Annals of Operations Research (2020) 286:173–200
https://doi.org/10.1007/s10479-018-3092-8

S.I.: CLAIO 2016

Strong RLT1 bounds from decomposable Lagrangean relaxation for some quadratic 0–1 optimization problems with linear constraints

Monique Guignard[1]

Published online: 21 November 2018

Abstract

The Reformulation Linearization Technique (RLT) of Sherali and Adams (Manag Sci 32(10):1274–1290, 1986; SIAM J Discrete Math 3(3):411–430, 1990), when applied to a pure 0–1 quadratic optimization problem with linear constraints (P), constructs a hierarchy of LP (i.e., continuous and linear) models of increasing sizes. These provide monotonically improving continuous bounds on the optimal value of (P) as the level, i.e., the stage in the process, increases. When the level reaches the dimension of the original solution space, the last model provides an LP bound equal to the IP optimum. In practice, unfortunately, the problem size increases so rapidly that for large instances, even computing bounds for RLT models of level k (called RLTk) for small k may be challenging. Their size and their complexity increase drastically with k. To our knowledge, only results for bounds of levels 1, 2, and 3 have been reported in the literature. We are proposing, for certain quadratic problem types, a way of producing stronger bounds than continuous RLT1 bounds in a fraction of the time it would take to compute continuous RLT2 bounds. The approach consists in applying a specific decomposable Lagrangean relaxation to a specially constructed RLT1-type linear 0–1 model. If the overall Lagrangean problem does not have the integrality property, and if it can be solved as a 0–1 rather than a continuous problem, one may be able to obtain *0–1 RLT1 bounds* of roughly the same quality as standard *continuous RLT2 bounds*, but in a fraction of the time and with much smaller storage requirements. If one actually decomposes the Lagrangean relaxation model, this two-step procedure, reformulation plus decomposed Lagrangean relaxation, will produce linear 0–1 Lagrangean subproblems with a dimension no larger than that of the original model. We first present numerical results for the Cross-dock Door Assignment Problem, a special case of the Generalized Quadratic Assignment Problem. These show that just solving one Lagrangean relaxation problem in 0–1 variables produces a bound close to or better than the standard continuous RLT2 bound (when available) but much faster, especially for larger instances, even if one does not actually decompose the Lagrangean problem. We then present numerical results for the 0–1 quadratic knapsack problem, for which no RLT2 bounds are available to our knowledge, but we show that solving an initial Lagrangean relaxation of a specific 0–1 RLT1 decomposable model drastically improves the quality of the bounds. In both cases, solving the fully decomposed rather than

✉ Monique Guignard
guignard_monique@yahoo.fr

Extended author information available on the last page of the article

the decomposable Lagrangean problem to optimality will make it feasible to compute such bounds for instances much too large for computing the standard continuous RLT2 bounds.

Keywords Generalized Quadratic Assignment Problem · Crossdock Door Assignment Problem · Quadratic knapsack problem · RLT bounds · Lagrangean relaxation · Integer Linearization Property

1 Notation

For an optimization problem (P), v(P) will denote the optimal value if it exists, and, by abuse of language, +/− ∞ if problem (P) is infeasible or unbounded (depending on whether it is a minimization or a maximization problem).

Names in capital letters may refer to a method or a property, for instance LR for **L**agrangean **R**elaxation, LP for **L**inear **P**rogramming, LD for Lagrangean decomposition, IP for **I**nteger **P**rogramming, RLT for **R**eformulation **L**inearization **T**echnique, BB for Branch-and-Bound. They may also refer to a specific problem type.

Names used in the text are: LP (linear programming problem), IP (Integer Programming Problem), AP (Assignment Problem), QAP (Quadratic Assignment Problem), GAP (Generalized Assignment Problem). GQAP (Generalized Quadratic Assignment Problem), CDAP (Crossdock Door Assignment Problem), KP (0–1 Knapsack Problem), QKP (Quadratic 0–1 Knapsack Problem).

