth

The bound by Caprara et al. [9] can be described within the framework of upper planes as follows. First we derive a new upper plane as

$$\pi_j^5 := p_{jj} + \max\left\{ \sum_{i \in N\setminus\{j\}} p_{ij}\overline{x}_i : \sum_{i\in N\setminus\{j\}} w_i\overline{x}_i \leqslant (c - w_j), \overline{x}_i \in \{0,1\} \text{ for } i \in N\setminus\{j\} \right\}. \tag{24}$$

Then an upper bound U_{CPT}^1 is derived as the optimal solution value to (2). If we relax the integrality constraints in the above subproblems we obtain the bound U_{CPT}^{1*} which can be derived in $\mathrm{O}(n^2)$ time by solving the n continuous KP on the form (24) and one continuous KP on the form (2).

Caprara et al. [9] further strengthened the bound by noting that the objective function can be reformulated as

$$\sum_{i\in N}\sum_{j\in N} p_{ij}x_ix_j = \sum_{i\in N}\sum_{j\in N}(p_{ij}+\lambda_{ij})x_ix_j \tag{25}$$

for any matrix satisfying that $\lambda_{ij} = -\lambda_{ji}$. I.e. $\Lambda = \{\lambda_{ij}\}$ should be a *skew-symmetric matrix*. Let $\mathrm{QKP}(\Lambda)$ denote the problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i\in N}\sum_{j\in N}(p_{ij}+\lambda_{ij})x_ix_j \\ \text{subject to} \quad & \sum_{j\in N} w_jx_j \leqslant c, \\ & x_j \in \{0,1\}, \quad j \in N \end{aligned} \tag{26}$$

and $U_{\mathrm{CPT}}^{1*}(\Lambda)$ the corresponding upper bound obtained by solving the n continuous KP on the form (24) and the continuous KP on the form (2). In order to obtain the tightest bound we solve the Lagrangian dual problem

$$U_{\mathrm{CPT}}^2 = \min_{\{\lambda_{ij}:\lambda_{ij}=-\lambda_{ji}\}} U_{\mathrm{CPT}}^{1*}(\Lambda). \tag{27}$$

The latter problem may be solved through subgradient optimization leading to the bound $\hat{U}_{\mathrm{CPT}}^2$ for some near-optimal matrix Λ of Lagrangian multipliers. Obviously $U_{\mathrm{CPT}}^2 \leqslant \hat{U}_{\mathrm{CPT}}^2 \leqslant U_{\mathrm{CPT}}^{1*}(0)$.

2.4.1. Corresponding ILP-formulation

In order to determine the relative tightness of bound U_{CPT}^1 and U_{CPT}^2 compared to previously presented bounds, Caprara et al. [9] consider the ILP-models corresponding to the bounds.

Bound U_{CPT}^1 may be seen as an extension of model (14) to which the redundant capacity constraints (15) have been added, and the constraints (13) have been replaced by

$$y_{ij} \leqslant x_j, \quad y_{ij} = y_{ji}. \tag{28}$$

Notice that the last constraints in (13) are not strictly necessary. These constraints could be used to tighten the LP relaxation of the model, but they cannot be handled by the upper plane algorithm used for solving the LP relaxation.

In this way we reach the ILP reformulation of QKP:

$$\text{maximize} \quad \sum_{j\in N}\sum_{i\in N\setminus\{j\}} p_{ij}y_{ij} + \sum_{j\in N} p_{jj}x_j \tag{29}$$

$$\text{subject to} \quad \sum_{j\in N} w_j x_j \leqslant c, \tag{30}$$

$$\sum_{i\in N\setminus\{j\}} w_i y_{ij} \leqslant (c-w_j)x_j, \quad j\in N, \tag{31}$$

$$0\leqslant y_{ij}\leqslant x_j\leqslant 1, \quad i,j\in N, j\neq i, \tag{32}$$

$$y_{ij}=y_{ji}, \quad i,j\in N, \quad j>i, \tag{33}$$

$$x_j, y_{ij}\in\{0,1\}, \quad i,j\in N, \quad j\neq i. \tag{34}$$

The reason for an explicit use of two distinct variables y_{ij} and y_{ji}, linked by equality constraints (33), will be clear in a moment. If Eqs. (33) are removed, the resulting LP relaxation (29)–(32) can be solved in a very effective way: For each $j\in N$, variables y_{ij} ($i\in N\setminus\{j\}$), besides having a lower bound of 0 and a variable upper bound of x_j, appear only in constraint (31) associated with j, and in the objective function. Hence, if variable x_j is fixed to value $\overline{x}_j$ for all $j\in N$, the relaxed problem decomposes into n independent subproblems, one for each $j\in N$, of the form (24).

The ILP formulation can also be used to characterize the bounds U^1_{GHS}–U^4_{GHS} by Gallo, Hammer, Simeone. Forcing $\overline{x}_j=1$ in the computation of π_j, the upper bounds coincide, respectively, with the following formulations:

- U^1_{GHS} corresponds to the solution of (29), (30), (32), and (34).
- U^{1*}_{GHS} corresponds to the solution of (29), (30), (32).
- U^2_{GHS} corresponds to the solution of (29), (30), (32), and (34) with the additional cardinality constraints $\sum_{i\in N\setminus\{j\}} y_{ij}\leqslant k-1$ for $j\in N$; where k is given by (5).
- U^{2*}_{GHS} corresponds to the solution of (29), (30), (32), with the additional cardinality constraints $\sum_{i\in N\setminus\{j\}} y_{ij}\leqslant k-1$ for $j\in N$; where k is given by (5).
- U^{3*}_{GHS} corresponds to the solution of (29)–(32), if the continuous relaxation of the final KP is solved.
- U^4_{GHS} corresponds to the solution of (29)–(32) and (34).

As bound U^1_{CPT} corresponds to the solution of (29)–(32) and (34), we immediately get that $U^1_{\text{CPT}}\leqslant U^4_{\text{GHS}}$ hence the bound by Caprara, Pisinger, Toth dominates all the bounds by Gallo, Hammer, Simeone.

Instead of removing constraints (33), we may obtain a tighter bound through Lagrangian relaxation, leading to the bound $U^{1*}_{\text{CPT}}(\Lambda)$. We introduce a matrix $\Lambda=\{\lambda_{ij}\}$, where, for $i,\ j\in N,\ j>i,\ \lambda_{ij}$ is the *Lagrangian multiplier* associated with the corresponding equation in (33) and, for notational convenience, $\lambda_{ji} := -\lambda_{ij}$. Accordingly, the Lagrangian modified objective function reads:

$$\text{maximize } z(\text{QKP}(\Lambda)) = \sum_{i\in N}\sum_{j\in N} \hat{p}_{ij}y_{ij}, \tag{35}$$

where

$$\hat{p}_{ij} = \begin{cases} p_{ij}+\lambda_{ij} & \text{if } i\neq j, \\ p_{ij} & \text{if } i=j \end{cases} \tag{36}$$

is the *Lagrangian profit* associated with variable y_{ij}. The corresponding Lagrangian relaxed problem is given by (35) subject to (30)–(32) and (34). For a given Λ, the continuous relaxation of this problem can be solved efficiently by solving the n continuous KP on the form (24) and one continuous KP on the form (2).

A well-known result in Lagrangian relaxation (see, e.g. [15]) states that the upper bound $U^2_{\text{CPT}} = z(\text{QKP}(\Lambda^*))$, where Λ^* is an optimal multiplier matrix, coincides with the optimal value of the LP relaxation (29)–(33). However, exact solution of this LP relaxation would be computationally very expensive due to the large number of variables and constraints involved, although one could add the constraints to the model gradually, as described in Billionnet and Calmels [5].

2.4.2. Reformulation

For each Λ such that $\hat{p}_{ji} \geqslant 0$ for $i, j \in N, j \neq i$, the corresponding Lagrangian profit matrix $\hat{P} = \{\hat{p}_{ij}\}$ defines a QKP instance which is equivalent to the initial one, i.e., we have a *reformulation* of the original problem. The use of problem reformulations for the *quadratic assignment problem* was proposed by a few authors, see the paper by Carraresi and Malucelli [10] for a unified analysis of the various approaches.

Caprara, Pisinger, Toth used the reformulation associated with the best upper bound obtained at the root node throughout the branch-and-bound algorithm. This results in a quite tight bound which can be derived very efficiently. Indeed, it can be derived in $O(n)$ expected time inside a branch-and-bound algorithm.

2.5. Michelon, Veilleux

Michelon and Veilleux [32] proposed a bound based on Lagrangian decomposition, by introducing some copy variables which are linked to the original variables through a number of equality constraints. Lagrangian relaxing the equality constraints, one is able to split the problem into a number of subproblems formulated in the disjoint sets of variables.

Michelon and Veilleux consider the formulation:

$$\begin{aligned} \text{maximize} \quad & \sum_{i\in N}\sum_{j\in N} p_{ij}x_i x_j \\ \text{subject to} \quad & \sum_{j\in N} w_j y_j \leqslant c, \\ & x_j = y_j, \quad j \in N, \\ & x_j,\ y_j \in \{0,1\}, \quad j \in N. \end{aligned} \tag{37}$$

Relaxing the constraints $x_j = y_j$ for $j \in N$ using multipliers $\lambda_j \in \mathbb{R}$ the problem $L(\text{QKP}, \Lambda)$ is obtained:

$$\begin{aligned} \text{maximize} \quad & \sum_{i\in N}\sum_{j\in N} p_{ij}x_i x_j - \sum_{j\in N} \lambda_j (x_j - y_j) \\ \text{subject to} \quad & \sum_{j\in N} w_j y_j \leqslant c, \\ & x_j, y_j \in \{0,1\}, \quad j \in N \end{aligned} \tag{38}$$

which may be decomposed into two subproblems

$$\begin{aligned} (\text{QP})\ \text{maximize} \quad & \sum_{i\in N}\sum_{j\in N} \tilde{p}_{ij}x_i x_j \\ \text{subject to} \quad & x_j, \in \{0,1\}, \quad j \in N \end{aligned} \tag{39}$$

and

$$\begin{aligned} \text{maximize} \quad & \sum_{i\in N} \lambda_j y_j \\ \text{subject to} \quad & \sum_{j\in N} w_j y_j \leqslant c, \\ & y_j \in \{0,1\}, \quad j \in N. \end{aligned} \tag{40}$$

The second subproblem (40) is an ordinary KP. In the first subproblem (39) the Lagrangian profits are $\tilde{p}_{ij} = p_{ij}$ if $i \neq j$ and $\tilde{p}_{ij} = p_{ij} - \lambda_j$ if $i = j$. Hence the problem can be solved in a similar way as the model $L^c(\text{QKP}, \lambda)$ described in Section 2.2. We construct a network $\mathcal{N} = (V, E, \overline{c})$ defined on the vertices $V = \{s, 1, \ldots, n, t\}$ where s is the source and t is the sink. The edge set E is defined on $V \times V$, having the capacities

$$\begin{aligned}
&\overline{c}_{si}(\lambda) = \max\left(0, \sum_{j\in N} p_{ij} - \lambda_i\right),\\
&\overline{c}_{ij}(\lambda) = p_{ij},\\
&\overline{c}_{it}(\lambda) = \max\left(0, \lambda_i - \sum_{j\in N} p_{ij}\right).
\end{aligned} \tag{41}$$

Assume that the value of the maximum flow from s to t in $\mathcal{N}$ is $\Psi(\mathcal{N})$. Then the solution value $z(\mathrm{QP})$ to (39) is given by

$$z(\mathrm{QP}) := \sum_{i\in N} c_{si} - \Psi(\mathcal{N}).$$

For a given vector Λ of Lagrangian multipliers, we get the bound $U^1_{\mathrm{MV}}(\Lambda) = L(\mathrm{QKP}, \Lambda)$. In order to get the tightest bound, the Lagrangian dual problem is solved

$$U^2_{\mathrm{MV}} = \min_{\Lambda\in\mathbb{R}^n} U^1_{\mathrm{MV}}(\Lambda). \tag{42}$$

Using subgradient optimization, one gets the bound $\hat{U}^2_{\mathrm{MV}}$ for a near-optimal matrix of Lagrangian multipliers $\hat{\Lambda}$. The bounds obviously satisfy $U^2_{\mathrm{MV}} \leqslant \hat{U}^2_{\mathrm{MV}} \leqslant U^1_{\mathrm{MV}}(0)$. Moreover, we have

Proposition 1. $U^2_{\mathrm{MV}} \leqslant U_{\mathrm{CHM}}$.

Proof. Let λ' be the optimal Lagrangian multiplier corresponding to the Lagrangian dual (12). We choose $\Lambda' = (w_1\lambda', w_2\lambda', \ldots, w_n\lambda')$. Then for any feasible solution x to $L^c(\mathrm{QKP}, \lambda')$ given by (8) all feasible solutions (x, y) to $L(\mathrm{QKP}, \Lambda')$ given by (38) will satisfy

$$\begin{aligned}
U_{\mathrm{CHM}} - U^1_{\mathrm{MV}}(\Lambda') &= \sum_{i\in N}\sum_{j\in N} p_{ij}x_i x_j - \lambda'\left(\sum_{j\in N} w_j x_j - c\right) - \sum_{i\in N}\sum_{j\in N} p_{ij}x_i x_j + \sum_{j\in N} w_j \lambda'_j (x_j - y_j)\\
&= \lambda'\left(c - \sum_{j\in N} w_j y_j\right) \geqslant 0,
\end{aligned} \tag{43}$$

where the last inequality holds due to constraint (38) and the fact that $\lambda' \geqslant 0$. □

If we further assume that the subgradient optimization for calculating $\hat{U}^2_{\mathrm{MV}}$ starts with $\Lambda = \Lambda'$ for the optimal λ' in U_{CHM} we also have the dominance $\hat{U}^2_{\mathrm{MV}} \leqslant U_{\mathrm{CHM}}$.

2.6. *Billionnet, Faye, Soutif*

The bound by Billionnet et al. [6] is based on a partitioning of N into m disjoint classes $\{I_1, \ldots, I_m\}$ satisfying $\cup_{k=1}^m I_k = N$. The main idea in the bound by Billionnet, Faye, Soutif is to use Lagrangian decomposition to split the problem into m independent subproblems. Each subproblem is solved by enumerating all decision variables in class I_k. For a fixed solution vector x_{I_k} the subproblem is an ordinary linear 0–1 KP which may be solved in $\mathrm{O}(n)$ time.

In order to partition the problem we notice that the objective function of QKP may be written as

$$\sum_{i\in N}\sum_{j\in N} p_{ij}x_ix_j = \sum_{k=1}^{m}\sum_{i\in I_k}\left(\sum_{j\in I_k} p_{ij}x_ix_j + \sum_{j\in N\setminus I_k} p_{ij}x_ix_j\right)$$
$$= \sum_{k=1}^{m}\left(\sum_{i\in I_k}\sum_{j\in I_k} p_{ij}x_ix_j + \sum_{i\in I_k}\sum_{j\in N\setminus I_k} p_{ij}x_ix_j\right). \tag{44}$$

Using Lagrangian decomposition we introduce binary copy variables $y_j^k = x_j$ for $j \in N\setminus I_k$ and add k redundant capacity constraints. Moreover, we add copy constraints concerning the quadratic terms $x_ix_j = x_iy_j^k$, getting the following formulation. The function $\mathrm{set}(i)$ returns the set index of the class to which item i belongs, i.e. the index k for which $i \in I_k$. Since the sets I_k are disjoint $\mathrm{set}(i)$ is well defined. Hence, we get the formulation

$$\begin{aligned}
\text{maximize}\quad & \sum_{k=1}^{m}\left(\sum_{i\in I_k}\sum_{j\in I_k} p_{ij}x_ix_j + \sum_{i\in I_k}\sum_{j\in N\setminus I_k} p_{ij}x_iy_j^k\right)\\
\text{subject to}\quad & x_j = y_j^k, \quad k=1,\dots,m,\ \ j\in N\setminus I_k,\\
& x_iy_j^{\mathrm{set}(i)} = x_jy_i^{\mathrm{set}(j)}, \quad i\in N,\ \ j\in N,\ \ \mathrm{set}(i)\neq\mathrm{set}(j),\\
& \sum_{i\in I_k} w_ix_i + \sum_{j\in N\setminus I_k} w_jy_j^k \leqslant c, \quad k=1,\dots,m,\\
& x_i\in\{0,1\}, \quad i\in I_k,\ \ k=1,\dots,m,\\
& y_j^k\in\{0,1\}, \quad k=1,\dots,m,\ \ j\in N\setminus I_k.
\end{aligned}\tag{45}$$

Lagrangian relaxing the two first constraints using multipliers $\Lambda=\{\lambda_j^k\}, k=1,\dots,m,\ j\in N\setminus I_k$, respectively, $M=\{\mu_{ij}\}$, $i\in N,\ j\in N$, $\mathrm{set}(i)\neq\mathrm{set}(j)$ we get

$$\begin{aligned}
\text{maximize}\quad & \sum_{k=1}^{m}\sum_{i\in I_k}\sum_{j\in I_k} p_{ij}x_ix_j + \sum_{k=1}^{m}\sum_{i\in I_k}\sum_{j\in N\setminus I_k} p_{ij}x_iy_j^k\\
& + \sum_{k=1}^{m}\sum_{j\in N\setminus I_k}\lambda_j^k(x_j-y_j^k) + \sum_{i\in N}\sum_{\substack{j\in N\\ \mathrm{set}(j)\neq\mathrm{set}(i)}}\mu_{ij}(x_iy_j^{\mathrm{set}(i)} - x_jy_i^{\mathrm{set}(j)})\\
\text{subject to}\quad & \sum_{j\in I_k} w_ix_i + \sum_{j\in N\setminus I_k} w_jy_j^k \leqslant c, \quad k=1,\dots,m,\\
& x_i\in\{0,1\}, \quad i\in I_k,\ \ k=1,\dots,m,\\
& y_j^k\in\{0,1\}, \quad k=1,\dots,m,\ \ j\in N\setminus I_k.
\end{aligned}\tag{46}$$

Since

$$\sum_{i\in N}\sum_{\substack{j\in N\\ \mathrm{set}(j)\neq\mathrm{set}(i)}}\mu_{ij}(x_iy_j^{\mathrm{set}(i)} - x_jy_i^{\mathrm{set}(j)}) = \sum_{k=1}^{m}\sum_{i\in I_k}\sum_{j\in N\setminus I_k}\mu_{ij}x_iy_j^k - \sum_{k=1}^{m}\sum_{i\in I_k}\sum_{j\in N\setminus I_k}\mu_{ji}x_iy_j^k$$

and

$$\sum_{k=1}^{m}\sum_{j\in N\setminus I_k}\lambda_j^kx_j = \sum_{i\in N}\left(\sum_{h\neq\mathrm{set}(i)}\lambda_i^h\right)x_i = \sum_{k=1}^{m}\sum_{i\in I_k}\left(\sum_{h\neq k}\lambda_i^h\right)x_i$$

we can split (46) into m independent problems $L_k(I_k, \Lambda, M)$, $k = 1, \ldots, m$ of the form

$$\begin{aligned}
\text{maximize} \quad & \sum_{i\in I_k}\sum_{j\in I_k} p_{ij}x_i x_j + \sum_{i\in I_k}\left(\sum_{h\neq k}\lambda_i^h\right)x_i \\
& + \sum_{j\in N\setminus I_k}\left(\sum_{i\in I_k} p_{ij}x_i\right)y_j^k - \sum_{j\in N\setminus I_k}\lambda_j^k y_j^k + \sum_{j\in N\setminus I_k}\left(\sum_{i\in I_k}(\mu_{ij}-\mu_{ji})x_i\right)y_j^k \\
\text{subject to} \quad & \sum_{i\in I_k} w_i x_i + \sum_{j\in N\setminus I_k} w_j y_j^k \leqslant c, \\
& x_i \in \{0,1\}, \quad i \in I_k, \quad k = 1, \ldots, m, \\
& y_j^k \in \{0,1\}, \quad j \in N\setminus I_k.
\end{aligned} \tag{47}$$

Assuming that the sets I_k are small we may enumerate all solutions of I_k in problem $L_k(I_k, \Lambda, M)$. For a fixed value of x_i, $i \in I_k$ the problem can be recognized as a 0–1 KP defined in the variables y_j^k by setting

$$\tilde{p}_j = \sum_{i\in I_k} p_{ij}x_i - \lambda_j^k + \sum_{i\in I_k}(\mu_{ij}-\mu_{ji})x_i$$

hence getting $L_k(I_k, \Lambda, M)$ in the form

$$\begin{aligned}
\text{maximize} \quad & \text{const} + \sum_{j\in N\setminus I_k} \tilde{p}_j y_j^k \\
\text{subject to} \quad & \sum_{j\in N\setminus I_k} w_j y_j^k \leqslant c - \sum_{i\in I_k} w_i x_i, \\
& y_j^k \in \{0,1\}, \quad j \in N.
\end{aligned} \tag{48}$$

The objective function may hence be written $U^1_{\mathrm{BFS}}(\Lambda, M) = \sum_{k=1}^m L_k(I_k, \Lambda, M)$. In order to make the computation of $U^1_{\mathrm{BFS}}(\Lambda, M)$ faster we solve the continuous relaxation of (48) which can be done in $\mathrm{O}(n)$ time. The tightest bound U^2_{BFS} is now found as a solution to the Lagrangian dual problem

$$U^2_{\mathrm{BFS}} = \min_{\Lambda, M} U^1_{\mathrm{BFS}}(\Lambda, M). \tag{49}$$

A sub-optimal choice of Lagrangian multipliers Λ and M can be found through subgradient optimization, leading to the bound $\hat{U}^2_{\mathrm{BFS}}$ with $U^2_{\mathrm{BFS}} \leqslant \hat{U}^2_{\mathrm{BFS}} \leqslant U^1_{\mathrm{BFS}}(0, 0)$.

The time complexity of the bound for given values of Λ, M is derived as follows. If we assume that the set N was partitioned into m sets I_k of equal size n/m then the time consumption for solving problem $L_k(I_k, \Lambda, M)$ for a given choice of λ, μ is $\mathrm{O}(2^{n/m}n)$. As we have to solve m subproblems the total time consumption is $\mathrm{O}(2^{n/m}mn)$. Billionnet et al. [6] notice that the larger size n/m the better bounds—however, at the cost of an exponentially increasing time consumption. The time complexity should be multiplied by the number of iterations used for iterating the Lagrangian multipliers in the bound $\hat{U}^2_{\mathrm{BFS}}$. In their computational experiments, Billionnet, Faye, Soutif used relatively small sets of size $n/m = 5$ to keep the computational times at a reasonable level.

2.7. Helmberg, Rendl, Weismantel

Helmberg et al. [27,28] and Helmberg [26] proposed a number of upper bounds for QKP based on semidefinite programming. These bounds are valid for a more general version of the problem where the profit matrix P may contain negative entries. For reasons of completeness, the bounds are presented in the sequel, although it is outside the scope of the present survey to give a deeper introduction to semidefinite programming. Instead we refer to e.g. Wolkowicz et al. [47].

First we need some notation. We will write the solution variables and the weights as column vectors $x = (x_1, \ldots, x_n)^{\mathrm{T}}$, $w = (w_1, \ldots, w_n)^{\mathrm{T}}$. The set of real $m \times n$ matrices is denoted $M_{m,n}$. The *inner product* between matrices $A, B \in M_{m,n}$ is defined as $\langle A, B\rangle = \sum_{i=1}^m \sum_{j=1}^n a_{ij}b_{ij}$.

The vector of *diagonal elements* of a square matrix $A \in M_{n,n}$ will be denoted $\mathrm{diag}(A)$ and is defined as $\mathrm{diag}(A) = (a_{11}, \ldots, a_{nn})^{\mathrm{T}}$. The somehow reverse operation $\mathrm{Diag}(a)$ takes a vector $a \in \mathbb{R}^n$ and converts it into a *diagonal matrix*.

$$\mathrm{Diag}(a) = \mathrm{Diag}(a_1, \ldots, a_n) = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & a_n \end{pmatrix}. \tag{50}$$

The *rank* of a matrix A will be written $\mathrm{rank}(A)$. It is defined as the number of linearly independent columns of A. A matrix $A \in M_{n,n}$ is said to be *positive semidefinite* if for all vectors $y \in \mathbb{R}^n$ we have

$$y^{\mathrm{T}} A y \geqslant 0. \tag{51}$$

We will use the notation $A \succcurlyeq 0$ to indicate that A is symmetric and positive semidefinite. Notice that

$$X_1 \succcurlyeq 0, X_2 \succcurlyeq 0, \ldots, X_k \succcurlyeq 0 \quad \Leftrightarrow \quad \mathrm{Diag}(X_1, X_2, \ldots, X_k) \succcurlyeq 0 \tag{52}$$

meaning that we may express a problem defined in several semidefinite matrices as a single semidefinite matrix.

Grötschel et al. [22] proved that a semidefinite optimization problem can be solved in polynomial time measured in the input size and accuracy, hence making *semidefinite relaxations* an attractive tool for deriving upper bounds in combinatorial optimization.

In order to formulate a number of bounds for the QKP we will use the following proposition

Proposition 2. *The following two properties are equivalent*

(1) $X \succcurlyeq 0$ *and* $\mathrm{rank}(X) = 1$.
(2) $X = xx^{\mathrm{T}}$ *for some vector* $x \in \mathbb{R}^n$.

The consequence of the above proposition is that we may write the objective function of QKP as

$$x^{\mathrm{T}} P x = \sum_{i \in N} \sum_{j \in N} p_{ij} x_i x_j = \langle P, X \rangle, \tag{53}$$

where $X = xx^{\mathrm{T}}$. This leads to the following formulation of QKP:

$$\begin{array}{ll} \text{maximize} & \langle P, X \rangle \\ \text{subject to} & \langle \mathrm{Diag}(w), X \rangle \leqslant c, \\ & X \succcurlyeq 0, \\ & \mathrm{rank}(X) = 1, \\ & X_{ii} \in \{0, 1\}. \end{array} \tag{54}$$

By dropping the $\mathrm{rank}(X) = 1$ constraint, and relaxing the last constraint to $0 \leqslant X_{ii} \leqslant 1$ we get a semidefinite relaxation which gives us an upper bound U^0_{HRW} on QKP. According to Helmberg et al. [28] this bound is of poor quality.

To reach tighter bounds, we observe that we may reformulate the last three constraints in formulation (54).