2 Introduction

This paper presents an approach for computing, more efficiently and more accurately than has been done so far, improved bounds on the optimal value of some pure 0–1 quadratic optimization problems with linear constraints. Such problems are often not convex, which makes them even more difficult to solve in integers. They may become more tractable, though, after being linearized (and thus convexified) into equivalent 0–1 problems. Linearization can be achieved in a number of ways, mostly based on similar principles, involving products of the original constraints by each 0–1 variable or its complement. The two key linearization steps are then, for each new constraint, (1) to replace the square of a 0–1 variable by the variable itself, and (2) to replace each product of two distinct 0–1 variables by a new variable. Step (1) may already strengthen the model by using the fact that the original variables are binary. Step (2) introduces new variables, and one needs to enter additional (linear) information to maintain the equivalence with the original quadratic model. Indeed, given two 0–1 variables x and y, defining a new 0–1 variable v as the product of x by y is a nonlinear equation $v = x.y$ that cannot be used as such in a linear model. Fortet (1959) and later McCormick (1976) proposed sets of linear constraints that can be used to force the new binary variable v to be equal to the product of the two binary variables x and y, i.e., to be 0 unless both x and y are 1. Later approaches iterate on that process. They construct sequences of larger and larger models, equivalent in 0–1 variables, and producing increasingly tighter LP bounds. One of these linearization methods is called RLT, for **R**eformulation **L**inearization **T**echnique (Adams and Sherali 1986; Sherali and Adams 1990). They proved that when the level (i.e., iteration number) reaches the number of 0–1 variables of the original problem, the optimal value of the last linear programming problem is equal to the optimal value of the original quadratic 0–1 problem. In practice, however, the problem size increases so rapidly that even for moderate size instances, computing low level *continuous* bounds can already be computationally expensive. If the original model has n 0–1 variables x_i, $i \in \mathrm{I}$, $|\mathrm{I}| = n$, the RLT1[1] model, i.e., the RLT model generated at level 1, is an LP with n^2 additional variables, say, v_{ij}, representing the product of x_i by x_j. Then RLT2 has n^3 additional variables w_{ijk}, representing the product of v_{ij} by x_k, and so on. RLTk produces tighter LP bounds than RLT$(k-1)$, for all $k \leq n$, but at the cost of a large increase in computational and storage cost created by the addition of n^{k+1} additional variables. At this point in time, to our knowledge, only RLT bounds of levels 1, 2 and 3 have been attempted (see for instance Hahn et al. 2012 for RLT3).

The paper is based on two ideas that can be used to simplify the computation and increase the strength of RLT1 bounds. We often refer to the RLT1 models described in the paper as RLT1-type models as a reminder that they may not be standard RLT1's as defined in Sherali and Adams. Similar ideas have appeared to some extent in the literature, see for instance Adams and Johnson (1994) for the QAP, and Caprara et al. (1999) for the QKP. First, using Lagrangean relaxation[2] of RLT1 models rather that LP relaxation, treating them as *integer* models rather than as LPs, may produce improved bounds similar in quality to continuous RLT2 bounds. Second, we suggest constructing specific types of RLT1 models following relatively simple rules. The goal is to create a Lagrangean relaxation that is not only strong, but also *decomposable* into small subproblems, using a property called ILP, for Integer Linearization Property (Geoffrion 1974; Geoffrion and McBride 1978; see also Guignard 2003). The end result when fully implemented is (a) improved bounds with (b) a much smaller computational and storage burden. A small computational experiment for the GQAP, and a more extensive one on the CDAP and the QKP confirm the feasibility and efficiency of the approach, even without a full implementation of the decomposition or full optimization of the Lagrangean dual. Clearly, beyond a certain size, decomposition will be the only way to generate these tighter bounds quickly and with reduced space requirements.

The paper is concentrating on creating RLT1-type models that lend themselves to decomposition and produce strong bounds by Lagrangean relaxation. It presents computational evidence of the strength of the bounds and the possibility of obtaining them with much reduced computational time and storage space for two difficult quadratic 0–1 problem types. The article is organized as follows. Section 3 describes in general terms the steps followed and the reasons for the various methodological choices. Section 4 describes in mathematical terms the ILP and shows how it is used in our general Lagrangean relaxation scheme. Section 5 specializes the above discussion to the GQAP, Sect. 6 to the CDAP and Sect. 7 to the QKP. Section 8 concludes and suggests possible extensions.

3 General methodology

Let us briefly explain the scheme advocated in this paper. Assume, as above, that in the quadratic model under consideration, the original 0–1 variables are x_i, $i \in \mathrm{I}$, and that there are p equality constraints and q inequality constraints, p and q finite nonnegative integers,

[1] In the RLT theory, problems RLTk are normally meant to be LP problems. In this paper, however, they may be either continuous or 0–1 problems and to avoid confusion, we will refer to them as continuous RLTk or 0–1 RLTk. If we talk about just RLTk, we mean 0–1 RLTk, because this is what we are advocating.

[2] In the paper, the term "Lagrangean relaxation problem" always means that it is solved as an integer programming problem, not as an LP. If it is not the case, we will explicitly talk of a *continuous* Lagrangean relaxation, or a *continuous* Lagrangean bound.

not both 0. In this model's constraints, we assume that all variables are in the left hand side and constants in the right hand side. At level 1, in the standard RLT scheme, one multiplies, both on the right and on the left, each of the p equality constraint terms by x_i and each of the q inequality constraint terms by both x_i and $1 - x_i$, for every i. One then introduces new distinct variables v_{ij} and v_{ji} that will replace respectively the left and right products $x_i x_j$ and $x_j x_i$ of x_j by x_i to obtain linear objective function and constraints. One could replace v_{ii} by x_i, and eliminate obviously redundant constraints, if any. One could also make use of the fact that $x_i x_j = x_j x_i$ (see Padberg and Rijal 1996) to replace v_{ji} by v_{ij}, or vice versa. The equalities $v_{ji} = v_{ij}$, for all $i < j$, to avoid duplicate constraints, are called the symmetry constraints. But one has to do it selectively, according to a specific goal: one wants to

Produce a relaxation bound stronger than the continuous relaxation bound

The first contribution of the paper is to show that for at least some quadratic 0–1 problem types, it is possible to create a 0–1 RLT1-like model producing a Lagrangean bound tighter than the standard continuous RLT1 model. Geoffrion proved that the Lagrangean relaxation bound is equal to the optimum of the original objective function over the intersection of the polyhedron of the relaxed constraints and the convex hull of the integer solutions of the constraints kept in the relaxed model (Geoffrion 1974). If the polyhedron of the kept constraints (the kept polyhedron, for short) has all corner points integer, one says that the problem has the Integrality Property, and then the Lagrangean bound is automatically equal to the LP bound, i.e., there is no bound improvement possible. If the kept polyhedron has at least some fractional vertices, the Lagrangean bound may be tighter than the LP bound, at least for some objective functions. Keeping—or adding—more constraints in the LR model may result in providing tighter bounds. To summarize, in order to obtain a stronger bound than the LP bound, the following condition is necessary (but not sufficient):

Condition 1: The Lagrangean problem should not have the integrality property

This will depend on the underlying problem structure. For instance, we do not think it possible to come up with such a scheme for the QAP problem, given its underlying assignment problem structure. Adams and Johnson in their paper on the QAP (Adams and Johnson 1994) used a similar scheme, however due to the nature of the assignment problem, their Lagrangean problem decomposed into a family of small linear assignment problems *with the integrality property*, and their optimal bound was equal to the continuous RLT1 bound. We will present examples, and some computational results, showing that for some other well-known difficult quadratic 0–1 problem types, at least one RLT1-like model exists that produces stronger bounds than the standard continuous RLT1 bound. Notice that condition (1) does not reduce the memory requirements over the standard RLT1 model. The advantage, though, is that there may be such a substantial improvement over the standard RLT1 bound that it produces RLT2 quality bounds with much smaller storage requirements.

The second objective is to

Reduce the computational and storage burden

We will show that one can *also* substantially reduce the computational burden if, in addition, the RLT1 Lagrangean problem is decomposable into smaller, relatively easy to solve subproblems. We assumed that initially, all variable terms, and only variable terms, are in the left-hand side of the model constraints, and the constant terms are in the right-hand side. Before the construction of the RLT1-like model, which involves adding new constraints (we will refer to these as the "added" constraints), no term is moved to the other side of the equality or inequality sign. Each added constraint comes from an original constraint in which each term is multiplied (on the right or on the left) by one of the original variables, say x_i, or its complement. The "added" constraint obtained this way may therefore contain only variables x_i, v_{ij} and v_{ji}, for a given i and all j (see the beginning of Sect. 1 for the notation). The right hand side of an "added" constraint is equal to the original right-hand side coefficient, i.e., a scalar, multiplied by x_i or $1 - x_i$. The original constraints form a submodel of the RLT1-like model. The "added" constraints part is what creates a substantial increase in size and computational complexity. One possible way to make the solution of the Lagrangean problem more manageable is to

Create a decomposable model

One possible approach to create a decomposition is to create one submodel for each original variable x_i. This is the approach we are describing below.