Proposition 3. *If* $X \succcurlyeq 0$ *and* $\mathrm{rank}(X) = 1$ *and* $X_{ii} \in \{0, 1\}$ *then also*

$$X - \mathrm{diag}(X)\mathrm{diag}(X)^{\mathrm{T}} \succcurlyeq 0. \tag{55}$$

Proof. Due to Proposition 2 we can write $X = xx^{\mathrm{T}}$. For any vector $v \in \mathbb{R}^n$ we have that the matrix $(x + v)(x + v)^{\mathrm{T}} \succcurlyeq 0$ hence by multiplication

$$xx^{\mathrm{T}} + vx^{\mathrm{T}} + xv^{\mathrm{T}} + vv^{\mathrm{T}} \succcurlyeq 0 \quad \forall v \in \mathbb{R}^n.$$

Using the fact that $\operatorname{diag}(xx^{\mathrm{T}}) = x$ when $x \in \{0, 1\}^n$ we get

$$X + v\operatorname{diag}(X)^{\mathrm{T}} + \operatorname{diag}(X)v^{\mathrm{T}} + vv^{\mathrm{T}} \succcurlyeq 0 \quad \forall v \in \mathbb{R}^n \quad \Leftrightarrow$$
$$X + (v + \operatorname{diag}(X))(v + \operatorname{diag}(X))^{\mathrm{T}} - \operatorname{diag}(X)\operatorname{diag}(X)^{\mathrm{T}} \succcurlyeq 0 \quad \forall v \in \mathbb{R}^n$$

choosing $v = -\operatorname{diag}(X)$ we get the stated. $\square$

To express (55) as a semidefinite optimization problem we note that

Proposition 4. $X - \operatorname{diag}(X)\operatorname{diag}(X)^{\mathrm{T}} \succcurlyeq 0$ *if and only if* $\overline{X} \succcurlyeq 0$, *where*

$$\overline{X} = \begin{pmatrix} 1 & \operatorname{diag}(X)^{\mathrm{T}} \\ \operatorname{diag}(X) & X \end{pmatrix}. \tag{56}$$

Proof. Define the regular matrix B as

$$B = \begin{pmatrix} 1 & -\operatorname{diag}(X)^{\mathrm{T}} \\ 0 & I \end{pmatrix}$$

and observe that

$$B^{\mathrm{T}}\overline{X}B = \begin{pmatrix} 1 & 0 \\ -\operatorname{diag}(X) & I \end{pmatrix} \begin{pmatrix} 1 & \operatorname{diag}(X)^{\mathrm{T}} \\ \operatorname{diag}(X) & X \end{pmatrix} \begin{pmatrix} 1 & -\operatorname{diag}(X)^{\mathrm{T}} \\ 0 & I \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 \\ 0 & X - \operatorname{diag}(X)\operatorname{diag}(X)^{\mathrm{T}} \end{pmatrix}.$$

Due to the fact that

$$y^{\mathrm{T}}\overline{X}y = y^{\mathrm{T}}B^{-\mathrm{T}}B^{\mathrm{T}}\overline{X}BB^{-1}y = \overline{y}^{\mathrm{T}}B^{\mathrm{T}}\overline{X}B\overline{y} \tag{57}$$

we have that $\overline{X}$ satisfies property (51) for a given y if and only if $B^{\mathrm{T}}\overline{X}B$ satisfies the property with $\overline{y} = B^{-1}y$. Hence $\overline{X} \succcurlyeq 0$ if and only if $B^{\mathrm{T}}\overline{X}B \succcurlyeq 0$. Observation (52) now gives the stated. $\square$

Based on the two above propositions, we may relax QKP to QKP′ on the form

$$\begin{aligned} \text{maximize} \quad & \langle P, X\rangle \\ \text{subject to} \quad & \langle \operatorname{Diag}(w), X\rangle \leqslant c, \\ & X - \operatorname{diag}(X)\operatorname{diag}(X)^{\mathrm{T}} \succcurlyeq 0, \\ & \operatorname{rank}(X) = 1, \\ & X_{ii} \in \{0, 1\}. \end{aligned} \tag{58}$$

The bound U^1_{HRW} is obtained by dropping the $\operatorname{rank}(X) = 1$ constraint and relaxing the last constraint to $0 \leqslant X_{ii} \leqslant 1$ getting the model

$$\begin{aligned} \text{maximize} \quad & \langle P, X\rangle \\ \text{subject to} \quad & \langle \operatorname{Diag}(w), X\rangle \leqslant c, \\ & X - \operatorname{diag}(X)\operatorname{diag}(X)^{\mathrm{T}} \succcurlyeq 0, \\ & X_{ii} \leqslant 1. \end{aligned} \tag{59}$$

The next bound U^2_{HRW} is based on the fact that $w^{\mathrm{T}}x = x^{\mathrm{T}}w$, hence $w^{\mathrm{T}}x \leqslant c$ implies that $w^{\mathrm{T}}xx^{\mathrm{T}}w \leqslant c^2$. By rewriting $w^{\mathrm{T}}xx^{\mathrm{T}}w$ as $\langle ww^{\mathrm{T}}, xx^{\mathrm{T}}\rangle$ and relaxing xx^{T} to X we get the relaxation

$$\begin{aligned} \text{maximize} \quad & \langle P, X\rangle \\ \text{subject to} \quad & \langle ww^{\mathrm{T}}, X\rangle \leqslant c^2, \\ & X - \operatorname{diag}(X)\operatorname{diag}(X)^{\mathrm{T}} \succcurlyeq 0. \end{aligned}$$

The third semidefinite relaxation is based on the observation that $w^{\mathrm{T}}x \leqslant c$ can be multiplied by the real number $w^{\mathrm{T}}x$ on both sides gives the constraint $(w^{\mathrm{T}}x)^2 \leqslant w^{\mathrm{T}}xc$, leading to the inequality

$$0 \leqslant w^{\mathrm{T}}x(c - x^{\mathrm{T}}w) = w^{\mathrm{T}}x \begin{pmatrix} 1 & x^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} c \\ -w \end{pmatrix} = \begin{pmatrix} 0 & w^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 & x^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} c \\ -w \end{pmatrix}. \tag{60}$$

Setting $X' = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 & x^{\mathrm{T}} \end{pmatrix} = x'x'^{\mathrm{T}}$ the right expression in (60) can be written

$$\left\langle \begin{pmatrix} c \\ -w \end{pmatrix} \begin{pmatrix} 0 & w^{\mathrm{T}} \end{pmatrix}, X' \right\rangle. \tag{61}$$

This leads to the following relaxation

$$\begin{aligned} \text{maximize} \quad & \langle P, X \rangle \\ \text{subject to} \quad & \left\langle \begin{pmatrix} c \\ -w \end{pmatrix} \begin{pmatrix} 0 & w^{\mathrm{T}} \end{pmatrix}, X' \right\rangle \geqslant 0, \\ & X - \operatorname{diag}(X) \operatorname{diag}(X)^{\mathrm{T}} \succcurlyeq 0. \end{aligned} \tag{62}$$

Solving the above problem gives bound U^3_{HRW}.

The last relaxation is obtained by multiplying the capacity constraint with each of the variables x_i for $i \in N$ getting $x_i w^{\mathrm{T}} x \leqslant x_i c$. By writing the vector product explicitly we get the sum

$$\sum_{j \in N} w_j x_i x_j \leqslant x_i c. \tag{63}$$

By introducing X_{ij} for $x_i x_j$ and X_{ii} for x_i we get

$$\sum_{j \in N} w_j X_{ij} \leqslant X_{ii} c \tag{64}$$

which leads to the following relaxation

$$\begin{aligned} \text{maximize} \quad & \langle P, X \rangle \\ \text{subject to} \quad & \sum_{j \in N} w_j X_{ij} - X_{ii} c \leqslant 0 \quad \text{for } i \in N, \\ & X - \operatorname{diag}(X) \operatorname{diag}(X)^{\mathrm{T}} \succcurlyeq 0 \end{aligned} \tag{65}$$

getting the bound U^4_{HRW}.

Helmberg et al. [28] together with Bauvin and Goemans [4] prove that $U^1_{\mathrm{HRW}} \geqslant U^2_{\mathrm{HRW}} \geqslant U^3_{\mathrm{HRW}} \geqslant U^4_{\mathrm{HRW}}$ hence showing that the formulation (65) is to be preferred.

Proposition 5 (*Helmberg et al. [28], Bauvin and Goemans [4]*). *Let $\mathscr{X}_1, \mathscr{X}_2, \mathscr{X}_3, \mathscr{X}_4$ be the feasible set corresponding to bounds $U^1_{\mathrm{HRW}}, U^2_{\mathrm{HRW}}, U^3_{\mathrm{HRW}}$ and U^4_{HRW}. Then we have the relation $\mathscr{X}_1 \supseteq \mathscr{X}_2 \supseteq \mathscr{X}_3 \supseteq \mathscr{X}_4$.*

Proof. To prove that $\mathscr{X}_1 \supseteq \mathscr{X}_2$ assume that $X \in \mathscr{X}_2$. Introducing the positive semidefinite matrix $Z = X - \operatorname{diag}(X) \operatorname{diag}(X)^{\mathrm{T}}$ we have

$$c^2 \geqslant w^{\mathrm{T}} X w = w Z w + (w^{\mathrm{T}} \operatorname{diag}(X))^2. \tag{66}$$

Since $Z \succcurlyeq 0$ by the feasibility of X it follows that $(w^{\mathrm{T}} \operatorname{diag}(X))^2 \leqslant c^2$, which proves $X \in \mathscr{X}_1$.

Next, let $X \in \mathscr{X}_3$. Using the same matrix $Z = X - \operatorname{diag}(X) \operatorname{diag}(X)^{\mathrm{T}} \succcurlyeq 0$ we see that

$$\begin{aligned} 0 \leqslant & \left\langle \begin{pmatrix} c \\ -w \end{pmatrix} \begin{pmatrix} 0 & w^{\mathrm{T}} \end{pmatrix}, X' \right\rangle \\ & = c w^{\mathrm{T}} \operatorname{diag}(X) - w^{\mathrm{T}} X w \\ & = c w^{\mathrm{T}} \operatorname{diag}(X) - w^{\mathrm{T}} Z w - w^{\mathrm{T}} \operatorname{diag}(X) \operatorname{diag}(X)^{\mathrm{T}} w \\ & = (c - w^{\mathrm{T}} \operatorname{diag}(X)) w^{\mathrm{T}} \operatorname{diag}(X) - w^{\mathrm{T}} Z w. \end{aligned} \tag{67}$$

Since $-w^{\mathrm{T}}Zw \leqslant 0$ we have $c \geqslant w^{\mathrm{T}}\mathrm{diag}(X)$ and hence $c^2 \geqslant w^{\mathrm{T}}Xw$ in the first row of the equation. This shows that $X \in \mathscr{X}_2$.

Finally, assume that $X \in \mathscr{X}_4$, and multiply each x_i representation of (65) by $w_i \geqslant 0$. Summing all these inequalities over i we get

$$\sum_{i\in N}\sum_{j\in N} w_i w_j x_{ij} = \langle ww^{\mathrm{T}}, X\rangle \leqslant \sum_{i\in N} c w_i x_{ii} \tag{68}$$

which gives exactly the inequality of (62) hence $X \in \mathscr{X}_4$. □

3. Variable reduction

The size of a QKP instance may be considerably reduced by using some reduction rules from the classical KP as described in e.g. Martello and Toth [31] and Pisinger and Toth [39]. Assume that an incumbent solution of value z has been determined by some initial heuristic. Let U_j^1 be an upper bound on the QKP obtained by imposing the additional constraint $x_j = 1$. If $U_j^1 \leqslant z$ then we can fix x_j at 0. Similarly, if U_j^0 is an upper bound on the QKP obtained by imposing the additional constraint $x_j = 0$ and $U_j^0 \leqslant z$ we can fix x_j at 1. Whenever a variable x_j is fixed at some value we remove the corresponding row and column in P. Moreover, if it is fixed at 1, we also increase diagonal entry p_{jj} by $p_{ij} + p_{ji}$, for $i \in N\backslash\{j\}$, and decrease c by w_j.

Caprara et al. [9] used the bound U_{CPT}^1 which can be determined in $\mathrm{O}(n^2)$ time for each j by solving the Lagrangian relaxed problem for a fixed set of Λ values corresponding to the solution of (27) at the root node. If the reduction procedure fixes at least one variable, the subgradient optimization may be applied to the now reduced problem, followed by a new reduction. In worst case this approach runs in $\mathrm{O}(n^4)$ although in practice a very limited set of iterations are needed of the latter part.

Hammer and Rader [24] used the bound U_{CHM} based on Lagrangian relaxation of the capacity constraint as described in Section 2.2. In addition, they used some order relations which may be used to fix variables at their proper value inside a branch-and-bound algorithm: assume that two items i, j satisfy that $w_i \geqslant w_j$, and consider the so-called “second order derivative” $\Delta_{ij} = f(x|_{x_i=1,x_j=0}) - f(x|_{x_i=0,x_j=1})$ where $f(x) = \sum_{i\in N}\sum_{j\in N} p_{ij}x_i x_j$. If $\Delta_{ij} \leqslant 0$ at some optimal solution x then there exists an optimal solution x^* so that $x_i^* \leqslant x_j^*$. Hence, whenever we branch at $x_i^* = 1$ we may immediately fix $x_j^* = 1$ and if we branch at $x_j^* = 0$ we may fix $x_i^* = 0$. See [24] for additional details.

4. Heuristics

We may roughly divide heuristics for QKP into two classes: the *primal*, which maintain feasibility throughout the construction, and the *dual* which start from an infeasible solution and strive towards a feasible solution.

Gallo et al. [18] presented a family of primal heuristics corresponding to the bounds based on upper planes. Solving (2) immediately gives a feasible solution to QKP. If the continuous relaxation of (2) is solved, one may obtain a feasible solution by truncating the fractional variables. Gallo, Hammer Simeone proposed to further improve a feasible solution, through a sequence of *fill-up* and *exchange* operations as proposed by Peterson [37]. When choosing the items to exchange, second order “*derivatives*” are used.