Two comments will shed some light on the road leading to a decomposable Lagrangean relaxation. First, the symmetry constraints, which include both one v_{ij} and its symmetric v_{ji}, are the first constraints that need to be dualized, as they would link indirectly an x_i and an x_j, and this would prevent the complete model from decomposing into one subproblem per variable x. Notice that the symmetry constraint is not a copy constraint, in the sense of Lagrangean decomposition (Guignard and Kim 1987). All variables are part of the *original* RLT-like model, they are not newly created copies of (some of) the original variables. What we are proposing is therefore *not* a Lagrangean decomposition (or variable splitting), but a standard Lagrangean relaxation.

Secondly, if the model contains inequality constraints, multiplying them by $1 - x_i$ will introduce an additional *constant* in the right hand side (instead of only an x_i term), and the "1" in $1 - x_i$, when multiplied in the left hand side by a variable x_j, for j different from i, will keep that variable x_j in the constraint, in addition to x_i. This will prevent the Lagrangean relaxation of the "added" part of the RLT1-like model from decomposing into a separate subproblem for each x_i variable, since two different x variables would appear in the same constraint.

The following condition is one way (but maybe not the only one) that will allow the chosen RLT1 to yield a Lagrangean relaxation that reduces storage and/or computational burden by decomposing the added constraints into one subproblem per original variable x.

Condition 2: After dualizing the symmetry constraints, the Lagrangean problem should decompose into two subproblems, the first subproblem with all the original constraints, containing only variables x_i, and the second subproblem with all added constraints, each containing only one x_i and its associated v_{ij}'s, for some *i*. In that case, the second subproblem itself decomposes into one subproblem for each x_i.

In what follows, we assume that the subproblem associated with x_i has optimal value 0 if x_i is 0, basically because it forces all associated v_{ij} to be 0, and it will need to be solved to optimality over the v_{ij}'s if x_i is 1.

If both Condition 1 and Condition 2 hold, then the Integer Linearization Property holds (see the explanation in Sect. 4), and one can solve the "added" subproblems separately, or, more precisely, one can solve n smaller 0–1 subproblems, one for each x_i at 1, and optimal value, say, β_i, plus one additional 0–1 linear subproblem over all x's, consisting of all the original constraints and an objective function that includes the value β_i in the coefficient of x_i for all i (see Sect. 4 for details). If the integrality property does not hold, one may obtain a stronger bound that the standard continuous RLT1 bound. The savings in computational complexity (time and storage), if one actually implements the decomposition, may be substantial, because it is usually much more efficient to solve k optimization problems in m 0–1 variables each, than one large combinatorial problem in $k \times m$ 0–1 variables.

The numerical experiments presented in Sect. 6 for the CDAP already show that using a decomposable Lagrangean relaxation, even *without* actually implementing the decomposition, and *without* iterating over the Lagrangean multipliers in the solution of the Lagrangean relaxation, one may obtain integer RLT1 bounds similar in quality to continuous RLT2 bounds in much less time than by solving RLT2 directly by dual ascent. In other words, even without exploiting the possibility of decomposing the integer subproblems in the v_{ij} variables, one already obtains a considerable reduction in computing time simply because one solves (in integers) Lagrangean problems of size n^2 to obtain RLT1 bounds of a quality similar to previously available bounds obtained by solving RLT2 problems of size n^3.

We first explain more formally the proposed scheme in Sect. 4. In Sect. 5 and beyond, we propose to consider difficult quadratic optimization problem types, other than the QAP, for which one can write some RLT1 model (a) whose Lagrangean relaxation obtained by dualizing the symmetry constraints does not have the Integrality Property, and (b) which could be *decomposed* into a family of small 0–1 problems that could be solved easily *in integers* when the size of the problem is such that it becomes difficult to solve them as a whole.

4 The Integer Linearization Property (ILP) applied to quadratic 0–1 problems

4.1 What is the ILP?

The ILP was presented without a name in (Geoffrion 1974), and used very effectively in (Geoffrion and McBride 1978). The idea is the following. For an optimization problem with a single parameter α between 0 and 1 in the right hand side of the constraints, parametric programming would normally represent the optimal value as a function $f(\alpha)$ of α in the interval [0, 1]. However, if α can take only value 0 or value 1, and if the optimum is equal to 0 when α is 0, given that the inside part of the curve, for $0<\alpha<1$, is irrelevant, one can replace it by a straight line joining the origin and the point $[1, f(1)]$ on the curve (Geoffrion 1974; Geoffrion and McBride 1978). In (Guignard 2003), this property was named Integer Linearization Property (or ILP), since one can linearize the curve of the optimal value between its end points. More accurately we now call it "the 0–1 ILP," or "the ILP" for short.

This property is useful for some special forms of the Lagrangean function, as we will see in the next subsection. It leads to a Lagrangean relaxation problem that decomposes into smaller subproblems.

4.2 Using the 0–1 ILP in Lagrangean relaxation

Decomposable here will not mean, as is usually the case, that the Lagrangean relaxation model simply decomposes into, say, p independent but similar subproblems for some integer p. A standard example of this would be the case of a 0–1 loading problem with weight and volume constraints (numbered 1 and 2 in the model below), to which one would apply Lagrangean decomposition (Guignard and Kim 1987). Suppose the original model reads

$$\text{Max}\left\{\sum_i f_i y_i \text{ s.t. } \sum_i A_{ki} y_i \le b_k,\ k \in \{1,\ 2\},\ \text{and } y_i \in \{0, 1\}, \text{all } i\right\}.$$

One possible Lagrangean decomposition would first create a copy of each variable y_i, call it w_i, say, and after replacing y_i by w_i in the $k = 2$ constraint, one would dualize the constraints $y_i = w_i,\ \forall i$, with Lagrangean multipliers λ_i unrestricted in sign. The resulting Lagrangean problem would decompose as follows:

$$\text{Max}\left\{\sum_i f_i y_i + \sum_i \lambda_i(y_i - w_i) \text{ s.t. } \sum_i A_{1i} y_i \le b_1, \sum_i A_{2i} w_i \le b_2, \quad y_i, w_i \in \{0,1\}, \text{ all } i\right\}$$

$$= \text{Max}\left\{\sum_i (f_i - \lambda_i) y_i \text{ s.t. } \sum_i A_{1i} y_i \le b_1, \text{ and } y_i \in \{0,1\}, \text{ all } i\right\}$$

$$+ \text{Max}\left\{\sum_i \lambda_i w_i \quad \text{s.t. } \sum_i A_{2i} w_i \le b_2, \text{ and } w_i \in \{0,1\}, \text{ all } i\right\}.$$

In this case each subproblem is a 0–1 knapsack problem and the two problems are independent except for sharing the Lagrangean multipliers λ_i. One can also write this in a "variable-splitting" way by simply decomposing the objective function f into the sum of two objective functions f_1 and f_2 to obtain the equivalent formulation

$$\text{Max}\left\{\sum_i f_{1i} y_i \quad \text{s.t. } \sum_i A_{1i} y_i \le b_1, \text{ and } y_i \in \{0,1\}, \text{ all } i\right\}$$

$$+ \text{Max}\left\{\sum_i f_{2i} w_i \quad \text{s.t. } \sum_i A_{2i} w_i \le b_2, \text{ and } w_i \in \{0,1\}, \text{ all } i\right\}.$$

The two subproblems have a very similar structure. One has really split the original constraint set into subsets with a similar structure. This is not the type of decomposition that is meant here. Both types of decomposition may have the same beneficial effect on the solution time because the subproblem sizes are smaller. Here the decomposition occurs only if the model has a special structure, and it requires some re-formulation of the original Lagrangean problem. This is explained in the remainder of this section.