Hammer and Rader [24] presented a different primal heuristic, named *LEX*, based on the *best linear L_2 approximation* of QKP as presented in Hammer and Holzman [23]. The best linear approximation is

$$\begin{aligned}
\text{maximize} \quad & \sum_{j\in N} \pi_j^1 x_j \\
\text{subject to} \quad & \sum_{j\in N} w_j x_j \leqslant c, \\
& x_j \in \{0,1\}, \quad j \in N.
\end{aligned} \tag{69}$$

where π^1 is the upper plane defined by (3). In each step of the LEX algorithm, the item with the highest *efficiency* π_j^1/w_j is selected and the corresponding solution variable x_j is assigned the value 1. All items, which no longer fit into the residual capacity of the knapsack, are removed from the problem and their solution variables x_j are set to 0. The process is repeated until no items fit into the knapsack. In the last phase of the algorithm, local improvements are performed, by either exchanging items, or filling up with new items that fit into the current residual capacity. When choosing the items to exchange or include, first and second order "*derivatives*" are used, using the framework of Peterson [37].

A well-performing dual heuristic was presented by Billionnet and Calmels [5]. This algorithm first generates a greedy solution by initially setting $x_j = 1$ for $j \in N$, and then iteratively setting the value of a variable from 1 to 0, so as to achieve the smallest loss in the objective value, until a feasible solution is obtained. In the second step a sequence of iterations is performed in order to improve the solution by local exchanges. Let $S = \{j \in N : x_j = 1\}$ be the set of the items selected in the current solution. For each $j \in N \backslash S$, if $w_j + \sum_{\ell \in S} w_\ell \leqslant c$ set $I_j = \emptyset$ and let the quantity δ_j be the objective function increase when x_j is set to 1. Otherwise, let δ_j be the largest profit increase when setting $x_j = 1$ and $x_i = 0$ for some $i \in S$ such that $w_j - w_i + \sum_{\ell \in S} w_\ell \leqslant c$, and let $I_j = \{i\}$. Choosing k such that $\delta_k = \max_{j \in N \backslash S} \delta_j$, the heuristic algorithm terminates if $\delta_k \leqslant 0$, otherwise the current solution is set to $S \backslash I_k \cup \{k\}$ and another iteration is performed.

Caprara et al. [9] improved this bound by building it into the subgradient algorithm used for deriving bound $\hat{U}^2_{\mathrm{CPT}}$ as described in Section 2.4. The heuristic by Billionnet and Calmels is used at the first step of the algorithm, while at each iteration of the subgradient optimization procedure a heuristic solution is derived as follows. The LP solution of (35) subject to (30)–(32) is rounded down, yielding an integer solution x. Starting from x the improvement part of the above algorithm is performed. The solutions obtained this way are typically substantially different from each other, even for slightly different Lagrangian profits, showing that the heuristic algorithm is worth applying often during the subgradient procedure.

Glover and Kochenberger [21] presented a tabu search method for solving QKP, based on a reformulation scheme.

5. Approximation algorithms

Since QKP is strongly $\mathcal{NP}$-hard we cannot expect to find a fully polynomial approximation scheme unless $\mathcal{NP} = \mathcal{P}$. However, Rader and Woeginger [40] developed a FPTAS for the special case where all profits $p_{ij} \geqslant 0$ and where the underlying graph $G = (V, E)$ is so-called *edge series parallel*. The result relies on the fact that it is quite easy to develop a dynamic programming algorithm for the present case. Rader and Woeginger also prove that if the underlying graph $G = (V, E)$ is so-called *vertex series parallel*, then the problem is strongly $\mathcal{NP}$-hard, and hence we cannot expect to find a FPTAS. The latter proof is based on reduction from the *balanced complete bipartite subgraph* problem (problem GT24 in [19]). Moreover, they prove the following negative result:

Proposition 6 (*Rader and Woeginger [40]*). *The QKP with positive and negative profits p_{ij} does not have any polynomial time approximation algorithm with fixed approximation ratio unless $\mathcal{P} = \mathcal{NP}$.*

Proof. The stated is proved by reduction from SSP-DECISION, which for a given set of nonnegative integer weights $w'_1, \ldots, w'_n$ and a capacity c' asks whether a subset of the weights can be chosen such that their total weight equals c'. For a given instance of SSP-DECISION we construct an instance of QKP with $n + 1$ items by choosing $w_0 = 0$, $w_i := w'_i$ and $c := c'$. The profits are $p_{00} = -c' + 1$ and $p_{0j} := w_j$ with all other profits set to 0.

A feasible solution to this QKP may either chose $x_0 = 0$ in which case the optimal solution value is $z^* = 0$. Otherwise, if $x_0 = 1$ then the solution value cannot exceed $z^* = -c + 1 + \sum_{j \in N} w_j = 1$ due to the capacity constraint $\sum_{j \in N} w_j \leqslant c$. The solution value $z^* = 1$ is attained if and only if SSP-DECISION has a feasible solution.

Now, assuming that an approximation algorithm with ratio bound ρ did exist for QKP, we could use the algorithm to decide SSP-DECISION by solving the corresponding QKP and observing whether the approximate solution z is strictly positive. $\square$

Notice that the proof is based on the assumption that the profits may take on negative values. For the considered case (1) where profits are nonnegative—to the best knowledge of the author—it is unknown whether QKP can be approximated with a constant approximation factor [13].

6. Valid inequalities

Several valid inequalities can be derived for the QKP. The simplest inequalities are derived from the classical KP polyhedron, while additional inequalities can be derived by considering the QKP or the QP polyhedron. The inequalities presented in this section will be given in their basic form, but standard lifting techniques as presented by Padberg [33] can be used.

All the following inequalities can in principle be used to tighten the bounds presented in Section 2, although it is most easy to add the inequalities in the bounds based on the solution of a linear or semidefinite optimization model [5,27]. Where the upper bound is based on decomposition to some specific subproblems, the valid inequalities will typically be transferred to the formulation of the subproblems. This may significantly change the structure and complexity of the subproblems, making the decomposition less attractive.

The 0–1 knapsack polyhedron is given by

$$\mathscr{K} = \operatorname{conv}\left\{ x \in \{0,1\}^n : \sum_{j \in N} w_j x_j \leqslant c \right\}. \tag{70}$$

The simplest valid inequalities for $\mathscr{K}$ are the *cardinality constraints* proposed by Balas [2]. For any set $S \subseteq N$ satisfying $\sum_{j \in S} w_j > c$ the inequality

$$\sum_{j \in S} x_j \leqslant |S| - 1 \tag{71}$$

is valid for $\mathscr{K}$. Notice that we used this property in Eq. (5).

Weight inequalities were proposed by Weismantel [45]. Assume that the subset $T \subseteq N$ satisfies $\sum_{j \in T} w_j < c$ and define the residual capacity as $r = c - \sum_{j \in T} w_j$. The *weight inequality* with respect to T is defined by

$$\sum_{j \in T} w_j x_j + \sum_{j \in N \setminus T} \max\{0, w_j - r\} x_j \leqslant \sum_{j \in T} w_j. \tag{72}$$

Weismantel [45] proved that the inequality is valid for $\mathscr{K}$.

Extended weight inequalities are defined as follows. Let $T_1 \subseteq N$ and $T_2 \subseteq N$ be two disjoint subsets satisfying $\sum_{j \in T_1 \cup T_2} w_j \leqslant c$, and $w_i \leqslant w_j$ for all $i \in T_1$ and $j \in T_2$, and satisfying $\sum_{i \in T_1} w_i \geqslant w_j$ for all $j \in T_2$. Define the relative weight of an item $k \in T_1 \cup T_2$ as

$$\overline{w}_k = 1 \quad \text{if } k \in T_1, \tag{73}$$

$$\overline{w}_k = \min\left\{ |S| : S \subseteq T_1, \sum_{j \in S} w_j \geqslant w_k \right\} \quad \text{if } k \in T_2. \tag{74}$$

Under these assumptions we define for an item $h \in N \setminus (T_1 \cup T_2)$ the *extended weight inequality*

$$\sum_{j \in T_1} x_j + \sum_{j \in T_2} \overline{w}_j x_j + \overline{w}_h x_h \leqslant |T_1| + \sum_{j \in T_2} \overline{w}_j, \tag{75}$$

where $\overline{w}_h = \min\{|S_1| + \sum_{j \in S_2} \overline{w}_j : S_1 \subseteq T_1, S_2 \subseteq T_2, \sum_{j \in S_1 \cup S_2} w_j \geqslant w_h - r\}$ and $r = c - \sum_{j \in T_1 \cup T_2} w_j$. Weismantel [45] proved that (75) is valid for $\mathscr{K}$ and that lifting coefficients can always be computed in polynomial time.

By lifting the KP polyhedron (70) to the space of quadratic variables we get the following QKP polyhedron

$$\mathscr{Q} = \operatorname{conv}\left\{ y \in \{0,1\}^{n(n+1)/2} : \sum_{j \in N} w_j y_{jj} \leqslant c,\ y_{ij} = y_{ii} y_{jj},\ \text{for all } i < j \right\}. \tag{76}$$

Helmberg et al. [27] presented a framework for deriving valid inequalities for $\mathscr{Q}$. Let $N_1, \ldots, N_m$ be a partitioning of N into m subsets as described in Section 2.6. For every subset N_k choose a spanning tree T_k in the complete graph defined on the node set N_k. Let $\deg_j^k$ denote the degree of node j in the tree T_k. The polyhedron

$$\mathscr{Q}' = \operatorname{conv}\left\{ y \in \{0,1\}^{n(n-1)/2} : \sum_{k=1}^{m} \left(\sum_{j \in N_k} w_j \right) \left(\sum_{(i,j) \in T_k} y_{ij} + \sum_{j \in N_k} (1 - \deg_j^k) y_{jj} \right) \leqslant c \right\} \tag{77}$$

contains all the feasible points of $\mathscr{Q}$.

The observation may be used to derive valid inequalities for $\mathcal{Q}$ by first choosing a partitioning of $N_1, \ldots, N_m$ of N and some corresponding spanning trees $T_1, \ldots, T_m$. The polyhedron $\mathcal{Q}'$ is an ordinary 0–1 knapsack polyhedron hence we may easily derive cover inequalities, weight inequalities or extended weight inequalities. Since $\mathcal{Q} \subseteq \mathcal{Q}'$ all valid inequalities for $\mathcal{Q}'$ are also valid for $\mathcal{Q}$.

Johnson et al. [29] consider the graph version of the QKP. For a given graph $G=(V,E)$, the problem may be formulated as (14), where $V=N$ and $E \subseteq N \times N$ and the variables are set to $x_i=1$ when node i is chosen, and $y_{ij}=1$ when edge (i,j) is chosen. The corresponding polyhedron is given by

$$\mathscr{P}(G) = \operatorname{conv}\left\{ x \in \{0,1\}^n,\ y \in \mathbb{R}^{n\times n} : \sum_{j\in N} w_j x_j \leqslant c,\ y_{ij} \leqslant x_i,\ y_{ij} \leqslant x_j,\ y_{ij} \geqslant 0 \right\}. \tag{78}$$

A subset $S \subseteq V$ is said to be an *independent set* if $\sum_{j\in S} w_j \leqslant c$; otherwise S is a *dependent set*.

Let S be a dependent set, and T be a set of edges that form a spanning tree on the nodes in S. Let $\delta(i)=\{j : (i,j) \in T\}$ for each $i \in S$. Then the following *tree inequality* is valid for $\mathscr{P}(G)$,

$$\sum_{(i,j)\in T} y_{ij} \leqslant \sum_{i\in C} (|\delta(i)|-1)x_i. \tag{79}$$

Let G' be the graph induced by S. Then the tree inequality (79) defines a facet for $\mathscr{P}(G')$ if and only if $C\backslash\{i\}$ is an independent set for every leaf of the three induced by T.

If S is a *minimal* dependent set (i.e. all subsets of S are independant), and T forms a star on S, then the following *star inequality* is valid for $\mathscr{P}(G)$,

$$\sum_{j\in S\backslash\{i\}} y_{ij} \leqslant (|S|-2)x_i, \quad i \in C. \tag{80}$$

The inequality is facet defining for $\mathscr{P}(G')$.

The two inequalities (79) and (80) are easily generalized to forests (see [29] for details).

The polyhedron corresponding to the QP is defined as

$$\mathscr{R} = \operatorname{conv}\{y \in \{0,1\}^{n(n+1)/2} : y_{ij} = y_{ii}y_{jj},\ i<j\}. \tag{81}$$

Since QKP is a constrained variant of QP all valid inequalities for $\mathscr{R}$ are also valid for $\mathcal{Q}$. Padberg [34] presented the following inequalities for $\mathscr{R}$ which are valid for all possible values of $i, j, k \in N$,

$$y_{ij} \geqslant 0, \tag{82}$$
$$y_{ij} \leqslant y_{ii}, \tag{83}$$
$$y_{ii} + y_{jj} \leqslant 1 + y_{ij}, \tag{84}$$
$$y_{ik} + y_{jk} \leqslant y_{kk} + y_{ij}, \tag{85}$$
$$y_{ij} + y_{ik} + y_{jk} + 1 \geqslant y_{ii} + y_{jj} + y_{kk}. \tag{86}$$

These inequalities can be recognized as the *triangle inequalities* of the max-cut polytope as presented in De Simone [43]. Helmberg et al. [27] argue that these triangle inequalities contribute significantly to the tightness of the bound obtained through branch-and-cut. Indeed their scheme used more triangle inequalities than knapsack specific inequalities.

7. Branch-and-bound algorithms

Several branch-and-bound algorithms for QKP have been presented in the literature, the main distinction being the upper bounds used.

Gallo et al. [18] developed the first branch-and-bound algorithm using the bounds U^1_{GHS}–U^4_{GHS} based on upper planes. The branching variable was selected by solving (2) and letting F be the set of variables for which $x_j=1$ in the present solution. Branching is then performed on the variable $j \in F$ which maximizes π_j/w_j.

Chaillou et al. [12] as well as Hammer and Rader [24] used the bound U_{CHM} from Lagrangian relaxation of the capacity constraint. The branching strategy by Chaillou, Hansen, Mahieu was based on choosing the variable which results in the smallest reduction in the bound when changing the variable to its complement. In this derivation of upper bounds an approximation of U_{CHM} is used based on the same value of λ, as found during the iteration of (12). Hammer and Rader followed the strategy to do as much analysis as possible at each branching node as possible in order to get a limited search tree. Hence, a three-step procedure was developed, making use of constraint pairing, fixation of variables by Lagrangian techniques, and order relations (see Section 3). Branching is performed on the variable which makes it possible to fix most variables at their optimal value.

Billionnet and Calmels [5] applied the bound U^2_{BC} using a branch-and-cut approach for gradually writing up the model. When deciding which variable to branch at, they consider the solution vector x^* to the model (16)–(23). Branching is then performed on the variable i which maximizes the quantity $|x^*_i - \frac{1}{2}|$.

Helmberg et al. [27] used bound U^2_{HRW} in their cutting plane algorithm. In several cases, the upper and lower bound coincide at the root node, hence solving the problem to optimality without any branching.

Caprara et al. [9] used the bound $\hat{U}^2_{\mathrm{CPT}}$, in their depth-first branch-and-bound algorithm, although the Lagrangian dual (27) was only solved at the root node, and the given Lagrangian profits $(\hat{p}_{ij})$ were used in deriving U^{1*}_{CPT} in subsequent nodes. The branching order is determined in advance at the root node, based on the values

$$\pi'_i = p_{ii} + \max\left\{ \sum_{j\in N\setminus\{i\}} \hat{p}_{ji}\overline{x}_j : \sum_{j\in N\setminus\{i\}} w_j\overline{x}_j \leqslant c - w_i,\ 0\leqslant \overline{x}_j \leqslant 1,\ j\in N\setminus\{i\} \right\}. \tag{87}$$

This can be recognized as the upper planes (24), however, with rows and columns interchanged. The motivation for using π'_i instead of π^{5*}_j is that the latter profits are "flattened" by the subgradient optimization procedure, hence leaving no information for defining the branching order. The variables are reordered according to nonincreasing values of π'_i, and each branching node branches on the variable with the smallest index among the unfixed ones. Using some proper data structures, the authors are able to derive the upper bound in $\mathrm{O}(n)$ expected time inside the branch-and-bound algorithm. Since the bound $\hat{U}^2_{\mathrm{CPT}}$ moreover is quite tight, the authors are able to solve some of the largest instances in the literature.

It should finally be mentioned that no branch-and-bound algorithm has been presented which makes use of the bound U^2_{BFS} by Billionnet et al. [6] or the bound $\hat{U}^2_{\mathrm{MV}}$ by Michelon and Veilleux [32].

8. Computational experiments

All the bounds have been implemented and computational experiments were run on Linux-pc with a Pentium III (Coppermine) 930 MHz processor. The implementation and experimental study was carried out by Rasmussen and Sandvik [41]. We consider classical randomly generated QKP instances as proposed by Gallo et al. [18], since these have become the standard tests QKP algorithms. The instances are constructed as follows: let Δ be the *density* of the instance, i.e., the percentage of nonzero elements in the profit matrix P. Each weight w_j is randomly distributed in $[1, 50]$ while the profits $p_{ij} = p_{ji}$ are nonzero with probability Δ, and in this case randomly distributed in $[1, 100]$. Finally, the capacity c is randomly distributed in $[50, \sum_{j=1}^n w_j]$.

The performance of the bounds with respect to tightness and computational effort is reported in the following tables. The tightness of the bounds is measured in comparison to the optimal solution values z^*, which were obtained by the algorithm of Caprara et al. [9]. In three instances it was not possible to find the optimal solution value z^* in reasonable time, in which case a lower bound found by the Lagrangian heuristic of [9] is used. These instances are: $(n = 140, \Delta = 25)$, $(n = 200, \Delta = 25)$ and $(n = 180, \Delta = 50)$, where a single instance out of 10 could not be solved to optimality.

The bound U^2_{BC} by Billionnet and Calmels was calculated using CPLEX 7.0. CPLEX was also used for solving the maximum flow problem in the bound U_{CHM} by Chaillou and Hansen, Mahieu and the bound $\hat{U}^2_{\mathrm{MV}}$ by Michelon and Veilleux. The bounds $U^0_{\mathrm{HRW}}, U^1_{\mathrm{HRW}}, U^2_{\mathrm{HRW}}, U^3_{\mathrm{HRW}}, U^4_{\mathrm{HRW}}$ based on semidefinite programming were calculated using the SeDuMi 1.05 package [44] to MatLab. The bounds $U^1_{\mathrm{GHS}}, U^2_{\mathrm{GHS}}, U^3_{\mathrm{GHS}}, U^4_{\mathrm{GHS}}$ by Gallo, Hammer, Simeone, as well as the bound $\hat{U}^2_{\mathrm{CPT}}$ by Caprara, Pisinger, Toth and the bound $\hat{U}^2_{\mathrm{MV}}$ by Michelon and Veuilleux, involve the solution of a KP. This problem is solved using the `minknap` algorithm by Pisinger [38], which also includes a routine for solving the continuous relaxation in $\mathrm{O}(n)$. Lagrangian multipliers for $\hat{U}^2_{\mathrm{CPT}}$, $\hat{U}^2_{\mathrm{MV}}$ and $\hat{U}^2_{\mathrm{BFS}}$ are calculated using Held and Karp

Table 1
Bounds $U^1_{\mathrm{GHS}}, U^2_{\mathrm{GHS}}, U^3_{\mathrm{GHS}} U^4_{\mathrm{GHS}}, \overline{U}^4_{\mathrm{GHS}}$ (Gallo, Hammer, Simeone) based on upper planes

Δ	n	U^1_{GHS}			U^2_{GHS}			U^3_{GHS}			U^4_{GHS}			$\overline{U}^4_{\mathrm{GHS}}$		
		ub	time	dev(%)	ub	time	dev(%)	ub	time	dev(%)	ub	time	dev(%)	ub	time	dev(%)
5	40	3841.1	0.000	6.38	3841.1	0.002	6.38	3841.1	0.001	6.38	3841.1	0.000	6.38	3841.1	0.000	6.38
	60	7073.5	0.000	17.61	7073.5	0.000	17.61	7073.5	0.001	17.61	7073.5	0.002	17.61	7066.3	0.001	17.49
	80	12 281.6	0.000	22.70	12 281.6	0.000	22.70	12 281.6	0.000	22.70	12 281.6	0.001	22.70	12 280.7	0.001	22.70
	100	20 309.4	0.000	13.84	20 309.4	0.002	13.84	20 309.4	0.003	13.84	20 309.4	0.001	13.84	20 309.4	0.000	13.84
	120	26 156.3	0.001	25.04	26 156.3	0.001	25.04	26 154.3	0.002	25.03	26 153.2	0.001	25.02	26 152.7	0.001	25.02
	140	29 070.8	0.000	40.17	29 070.8	0.000	40.17	28 873.0	0.006	39.22	28 845.9	0.002	39.09	28 825.7	0.003	38.99
	160	47 548.4	0.001	21.53	47 548.4	0.001	21.53	47 466.8	0.003	21.32	47 448.3	0.005	21.27	47 435.9	0.005	21.24
	180	63 277.5	0.002	21.69	63 277.5	0.002	21.69	63 277.5	0.008	21.69	63 277.5	0.002	21.69	63 277.5	0.002	21.69
	200	80 120.0	0.002	18.06	80 120.0	0.004	18.06	80 120.0	0.008	18.06	80 120.0	0.002	18.06	80 120.0	0.004	18.06
	Avg.	32 186.5	0.001	20.78	32 186.5	0.001	20.78	32 155.2	0.004	20.65	32 150.1	0.002	20.63	32 145.5	0.002	20.60
25	40	12 288.7	0.000	35.22	12 177.5	0.001	34.00	11 615.8	0.002	27.82	11 571.1	0.001	27.33	11 491.1	0.000	26.45
	60	33 259.9	0.000	21.47	33 128.1	0.000	20.99	32 331.5	0.000	18.08	32 277.8	0.001	17.88	32 245.3	0.001	17.76
	80	63 938.8	0.000	16.83	63 906.8	0.001	16.78	62 876.6	0.001	14.89	62 831.8	0.003	14.81	62 791.5	0.001	14.74
	100	70 563.5	0.000	61.73	70 402.9	0.003	61.36	68 748.6	0.002	57.57	68 680.3	0.000	57.41	68 573.6	0.001	57.17
	120	121 479.0	0.000	31.08	121 069.7	0.001	30.64	117 974.6	0.002	27.30	117 922.4	0.001	27.24	117 858.6	0.004	27.18
	140	171 338.8	0.002	36.89	171 338.8	0.002	36.89	169 802.6	0.004	35.66	169 756.9	0.001	35.62	169 688.5	0.005	35.57
	160	263 339.6	0.000	18.54	263 339.6	0.001	18.54	262 037.2	0.006	17.96	262 001.2	0.002	17.94	261 932.8	0.006	17.91
	180	285 091.7	0.000	31.80	284 296.4	0.002	31.44	281 104.7	0.008	29.96	281 073.0	0.003	29.95	281 019.8	0.005	29.92
	200	292 617.4	0.003	49.02	292 616.0	0.003	49.02	279 991.0	0.009	42.59	279 812.4	0.010	42.50	279 454.8	0.006	42.31
	Avg.	145 990.8	0.001	33.62	145 808.4	0.002	33.29	142 942.5	0.004	30.20	142 880.8	0.002	30.08	142 784.0	0.003	29.89
50	40	28 155.8	0.000	35.76	27 979.1	0.000	34.91	26 135.2	0.001	26.01	26 025.2	0.001	25.48	25 913.4	0.000	24.95
	60	59 877.1	0.000	36.11	59 025.0	0.000	34.17	56 138.9	0.000	27.61	56 027.2	0.001	27.35	55 863.0	0.002	26.98
	80	129 103.0	0.000	18.48	129 004.7	0.000	18.39	125 162.9	0.003	14.86	125 053.8	0.002	14.76	124 920.5	0.001	14.64
	100	139 788.9	0.001	52.13	134 151.9	0.001	46.00	123 927.1	0.003	34.87	123 774.5	0.001	34.70	123 520.1	0.003	34.42
	120	248 172.4	0.001	31.47	245 019.4	0.005	29.80	231 395.4	0.003	22.59	231 253.2	0.002	22.51	230 967.7	0.003	22.36
	140	337 212.5	0.000	43.17	337 134.5	0.002	43.14	325 147.5	0.003	38.05	324 971.3	0.006	37.97	324 662.3	0.003	37.84
	160	468 072.5	0.000	23.59	463 634.4	0.002	22.41	442 355.9	0.003	16.80	442 214.4	0.004	16.76	441 865.0	0.006	16.67
	180	538 720.9	0.003	30.88	530 614.0	0.007	28.91	501 763.0	0.005	21.90	501 646.6	0.005	21.87	501 303.2	0.007	21.79
	200	725 904.3	0.000	29.32	721 863.2	0.005	28.60	691 684.9	0.010	23.23	691 562.7	0.007	23.20	691 244.4	0.007	23.15
	Avg.	297 223.0	0.001	33.43	294 269.6	0.002	31.81	280 412.3	0.003	25.10	280 281.0	0.003	24.96	280 028.8	0.004	24.75
75	40	41 765.3	0.000	35.10	39 817.9	0.000	28.80	36 626.6	0.000	18.48	36 477.4	0.000	18.00	36 339.8	0.000	17.55
	60	85 625.0	0.001	37.38	80 476.4	0.000	29.12	73 145.1	0.000	17.36	72 960.7	0.000	17.06	72 754.6	0.002	16.73
	80	177 215.8	0.000	26.77	171 998.2	0.001	23.04	161 274.5	0.002	15.37	161 132.5	0.003	15.27	160 930.5	0.000	15.12
	100	252 779.6	0.000	31.20	236 771.4	0.001	22.89	219 355.9	0.004	13.85	219 207.8	0.002	13.77	218 921.5	0.002	13.63
	120	297 904.4	0.000	33.20	271 903.1	0.002	21.57	249 581.5	0.002	11.59	249 436.8	0.003	11.53	249 136.6	0.003	11.39
	140	471 259.2	0.000	36.53	439 300.2	0.002	27.27	403 159.4	0.005	16.80	402 999.0	0.005	16.75	402 665.0	0.004	16.66
	160	701 572.5	0.000	25.79	680 858.0	0.002	22.07	647 736.2	0.004	16.13	647 585.1	0.007	16.11	647 254.7	0.007	16.05
	180	597 836.3	0.002	52.66	528 335.2	0.006	34.91	473 309.0	0.006	20.86	473 137.0	0.009	20.82	472 695.4	0.006	20.71
	200	678 161.9	0.001	51.06	565 674.7	0.004	26.00	508 657.3	0.007	13.30	508 494.1	0.009	13.26	508 140.0	0.009	13.19
	Avg.	367 124.4	0.000	36.63	335 015.0	0.002	26.19	308 093.9	0.003	15.97	307 936.7	0.004	15.84	307 648.7	0.004	15.67
95	40	43 609.7	0.000	63.84	35 994.0	0.001	35.23	31 228.6	0.003	17.33	31 067.0	0.000	16.72	30 914.7	0.000	16.15
	60	97 880.1	0.000	61.20	81 986.1	0.000	35.03	70 530.9	0.002	16.16	70 348.6	0.001	15.86	70 135.2	0.003	15.51
	80	193 982.7	0.000	39.89	172 207.3	0.001	24.19	156 522.4	0.000	12.88	156 327.0	0.002	12.74	156 122.4	0.002	12.59
	100	268 381.2	0.002	49.27	228 352.0	0.002	27.00	204 073.2	0.001	13.50	203 867.7	0.006	13.39	203 674.9	0.000	13.28
	120	498 739.4	0.002	27.16	464 843.4	0.002	18.52	433 961.5	0.002	10.64	433 765.7	0.004	10.59	433 492.7	0.004	10.52
	140	656 028.0	0.002	28.02	613 855.5	0.004	19.79	572 352.4	0.006	11.69	572 135.2	0.003	11.65	571 804.3	0.005	11.58
	160	858 897.5	0.000	34.82	785 773.3	0.002	23.34	714 260.0	0.004	12.11	714 050.8	0.008	12.08	713 586.6	0.008	12.01
	180	1 038 034.9	0.002	31.71	945 744.9	0.004	20.00	876 930.0	0.006	11.27	876 731.3	0.008	11.24	876 250.1	0.003	11.18
	200	1 150 039.1	0.002	53.23	988 208.0	0.003	31.67	858 039.1	0.009	14.33	857 826.2	0.007	14.30	857 181.4	0.015	14.21
	Avg.	533 954.7	0.001	43.24	479 662.7	0.002	26.08	435 322.0	0.004	13.32	435 124.4	0.004	13.17	434 795.8	0.004	13.00
100	40	63 175.9	0.001	23.67	58 870.2	0.001	15.24	56 008.2	0.001	9.64	55 845.8	0.001	9.32	55 755.2	0.000	9.15
	60	136 748.3	0.000	19.93	126 666.9	0.000	11.09	120 976.0	0.001	6.09	120 824.6	0.001	5.96	120 729.2	0.001	5.88
	80	201 042.1	0.000	40.13	173 801.0	0.000	21.15	158 845.4	0.001	10.72	158 649.2	0.003	10.58	158 462.5	0.001	10.45
	100	346 321.2	0.001	33.11	310 725.3	0.001	19.42	287 406.8	0.003	10.46	287 187.8	0.003	10.38	286 972.2	0.002	10.29
	120	497 050.4	0.001	37.67	443 560.8	0.004	22.86	401 963.2	0.002	11.33	401 728.2	0.003	11.27	401 426.3	0.005	11.19
	140	739 307.0	0.000	26.21	688 295.1	0.002	17.50	646 400.4	0.006	10.35	646 167.2	0.004	10.31	645 807.9	0.004	10.25
	160	757 578.0	0.000	38.14	656 073.7	0.001	19.63	601 496.7	0.007	9.68	601 297.2	0.006	9.64	601 005.6	0.005	9.59
	180	1 000 484.5	0.002	46.30	849 344.1	0.005	24.20	751 328.5	0.005	9.86	751 114.8	0.007	9.83	750 657.8	0.009	9.77
	200	1 248 949.6	0.003	40.76	1 082 530.0	0.004	22.01	976 779.5	0.009	10.09	976 574.8	0.010	10.07	976 116.1	0.008	10.01
	Avg.	554 517.4	0.001	33.99	487 763.0	0.002	19.23	444 578.3	0.004	9.80	444 376.6	0.004	9.71	444 103.6	0.004	9.62
Average		321 832.8	0.001	33.62	295 784.2	0.002	26.23	273 917.4	0.004	19.18	273 791.6	0.003	19.06	273 584.4	0.003	18.92