We are assuming now that we have to solve a Lagrangean Relaxation for some special type of quadratic 0–1 optimization problems. Lagrangean problems of interest typically will have the form

$$\text{LR:} \quad \underset{x,y}{\text{Max}} \sum_i L_i(x_i, y_i) \text{ subject to } x_i \in Z_i \cap A_i y_i, \quad y \in P$$

where the component y_i of y is a 0–1 variable for each $i \in$ I, I a finite index set of dimension n, P is the set of 0–1 points in a polyhedron, x_i is an n_i-dimensional vector, where n_i is a positive integer greater than or equal to 1, A_i is a polyhedral set in R^{n_i}, Z_i can be either R^{n_i} or a subset of vectors in R^{n_i} with some specific components either integer or binary. That is, depending on Z_i, there may be integrality conditions on at least some components of $x_i \in Z_i$. We assume that the constraints $x_i \in Z_i \cap A_i y_i$ imply that $x_i = 0$ for $y_i = 0$, and $x_i \in Z_i \cap A_i$ for $y_i = 1$. Finally, we assume that the Max of $L_i(x_i, y_i)$ subject to $x_i \in Z_i \cap A_i y_i$, is 0 for $y_i = 0$ and V_i for $y_i = 1$, $i = 1,..,n_i$.

Without the constraint $y \in P$, LR would decompose into one problem for each index i:

$$\text{S}_i(\text{y}_i): \quad \underset{x_i, y_i}{\text{Max}} \ L_i(x_i, y_i) \text{ subject to } x_i \in Z_i \cap A_i y_i$$

whose optimal value would be 0 for $y_i = 0$ and, say, V_i for $y_i = 1$.

With the constraint $y \in P$, however, the optimum of (LR) can be found by solving the following problem in y:

$$V(y): \quad \text{Max} \sum_i V_i y_i \quad \text{subject to } y \in P,$$

which depends only on the binary variables y_i. Variables x_i have disappeared.

This is an important result: for a Lagrangean problem of this special type, after computing all individual V_i's for all $i \in$ I, the *final* Lagrangean problem reduces to a problem over variables *y only*, thus its size is much smaller than in its original formulation. This argument is what Geoffrion and McBride exploited so successfully in their 1978 paper for Capacitated Facility Location Problems.

4.3 Using the ILP in a Lagrangean relaxation of some 0–1 quadratic IP's

In the following sections, we will review possible uses of (LR) with the (ILP) for some (0–1) quadratic optimization problems with linear constraints. Given what has been said in Sect. 3, in order to be able to use the (ILP) property, it is essential to start from an RLT1-type model whose structure can produce a Lagrangean relaxation model without the Integrality Property, and decomposable. More specifically, it is necessary to be able to extract from all the variables a subset of (0–1) variables, call them y_i, such that

(1) the Lagrangean problem decomposes into one subproblem, say, $S_i(y_i)$, for each y_i,
(2) the variables of $S_i(y_i)$ other than y_i, call them x_i, scalar or multidimensional, will be 0 for $y_i = 0$, and
(3) problem $S_i(y_i)$ will have optimal value 0 for $y_i = 0$, and, say, V_i for $y_i = 1$.

Then the optimal value of LR can be computed simply by solving the linear problem

$$\text{V(y)}: \quad \text{Max} \sum_i V_i \, y_i \text{ subject to } y \in P$$

with $y \in P$ representing the original constraints on y alone.

The advantage of being able to produce a Lagrangean relaxation with the ILP is two-fold:

(a) the subproblems $S_i(y_i)$ will be simpler to solve than LR, because they are smaller, and they might just be LP's.
(b) the final problem V(y) is only over y, so, again, much smaller than LR. If it does not have the Integrality Property, then its integer optimum can be strictly larger than its LP relaxation value. This might actually hold even if it has the Integrality Property, and can be explained by the fact that the integrality of variables y_i has already been exploited in the linearization process, i.e., the fact that y_i can only be 0 or 1 and never strictly between 0 and 1 is used in the decomposition of the Lagrangean problem.