Table 2
Bounds U_{CHM} (Chailleu, Hansen, Mahieu), $\hat{U}^2_{\mathrm{MV}}$ (Michelon, Veilleux), $\hat{U}^2_{\mathrm{BFS}}$ (Billionnet, Faye, Soutif), $\hat{U}^2_{\mathrm{CPT}}$ (Caprara, Pisinger, Toth), U^2_{BC} (Billionnet, Calmels) based on linearization and Lagrangian relaxation

Δ	n	U_{CHM}			$\hat{U}^2_{\mathrm{MV}}$			$\hat{U}^2_{\mathrm{BFS}}$			$\hat{U}^2_{\mathrm{CPT}}$			U^2_{BC}		
		ub	time	dev(%)	ub	time	dev(%)	ub	time	dev(%)	ub	time	dev(%)	ub	time	dev(%)
5	40	3639.7	0.0	0.80	3620.4	0.3	0.27	3637.4	3.6	0.74	3655.1	0.0	1.23	3636.0	25.2	0.70
	60	6053.3	0.0	0.65	6036.3	0.6	0.37	6043.9	9.6	0.49	6099.1	0.1	1.41	6041.2	160.0	0.45
	80	10 053.9	0.0	0.45	10 038.1	1.4	0.29	10 052.5	15.8	0.43	10 165.5	0.2	1.56	–		
	100	17 896.2	0.0	0.32	17 868.9	2.0	0.16	17 894.4	24.8	0.31	18 075.6	0.4	1.32	–		
	120	20 977.1	0.0	0.28	20 954.0	3.2	0.17	20 981.1	40.4	0.30	21 204.1	0.7	1.37	–		
	140	20 854.8	0.0	0.56	20 848.0	2.2	0.53	20 883.8	75.6	0.70	21 185.4	1.5	2.15	–		
	160	39 175.6	0.0	0.13	39 155.0	12.5	0.08	39 178.4	76.5	0.14	39 707.1	2.1	1.49	–		
	180	52 075.7	0.0	0.14	52 057.9	7.3	0.11	52 086.6	95.0	0.17	52 853.8	3.7	1.64	–		
	200	67 917.9	0.1	0.08	67 897.1	22.0	0.05	67 926.2	111.0	0.09	68 838.9	4.8	1.44	–		
	Avg.	26 516.0	0.0	0.38	26 497.3	5.7	0.22	26 520.5	50.2	0.37	26 865.0	1.5	1.51	4838.6	92.6	0.57
25	40	9367.0	0.0	3.07	9303.9	0.6	2.38	9237.3	3.7	1.65	9352.2	0.0	2.91	9288.8	35.5	2.21
	60	27 539.1	0.0	0.58	27 483.9	1.8	0.37	27 477.8	8.5	0.35	27 633.3	0.1	0.92	27 486.2	326.4	0.38
	80	55 282.7	0.0	1.02	55 147.2	6.5	0.77	55 081.0	16.1	0.65	55 460.4	0.3	1.34	–		
	100	44 712.0	0.0	2.48	44 696.6	3.0	2.44	43 927.0	30.0	0.68	44 848.1	0.7	2.79	–		
	120	93 198.4	0.1	0.57	93 162.3	4.9	0.53	92 916.4	38.6	0.26	93 534.4	1.0	0.93	–		
	140	126 998.9	0.1	1.46	126 998.1	2.4	1.46	125 933.2	59.4	0.61	127 625.4	2.1	1.96	–		
	160	223 255.6	0.2	0.50	223 198.4	64.7	0.47	222 502.1	62.2	0.16	224 379.7	2.5	1.01	–		
	180	217 898.2	0.2	0.74	217 877.6	34.8	0.73	216 881.7	93.5	0.27	218 847.9	4.5	1.18	–		
	200	199 251.7	0.3	1.47	199 244.9	20.3	1.47	197 078.6	122.3	0.36	200 111.3	7.7	1.91	–		
	Avg.	110 833.7	0.1	1.32	110 790.3	15.5	1.18	110 115.0	48.3	0.55	111 310.3	2.1	1.66	18 387.5	181.0	1.30
50	40	21 507.2	0.0	3.70	21 430.6	1.1	3.33	21 015.5	3.7	1.33	21 423.1	0.1	3.29	21 092.1	55.9	1.70
	60	45 246.5	0.0	2.85	45 180.2	3.1	2.70	44 289.7	8.4	0.67	44 923.5	0.2	2.11	44 363.4	699.3	0.84
	80	109 858.4	0.0	0.82	109 724.1	9.4	0.69	109 385.4	14.7	0.38	109 899.2	0.3	0.85	–		
	100	95 205.3	0.1	3.61	95 169.7	15.4	3.57	92 343.2	24.7	0.50	94 097.5	0.7	2.40	–		
	120	191 407.1	0.2	1.40	191 308.8	48.5	1.35	189 421.8	36.1	0.35	191 226.2	1.2	1.31	–		
	140	238 657.8	0.3	1.33	238 641.4	28.4	1.32	235 975.8	47.3	0.19	238 957.6	2.2	1.45	–		
	160	383 141.4	0.5	1.16	382 994.6	164.9	1.12	379 500.9	66.1	0.20	382 115.9	2.9	0.89	–		
	180	418 091.0	0.6	1.57	418 083.1	27.9	1.57	412 487.2	84.8	0.21	418 087.4	5.2	1.57	–		
	200	565 996.5	0.8	0.83	565 987.2	43.8	0.83	562 206.9	107.0	0.16	566 054.7	6.0	0.84	–		
	Avg.	229 901.2	0.3	1.92	229 835.5	38.1	1.83	227 402.9	43.7	0.44	229 642.8	2.1	1.64	32 727.8	377.6	1.27
75	40	32 125.6	0.0	3.92	31 951.6	3.3	3.36	31 403.0	3.7	1.58	31 672.9	0.1	2.45	31 404.3	29.4	1.59
	60	63 666.2	0.0	2.15	63 500.4	6.4	1.88	62 831.3	8.1	0.81	63 194.6	0.1	1.39	62 827.7	116.4	0.80
	80	141 916.0	0.1	1.52	141 911.1	2.0	1.52	140 108.4	13.5	0.23	141 223.9	0.3	1.03	140 095.9	978.8	0.22
	100	196 450.7	0.2	1.96	196 389.3	15.1	1.93	193 462.1	23.3	0.41	194 996.5	0.5	1.21	193 449.6	2110.9	0.41
	120	227 887.5	0.3	1.89	227 679.9	195.5	1.80	224 747.6	31.2	0.49	225 292.5	0.8	0.73	224 712.6	2982.4	0.47
	140	351 873.0	0.5	1.94	351 869.9	30.6	1.94	345 831.8	47.5	0.19	348 988.8	1.7	1.11	–		
	160	564 646.6	0.7	1.24	564 514.8	506.8	1.21	558 749.1	63.1	0.18	562 433.9	2.5	0.84	–		
	180	398 651.4	0.8	1.80	398 651.4	14.4	1.80	392 090.6	77.8	0.12	394 650.6	3.6	0.78	–		
	200	458 038.2	3.7	2.03	458 038.2	17.9	2.03	449 686.5	102.5	0.17	452 453.2	4.2	0.78	–		
	Avg.	270 583.9	0.7	2.05	270 500.7	88.0	1.94	266 545.6	41.2	0.46	268 323.0	1.5	1.15	130 498.0	1243.6	0.70
95	40	29 092.9	0.0	9.30	29 086.9	0.4	9.28	27 043.1	3.8	1.60	27 334.2	0.1	2.69	27 041.3	17.7	1.59
	60	63 275.0	0.0	4.21	63 275.0	0.6	4.21	61 131.8	8.4	0.68	61 479.3	0.1	1.25	61 128.2	102.2	0.68
	80	142 257.8	0.1	2.59	142 190.8	11.2	2.54	139 650.0	14.6	0.71	140 216.3	0.3	1.12	139 635.3	445.8	0.70
	100	182 726.9	0.2	1.63	182 719.2	13.1	1.62	180 634.5	21.0	0.46	181 429.9	0.4	0.91	180 622.6	1346.5	0.46
	120	398 917.3	0.4	1.71	398 845.6	89.4	1.69	393 332.2	32.7	0.28	394 823.5	0.7	0.66	–		
	140	521 751.8	0.5	1.82	521 724.4	113.2	1.81	513 564.1	41.5	0.22	515 797.6	1.3	0.65	–		
	160	650 989.7	0.8	2.18	650 944.8	221.6	2.18	639 207.5	62.9	0.33	641 101.1	2.1	0.63	–		
	180	804 380.2	1.0	2.06	804 380.2	15.1	2.06	790 677.2	80.8	0.32	793 737.2	3.2	0.71	–		
	200	775 864.7	1.2	3.38	775 781.3	563.3	3.37	751 654.3	103.7	0.15	754 584.2	4.7	0.54	–		
	Avg.	396 584.0	0.5	3.21	396 549.8	114.2	3.20	388 543.9	41.0	0.53	390 055.9	1.4	1.02	102 106.8	478.0	0.86
100	40	53 236.7	0.0	4.22	53 166.3	2.0	4.08	51 876.2	3.6	1.55	52 049.7	0.1	1.89	51 876.7	24.8	1.55
	60	116 462.6	0.0	2.14	116 361.3	11.2	2.05	115 150.9	8.2	0.99	115 359.9	0.1	1.17	115 140.3	151.1	0.98
	80	146 642.8	0.1	2.22	146 622.1	20.4	2.20	144 221.2	14.0	0.53	144 521.1	0.3	0.74	144 201.2	397.7	0.51
	100	267 738.1	0.2	2.90	267 738.1	4.4	2.90	260 899.8	22.0	0.27	261 785.9	0.5	0.61	260 879.8	1990.1	0.27
	120	369 079.9	0.4	2.23	369 069.6	20.0	2.22	362 426.4	33.1	0.38	363 701.9	0.9	0.74	–		
	140	597 091.9	0.6	1.93	597 021.5	122.7	1.92	587 282.8	47.9	0.26	588 880.7	1.3	0.53	–		
	160	560 753.1	0.8	2.25	560 522.5	978.7	2.21	550 132.9	60.0	0.31	551 180.0	1.9	0.50	–		
	180	701 237.7	1.0	2.54	701 237.4	21.1	2.54	685 521.2	78.3	0.24	687 030.3	2.9	0.46	–		
	200	902 041.2	1.4	1.66	902 026.3	79.8	1.66	891 733.9	103.1	0.50	893 151.6	4.2	0.66	–		
	Avg.	412 698.2	0.5	2.45	412 640.6	140.0	2.42	405 471.7	41.1	0.56	406 406.8	1.3	0.81	143 024.5	640.9	0.83
Average		241 186.2	0.3	1.89	241 135.7	66.9	1.80	237 433.3	44.3	0.49	238 767.3	1.7	1.30	91 838.1	631.4	0.87