In what follows, we will concentrate on Lagrangean relaxations of RLT1-like models that do not have the Integrality Property, and use the ILP to decompose them so that the scheme can be used for large instances. The emphasis of this paper is in carefully constructing an RLT1-like model, which should yield a strong LR bound *and* be decomposable, if its size requires it, using the ILP. One needs to determine which "added" constraints one should keep in the decomposable Lagrangean problem to make the bound strong, and which need to be ignored, dualized or rewritten because they would not allow the Lagrangean problem to decompose. We will refer to the overall process as RLT1 + LR + ILP decomposition or RLT1 + LR + ILP for short.

We will show for two related assignment-type models that it is possible to obtain bounds strictly better than the optimal continuous RLT1 bounds, often better and always computed much faster than the best known RLT2 bounds, by selecting one specific Lagrangean multiplier, the 0 vector, without even iterating. For these two models, it was nearly impossible to get any type of subgradient method to converge. A separate paper with Jongwoo Park will present an adaptation of the surrogate subgradient method (this could be used to generate even stronger bounds than the initial bounds presented here), together with experiments with larger instances for which RLT2 bounds could not be computed before. We will also consider briefly in Sect. 7 a model with a much simpler structure, the quadratic 0–1 knapsack problem. A similar approach was used in Caprara et al., but they only tested the decomposed version of RLT1 + LR + ILP with continuous variables. We always consider that all variables are 0–1, and we will show that using a 0 Lagrangean multiplier also produces a much improved bound over the standard RLT1 continuous bound, without dramatically increasing solution time. We expect that implementing the fully decomposed version of RLT1 + LR + ILP will be both time saving and bound improving.

In the next section, we will consider a problem related to the QAP, the Generalized Quadratic Assignment Problem, or GQAP. It is probably as difficult as the QAP, if not more, but it has the advantage over the QAP of having a linear version that does *not* have the integrality property, and thus can produce 0–1 RLT1-like models with Lagrangean bounds *strictly better* than *continuous* RLT1 bounds. We will wait until Sect. 5.3, that is until after we will have presented a complete analysis of the desired structure of RLT1 models for the GQAP, to discuss our choice of Lagrangean multiplier values.

5 The Generalized Quadratic Assignment Problem (GQAP)

The GQAP was introduced by Lee and Ma in (2004) and studied early on, in particular in (Kim 2006), (Zhu 2007) and (Pessoa et al. 2010). It corresponds to the Generalized Assignment Problem (GAP) in the same way the QAP corresponds to the Assignment Problem (AP): by having the same constraints but a quadratic rather than a linear objective function. Unlike the AP which has the integrality property and can at the limit be solved as an LP, the GAP is NP-hard. One can construct decomposable Lagrangean Relaxations from RLT1 without the Integrality Property, in other words, the Lagrangean bound can be strictly better than the continuous RLT1 bound for the original 0–1 quadratic problem. In this section and the next one, we will study RLT-like models for the GQAP and for a special case of the GQAP, the CDAP, as well as Lagrangean relaxation schemes capable of yielding strong bounds at low computational and storage cost.

The original idea of the type of approach presented below was proposed in Guignard (2006), and later reproduced with permission by Y.-R. Zhu in Sect. 4.7 of her PhD dissertation (2007). Initial experiments, however, using the subgradient method or column generation were not successful. In Pessoa et al. (2010), a different Lagrangean relaxation scheme was implemented for the GQAP, but as shown at the end of this section, produced a weaker bound than the one proposed here.

The GQAP can be formulated as follows:

$$\text{GQAP:}\quad \underset{x}{\text{Minimize}}\quad Z = \sum_{i,j} B_{ij} \cdot x_{ij} + \sum_{i,j,k,n} C_{ijkn} \cdot x_{ij} \cdot x_{kn} \tag{4-1}$$

subject to

$$\sum_{i=1}^{M} a_{ij} x_{ij} \leq S_j \quad (j = 1, 2, \ldots, N), \tag{4-2}$$

$$\sum_{j=1}^{N} x_{ij} = 1 \quad (i = 1, 2, \ldots, M). \tag{4-3}$$

$$x_{ij} \in \{0, 1\} \quad (i = 1, 2, \ldots, M; j = 1, 2, \ldots, N), \tag{4-4}$$

where

B_{ij} is the linear cost of assigning facility i to location j.
C_{ijkn} is the quadratic cost of assigning facility i to location j and simultaneously facility k to location n.
A special case is when C_{ijkn} is of the form $f_{ik} d_{jn}$.
x_{ij} is 1 iff facility i is assigned to location j.
a_{ij} is the space needed if facility i is located at location j.
S_j is the space available at location j.
M is the number of facilities.
N is the number of locations.