Table 3
Bounds $U^0_{\mathrm{HRW}}, U^1_{\mathrm{HRW}}, U^2_{\mathrm{HRW}}, U^3_{\mathrm{HRW}}, U^4_{\mathrm{HRW}}$ (Helmberg, Rendl, Weismantel) based on semidefinite programming

Δ	n	U^0_{HRW}			U^1_{HRW}			U^2_{HRW}			U^3_{HRW}			U^4_{HRW}		
		ub	time	dev(%)	ub	time	dev(%)	ub	time	dev(%)	ub	time	dev(%)	ub	time	dev(%)
5	40	3733.5	3.9	3.40	3657.2	5.6	1.29	3654.6	5.3	1.22	3654.6	5.9	1.22	3654.4	14.2	1.21
	60	6473.5	9.7	7.64	6176.5	12.4	2.70	6125.0	13.5	1.84	6124.5	14.2	1.83	6124.1	75.8	1.83
	80	11 032.7	25.5	10.23	10 369.5	29.1	3.60	10 204.5	34.2	1.95	10 204.4	36.8	1.95	10 204.4	222.1	1.95
	100	19 056.3	54.3	6.82	18 239.4	68.1	2.24	18 040.2	83.5	1.12	18 040.0	90.6	1.12	18 039.7	579.5	1.12
	120	23 224.5	125.4	11.02	21 794.5	143.0	4.19	21 249.5	170.4	1.58	21 249.4	180.1	1.58	21 249.0	1053.8	1.58
	140	24 465.5	243.1	17.97	22 447.5	273.7	8.24	21 252.0	309.4	2.47	21 251.3	327.9	2.47	21 251.0	1810.4	2.47
	160	43 202.7	408.0	10.42	40 671.9	509.7	3.95	39 529.2	566.3	1.03	39 528.7	619.8	1.03	39 528.5	2410.6	1.03
	180	57 955.0	644.5	11.45	54 607.6	799.2	5.01	52 618.5	909.3	1.19	52 617.9	1014.2	1.19	52 617.4	3928.0	1.19
	200	74 487.9	904.3	9.76	70 189.1	1107.9	3.43	68 330.9	1345.5	0.69	68 330.8	1564.6	0.69	–		
	Avg.	29 292.4	268.8	9.86	27 572.6	327.6	3.85	26 778.3	381.9	1.46	26 778.0	428.2	1.45	21 583.6	1261.8	1.55
25	40	10 665.2	3.8	17.36	10 036.3	4.2	10.44	9359.5	4.7	2.99	9357.3	5.4	2.97	9355.6	17.8	2.95
	60	30 610.2	10.0	11.79	28 771.0	11.5	5.07	27 574.6	14.8	0.71	27 573.1	15.3	0.70	27 572.3	69.6	0.70
	80	60 079.2	24.4	9.78	57 439.9	28.7	4.96	55 204.5	38.2	0.87	55 198.2	42.1	0.86	55 193.7	255.5	0.85
	100	57 340.8	55.2	31.42	51 618.5	61.1	18.31	44 169.9	72.0	1.23	44 162.7	80.6	1.22	44 158.9	537.0	1.21
	120	108 196.4	120.3	16.75	100 799.9	138.6	8.77	93 123.7	185.1	0.49	93 118.0	204.2	0.48	93 113.9	1090.5	0.47
	140	151 620.2	240.7	21.13	139 678.8	274.3	11.59	125 784.4	340.3	0.49	125 782.4	350.7	0.49	125 781.6	1968.3	0.49
	160	247 966.9	373.4	11.62	234 982.6	457.7	5.78	222 693.4	626.3	0.25	222 690.8	679.3	0.25	222 689.6	3146.9	0.25
	180	257 229.1	583.3	18.92	237 930.8	664.0	10.00	216 969.9	888.5	0.31	216 953.5	929.8	0.30	216 944.7	4885.8	0.30
	200	244 981.3	916.2	24.76	224 514.0	1019.1	14.33	197 041.6	1306.5	0.34	197 038.1	1320.2	0.34	–		
	Avg.	129 854.4	258.6	18.17	120 641.3	295.5	9.92	110 213.5	386.3	0.85	110 208.2	403.1	0.85	99 351.3	1496.4	0.90
50	40	24 983.6	3.8	20.46	23 277.1	4.1	12.23	21 124.5	5.1	1.85	21 119.3	5.7	1.83	21 117.7	17.3	1.82
	60	53 244.1	10.1	21.03	49 342.1	11.0	12.16	44 447.9	13.9	1.03	44 437.1	15.7	1.01	44 429.1	76.0	0.99
	80	121 197.5	23.0	11.22	115 358.3	27.5	5.86	109 489.4	39.6	0.48	109 488.6	39.4	0.48	109 488.1	247.7	0.48
	100	117 328.0	55.1	27.69	107 124.0	61.0	16.58	92 496.8	79.2	0.66	92 462.2	87.7	0.63	92 447.6	576.6	0.61
	120	221 980.3	121.9	17.60	207 665.9	136.7	10.02	189 511.1	193.2	0.40	189 489.6	206.4	0.39	189 481.2	1185.8	0.38
	140	295 612.0	240.2	25.51	268 514.0	265.6	14.00	236 070.2	396.7	0.23	236 067.8	397.2	0.23	236 066.9	2370.4	0.23
	160	429 187.8	397.0	13.32	407 314.1	449.8	7.54	379 608.6	679.1	0.23	379 553.2	750.9	0.21	379 541.8	3333.8	0.21
	180	484 159.1	627.1	17.62	453 260.3	704.2	10.11	412 487.3	1022.0	0.21	412 469.9	1113.6	0.20	412 453.4	4668.1	0.20
	200	657 937.2	855.6	17.21	614 041.8	947.6	9.39	562 046.7	1399.9	0.13	562 031.4	1576.1	0.13	–		
	Avg.	267 292.2	259.3	19.07	249 544.2	289.7	10.88	227 475.8	425.4	0.58	227 457.7	465.9	0.57	185 628.2	1559.5	0.61
75	40	37 548.3	4.0	21.46	34 913.8	4.0	12.94	31 494.2	5.1	1.88	31 491.8	5.4	1.87	31 491.0	19.8	1.87
	60	75 237.2	9.5	20.72	69 692.8	10.8	11.82	62 953.6	14.3	1.01	62 939.2	16.4	0.98	62 931.5	72.8	0.97
	80	162 244.2	22.5	16.06	152 447.5	25.6	9.05	140 176.6	41.1	0.28	140 169.1	42.3	0.27	140 167.4	241.4	0.27
	100	225 336.3	53.9	16.96	212 220.2	61.0	10.15	193 579.0	89.8	0.47	193 564.9	94.0	0.47	193 561.1	628.0	0.46
	120	259 870.3	114.6	16.19	245 762.0	126.7	9.88	225 169.3	185.7	0.68	225 123.2	210.3	0.65	225 084.5	1254.4	0.64
	140	413 286.1	236.0	19.73	386 186.9	262.4	11.88	345 988.3	406.1	0.24	345 932.7	447.8	0.22	345 914.5	2168.5	0.21
	160	645 669.4	382.5	15.76	606 143.0	453.0	8.68	558 900.2	713.6	0.21	558 865.2	749.2	0.20	558 840.0	3191.2	0.20
	180	490 277.7	651.2	25.19	451 350.7	697.3	15.25	392 968.1	1064.5	0.35	392 810.0	1358.0	0.31	–		
	200	541 403.4	913.6	20.60	508 731.8	963.6	13.32	450 908.6	1382.2	0.44	450 544.0	2036.5	0.36	–		
	Avg.	316 763.7	265.3	19.19	296 383.2	289.4	11.44	266 904.2	433.6	0.61	266 826.7	551.1	0.59	222 570.0	1082.3	0.66
95	40	36 807.2	3.7	38.28	33 283.6	3.9	25.05	27 095.3	4.9	1.80	27 091.4	5.3	1.78	27 089.8	17.3	1.78
	60	81 456.9	9.4	34.16	73 408.4	10.7	20.90	61 213.7	14.9	0.82	61 197.2	16.6	0.79	61 194.6	75.6	0.78
	80	169 535.5	23.2	22.26	157 357.2	26.2	13.48	139 745.0	40.8	0.78	139 722.3	43.8	0.76	139 712.1	256.5	0.75
	100	224 435.2	54.6	24.82	206 369.1	59.6	14.78	180 961.0	96.8	0.65	180 902.5	102.5	0.61	180 857.3	655.5	0.59
	120	456 240.5	117.1	16.32	428 790.8	131.7	9.32	393 541.8	228.7	0.34	393 487.7	233.0	0.32	393 442.7	1248.6	0.31
	140	601 043.5	236.5	17.29	562 747.8	259.9	9.82	513 840.6	428.0	0.27	513 830.4	464.6	0.27	513 820.3	2165.8	0.27
	160	770 867.4	369.1	21.00	716 311.2	419.3	12.44	639 263.0	729.5	0.34	639 260.0	732.6	0.34	639 259.0	3426.4	0.34
	180	932 655.6	619.4	18.34	875 004.8	662.9	11.02	791 017.0	1182.8	0.37	790 957.8	1227.4	0.36	790 904.6	5121.6	0.35
	200	970 567.7	906.3	29.32	887 107.8	967.7	18.20	751 750.0	1678.9	0.16	751 698.7	1762.3	0.16	–		
	Avg.	471 512.2	259.9	24.64	437 820.1	282.4	15.00	388 714.2	489.5	0.61	388 683.1	509.8	0.60	343 285.0	1620.9	0.65
100	40	59 786.2	3.7	17.04	56 495.4	4.1	10.59	51 930.4	5.8	1.66	51 929.7	5.9	1.66	51 929.5	21.1	1.66
	60	127 785.8	9.3	12.07	122 263.6	10.5	7.22	115 225.5	16.2	1.05	115 218.1	15.9	1.05	115 216.4	83.8	1.04
	80	175 450.4	22.3	22.30	162 355.3	25.8	13.17	144 311.9	38.1	0.59	144 293.3	41.5	0.58	144 281.8	258.0	0.57
	100	311 829.0	50.5	19.85	292 046.7	57.3	12.25	260 964.3	93.8	0.30	260 939.3	100.5	0.29	260 935.9	639.7	0.29
	120	441 412.6	119.5	22.26	408 114.1	130.4	13.04	362 481.0	238.6	0.40	362 474.8	235.2	0.40	362 473.9	1257.3	0.40
	140	684 567.3	238.9	16.86	642 049.5	260.2	9.61	587 355.6	459.0	0.27	587 327.9	457.8	0.26	587 322.2	2203.8	0.26
	160	660 339.6	405.8	20.41	616 473.3	442.0	12.41	550 574.6	717.2	0.39	550 492.8	803.8	0.38	550 447.4	3503.7	0.37
	180	849 735.0	620.9	24.25	785 416.9	678.2	14.85	685 724.4	1133.7	0.27	685 629.1	1210.9	0.26	685 569.0	4833.8	0.25
	200	1 081 628.9	893.2	21.91	1 004 023.2	970.0	13.16	892 090.1	1620.9	0.54	891 970.9	1757.8	0.53	–		
	Avg.	488 059.4	262.7	19.66	454 359.8	286.5	11.81	405 628.6	480.4	0.61	405 586.2	514.4	0.60	344 772.0	1600.1	0.60
Average		283 795.7	262.4	18.43	264 386.9	295.2	10.48	237 619.1	432.8	0.79	237 590.0	478.7	0.78	202 445.8	1444.4	0.83

Table 4
Dominance of bounds: An entry $U_{\text{row}} \leqslant U_{\text{column}}$ means that U_{column} is dominated by U_{row}

	U^1_{GHS}	U^2_{GHS}	U^3_{GHS}	U^4_{GHS}	$\overline{U}^4_{\text{GHS}}$	U_{CHM}	$\hat{U}^2_{\text{MV}}$	$\hat{U}^2_{\text{BFS}}$	$\hat{U}^2_{\text{CPT}}$	U^2_{BC}	U^0_{HRW}	U^1_{HRW}	U^2_{HRW}	U^3_{HRW}	U^4_{HRW}
U^1_{GHS}															
U^2_{GHS}	$\leqslant$														
U^3_{GHS}	$\leqslant$	–													
U^4_{GHS}	$\leqslant$	$\leqslant$	$\leqslant$												
$\overline{U}^4_{\text{GHS}}$	$\leqslant$	$\leqslant$	$\leqslant$	$\leqslant$											
U_{CHM}	–	–	–	–	–										
$\hat{U}^2_{\text{MV}}$			–	–	–	$\leqslant$									
$\hat{U}^2_{\text{BFS}}$	–	–	–	–	–	–	–								
$\hat{U}^2_{\text{CPT}}$	–	–	–	–	–	–	–	–							
U^2_{BC}	–	–	–	–	–		–	–	$\leqslant$						
U^0_{HRW}	–	–	–	–	–										
U^1_{HRW}	–	–	–	–	–				–		$\leqslant$				
U^3_{HRW}	–	–	–	–	–	–	–	–	–	–	$\leqslant$	$\leqslant$			
U^3_{HRW}	–	–	–	–	–	–	–	–	–	–	$\leqslant$	$\leqslant$	$\leqslant$		
U^4_{HRW}	–	–	–	–	–	–	–	–	–	–	$\leqslant$	$\leqslant$	$\leqslant$	$\leqslant$	

An entry marked "–" indicates that no dominance exists between the bounds.

[25] subgradient optimization. The Lagrangian dual problem corresponding to U_{CHM} is solved to optimality using the technique described in Section 2.2.

When comparing solution times in the following tables, one should keep in mind that several of the bounds could have been implemented more efficiently, e.g. by using specialized algorithms for solving the maximum flow problems involved, or by using more efficient general solvers. However, the times still give an indication of the time complexity. What the tables do not show, is whether the bounds may be derived quicker inside a branch-and-bound algorithm, by reusing calculations from the previous bound. This information may be more important than the absolute calculation time, but only few of the bounds are described in the literature with focus on this aspect.

In Tables 1–3 we report the average values of 10 instances solved for each value of the density Δ and instance size n. For each bound we first report the average value of the bound (`ub`), then the average solution time (`time`) in seconds, and finally the average deviation (`dev`) from the optimal solution z^*. It is seen that the bounds based on upper planes $U^1_{\text{GHS}}, U^2_{\text{GHS}}, U^3_{\text{GHS}}, U^4_{\text{GHS}}, \overline{U}^4_{\text{GHS}}$ are very fast to derive, but the quality of the bounds is disappointing. The bounds based on Lagrangian relaxation $U_{\text{CHM}}, \hat{U}^2_{\text{MV}}, \hat{U}^2_{\text{BFS}}$ are in general much tighter, but the computational times are larger. The two bounds from linearization $\hat{U}^2_{\text{CPT}}, U^2_{\text{BC}}$ are in general both quite tight, but U^2_{BC} is very time consuming, and hence it can only be calculated for small instances. Finally the bounds based on semidefinite programming show that the bounds $U^0_{\text{HRW}}, U^1_{\text{HRW}}$ are of poor quality, while $U^2_{\text{HRW}}, U^3_{\text{HRW}}, U^4_{\text{HRW}}$ are all of the same good quality, but demand increasing computational time.

The tightest of all bounds appears to be the bound $\hat{U}^2_{\text{BFS}}$ although it is also one of the most expensive to derive as it demands an enumerative search. This means in particular, that it is not well suited for use inside a branch-and-bound algorithm since it is difficult to reuse calculations from previous branching nodes. The bound U_{CHM} by Chaillou, Hansen, Mahieu is on the other hand cheap to derive and has a good quality, in particular for low densities. For high densities the bound $\hat{U}^2_{\text{CPT}}$ by Caprara, Pisinger, Toth has a good quality and it is also relatively cheap to derive.

For variable reduction as described in Section 3, one may use some of the more expensive bounds, since only $O(n)$ calculations of the bounds are needed. Possibly, one could start by using the cheaper bounds, and gradually apply tighter bounds on each remaining set of variables.

8.1. *Relative strength of bounds*

The computational experiments have also been used to prove nonexistence of dominance between bounds. More formally, for two given bounds U_1 and U_2, if we can find two instances I_1 and I_2 such that $U_1(I_1) < U_2(I_1)$ while $U_1(I_2) > U_2(I_2)$ then no dominance exists between the bounds. Surprisingly enough, the experimental study was able to prove nonexistence of dominance between a majority of the bounds. Table 4 depicts the situations where dominance has been proved in the theory part, where no dominance exists due to experimentally found counter-examples, and finally situations where dominance is still open.

Notice, that the three bounds $\hat{U}^2_{\mathrm{MV}}$, $\hat{U}^2_{\mathrm{BFS}}$, $\hat{U}^2_{\mathrm{CPT}}$ are all based on sub-optimal Lagrangian multipliers, hence dominance of these bounds does not say anything about the quality of the bounds U^2_{MV}, U^2_{BFS}, U^2_{CPT} with optimal multipliers.

9. Conclusion

It should be clear from the introduction that QKP has numerous practical applications, and hence it is of great importance to develop exact and heuristic algorithms for solving large-sized instances of QKP.

The largest instances reported solved in the literature are due to Caprara et al. [9] who solved dense QKP instances with up to 400 variables, however using quite large solution times. These results are lacking behind similar achievements for the KP where problems with several thousands variables can be solved within fractions of a second. In order to reach similar results we need to find tighter and faster bounds for the QKP. Although both goals may be difficult to achieve, one could use the tight but computationally expensive bounds to reduce the size of an instance, while computationally cheaper bounds can be used inside a branch-and-bound algorithm. The reformulation framework of Caprara, Pisinger, Toth may also be a challenging direction of research, which possibly could be used in connection with the other bounds presented. Sparse instances of QKP are only reported solved for instances with a few hundreds of variables, hence lacking behind the dense problems. The present computational study has shown that several bounds are well suited for these instances.

Most authors in the literature perform their computational experiments on randomly generated instances with the profits and weights uniformly distributed in a narrow interval. It would be relevant to extend the benchmark tests to other instance types which reflect real-life data in a larger extent. Typical instances could include geometric problems, graph problems, and correlated problems.

One of the major open problems in the literature on QKP is whether the problem accepts an approximation algorithm with fixed approximation ratio if all coefficients $p_{ij} \geqslant 0$. Until this has been settled, the QKP is an ideal playground for experimenting with metaheuristics, since it is so easy to state and difficult to solve. Neighborhoods from the ordinary KP can be used to explore the solution space, complemented by neighborhoods specially designed for QKP. Hence also from this point of view we may expect an increasing interest in QKP.

Acknowledgments

The author would like to thank Jarl Friis for valuable comments during the writing of this paper—in particular with the section on semidefinite programming, and Anders Bo Rasmussen and Rune Sandvik for having implemented the upper bounds and performed the computational experiments. The authors also wish to thank the editor and two anonymous referees for their helpful comments.

References

[1] W. Adams, H. Sherali, A tight linearization and an algorithm for zero–one quadratic programming problems, Management Sci. 32 (1986) 1274–1290.
[2] E. Balas, Facets of the knapsack polytope, Math. Programming 8 (1975) 146–164.
[3] E. Balas, S. Ceria, G. Cornuejols, A lift-and-project cutting plane algorithm for mixed 0–1 programs, Math. Programming 58 (1993) 295–324.
[4] M. Bauvin, M. Goemans, Personal communication, see Helmberg, 2000, 1999.
[5] A. Billionnet, F. Calmels, Linear programming for the 0–1 quadratic knapsack problem, European J. Oper. Res. 92 (1996) 310–325.
[6] A. Billionnet, A. Faye, E. Soutif, A new upper-bound and an exact algorithm for the 0–1 quadratic knapsack problem, Eur. J. Oper. Res. 112 (1999) 664–672.
[7] K. Bretthauer, B. Shetty, S. Syam, A branch-and-bound algorithm for integer quadratic knapsack problems, ORSA J. Comput. 7 (1995) 109–116.
[8] P. Brucker, An $\mathrm{O}(n)$ algorithm for quadratic knapsack problems, Oper. Res. Lett. 3 (1984) 163–166.
[9] A. Caprara, D. Pisinger, P. Toth, Exact solution of the quadratic knapsack problem, INFORMS J. Comput. 11 (1999) 125–137.
[10] P. Carraresi, F. Malucelli, A reformulation scheme and new lower bounds for the qap, in: P. Pardalos, H. Wolkowicz (Eds.), Quadratic Assignment and Related Problems, AMS Press, 1994, pp. 147–160.
[11] E. Cela, The Quadratic Assignment Problem: Theory and Algorithms, Kluwer Academic Publishers, Dordrecht, 1998.
[12] P. Chaillou, P. Hansen, Y. Mahieu, Best network flow bound for the quadratic knapsack problem, in: B. Simeone (Ed.), Combinatorial Optimization, Lecture Notes in Mathematics, vol. 1403, Springer, Berlin, 1989, pp. 225–235.
[13] P. Crescenzi, V. Kann, A compendium of NP optimization problems, ⟨http://www.nada.kth.se/viggo/problemlist/compendium.html⟩, 2002.
[14] G. Dijkhuizen, U. Faigle, A cutting-plane approach to the edge-weighted maximal clique problem, Eur. J. Oper. Res. 69 (1993) 121–130.

[15] M.L. Fisher, The Lagrangian relaxation method for solving integer programming problems, Management Sci. 27 (1981) 1–18.

[16] C. Ferreira, A. Martin, C. de Souza, R. Weismantel, L. Wolsey, Formulations and valid inequalities for node capacitated graph partitioning, Math. Programming 74 (1996) 247–266.

[17] G. Gallo, M. Grigoriadis, R. Tarjan, A fast parametric maximum flow algorithm and applications, SIAM J. Comput. 18 (1989) 30–55.

[18] G. Gallo, P. Hammer, B. Simeone, Quadratic knapsack problems, Math. Programming Study 12 (1980) 132–149.

[19] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.

[20] A.M. Geoffrion, Lagrangian relaxation for integer programming, Math. Programming Study 2 (1974) 82–114.

[21] F. Glover, G. Kochenberger, Solving quadratic knapsack problems by reformulation and tabu search, single constraint case, in: P.M. Pardalos, A. Migdalas, R.E. Burkard (Eds.), Combinatorial and Global Optimization, vol. 14, World Scientific, Singapore, 2002.

[22] M. Grötschel, A.S.L. Lovász, Geometric Algorithms and Combinatorial Optimization, Algorithms and Combinatorics, vol. 2, Springer, Berlin, 1988.

[23] P. Hammer, R. Holzman, Approximations of pseudo-Boolean functions; application to game theory, ZOR—Methods Models Oper. Res. 36 (1992) 3–21.

[24] P. Hammer, D. Rader Jr., Efficient methods for solving quadratic 0–1 knapsack problems, INFOR 35 (1997) 170–182.

[25] M. Held, R. Karp, The traveling salesman problem and minimum spanning trees: part ii, Math. Programming 1 (1971) 6–25.

[26] C. Helmberg, Semidefinite programming for combinatorial optimization, Technical Report ZIP-Report ZR-00-34, Konrad-Zuse-Zentrum für Informationstechnik Berlin, habilitationsschrift, TU Berlin, 2000.

[27] C. Helmberg, F. Rendl, R. Weismantel, Quadratic knapsack relaxations using cutting planes and semidefinite programming, in: W. Cunningham, S. McCormick, M. Queyranne (Eds.), Proceedings of the Fifth IPCO Conference, Lecture Notes in Computer Science, vol. 1084, Springer, Berlin, 1996, pp. 175–189.

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

[29] E. Johnson, A. Mehrotra, G. Nemhauser, Min-cut clustering, Math. Programming 62 (1993) 133–152.

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

[31] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, New York, 1990.

[32] P. Michelon, L. Veilleux, Lagrangean methods for the 0–1 quadratic knapsack problem, Eur. J. Oper. Res. 92 (1996) 326–341.

[33] M. Padberg, A note on zero–one programming, Oper. Res. 23 (1975) 833–837.

[34] M. Padberg, The Boolean quadratic polytope: some characteristics, facets and relatives, Math. Programming 45 (1989) 139–172.

[35] K. Park, K. Lee, S. Park, An extended formulation approach to the edge-weighted maximal clique problem, Eur. J. Oper. Res. 95 (1996) 671–682.

[36] J. Picard, H. Ratliff, Minimum cuts and related problems, Networks 5 (1974) 357–370.

[37] C. Peterson, A capital budgeting heuristic algorithm using exchange operations, AIIE Trans. 6 (1974) 143–150.

[38] D. Pisinger, A minimal algorithm for the 0–1 knapsack problem, Oper. Res. 45 (1997) 758–767.

[39] D. Pisinger, P. Toth, Knapsack problems, in: D. Du, P. Pardalos (Eds.), Handbook of Combinatorial Optimization, Kluwer Academic Publishers, Dordrecht, 1998, pp. 1–89.

[40] D. Rader Jr., G. Woeginger, The quadratic 0–1 knapsack problem with series–parallel support, Oper. Res. Lett. 30 (2002) 159–166.

[41] A. Rasmussen, R. Sandvik, Kvaliteten af grænseværdier for det kvadratiske knapsack problem, project 02-09-7, DIKU, University of Copenhagen (D. Pisinger, supervisor), 2003.

[42] J. Rhys, A selection problem of shared fixed costs and network flows, Management Sci. 17 (1970) 200–207.

[43] C.D. Simone, The cut polytope and the Boolean quadric polytope, Discrete Math. 79 (1989) 71–75.

[44] J. Sturm, Sedumi 1.05, Matlab toolbox for solving optimization problems over symmetric cones, ⟨http://fewcal.kub.nl/sturm/software/sedumi.html⟩, 2002.

[45] R. Weismantel, On the 0/1 knapsack polytope, Math. Programming 77 (1997) 49–68.

[46] C. Witzgall, Mathematical methods of site selection for electronic message system (EMS), Technical Report, NBS Internal Report, 1975.

[47] H. Wolkowicz, R. Saigal, L. Vandenberghe (Eds.), Handbook of Semidefinite Programming, International Series in Operations Research and Management Science, vol. 27, Kluwer Academic Publishers, Dordrecht, 2000.

[48] H. Kellerer, U. Pferschy, D. Pisinger, Knapsack Problems, Springer, Berlin, Germany, 2004.