The original quadratic model GQAP can be transformed through the introduction of new variables into a linearized model called LIP similar to the model used by Adams and Johnson (1994) for the QAP. In order to transform GQAP into an equivalent **L**inearized mixed **I**nteger Programming **P**roblem (LIP), let us first define:

$$v_{ijkn} = x_{ij} x_{kn}, \quad \forall (i, j, k, n) \tag{4-5}$$

Relation (4-5) cannot be included in LIP, since it is nonlinear. Instead, in order to maintain the equivalence, we must include some constraints that are implied by it. Notice that, with x binary, if (4-3) holds

$$\sum_{j=1}^{N} x_{ij} = 1 \quad \forall (i = 1, 2, \ldots, M) \tag{4-3}$$

it follows that

$$x_{ij} x_{in} = 0 \tag{4-6}$$

for all j different from n, and if $v_{ijkn} = x_{ij} x_{kn} \ \forall (i, j, k, n)$, then $v_{ijin} = x_{ij} x_{in} = 0, \ \forall (i, j, n)$ with $j \neq n$, while $v_{ijij} = x_{ij} x_{ij} = x_{ij}, \ \forall (i, j)$, for x binary. We will keep this in mind in what follows.
Now, we multiply on the left by x_{ij}, $i \neq k$, both sides of (4-3), written as

$$\sum_{n} x_{kn} = 1, \quad \forall (k), \tag{4-7}$$

and obtain:

$$\sum_{n} v_{ijkn} = x_{ij} \quad \forall (i, j, k), \ i \neq k \tag{4-8}$$

One could also multiply both sides of (4-7) on the right by $x_{ij},\ i \neq k$, and obtain:

$$\sum_n v_{knij} = x_{ij} \quad \forall (k, i, j),\ i \neq k \tag{4-9}$$

In addition to constraints (4-8) and (4-9) in LIP, given that $x_{ij}x_{kn} = x_{kn}x_{ij}, \forall(i, j, k, n)$, as pointed out in Padberg and Rijal (1996) for the QAP, one also has the implied symmetry constraint

$$v_{ijkn} = v_{knij} \quad \forall (i,\ j, k, n). \tag{4-10}$$

In fact, one can see that it is sufficient to keep either (4-8) and (4-10) [then (4-9) follows], or (4-9) and (4-10) [then (4-8) follows] to get a linear model equivalent to GQAP.

We also require that v_{ijkn} be nonnegative:

$$v_{ijkn} \geq 0 \quad \forall (i, j, k, n),\ i \neq k. \tag{4-11}$$

Using implications of (4-4) and (4-5), we can rewrite the objective function (4-1) of GQAP as

$$\sum_{j,n,i \neq k} C_{ijkn} v_{ijkn} + \sum_{i,j} (C_{ijij} + B_{ij}) x_{ij}. \tag{4-12}$$

Here we choose to work with the following LIP formulation, consisting of all equations from (4-2) through (4-12), except for (4-5) and (4-9):

$$\text{LIP:} \quad \text{Minimize :} \quad \sum_{j,n,i \neq k} C_{ijkn} v_{ijkn} + \sum_{i,j} (C_{ijij} + B_{ij}) x_{ij} \tag{4-12}$$

subject to

$$\sum_i a_{ij} x_{ij} \leq S_j \quad \forall (j) \tag{4-2}$$

$$\sum_j x_{ij} = 1 \quad \forall (i) \tag{4-3}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \tag{4-4}$$

$$\sum_n v_{ijkn} = x_{ij} \quad \forall (i, j, k),\ i \neq k \tag{4-8}$$

$$v_{ijkn} = v_{knij} \quad \forall (i, j, k, n),\ i < k \tag{4-10}$$

$$v_{ijkn} \geq 0 \quad \forall (i, j, k, n),\ i \neq k \tag{4-11}$$

With the x_{ij} variables binary, one can prove that the linear mixed-integer model LIP is equivalent to GQAP. It is clear that (4-8) and (4-10) imply (4-9), so (4-9) was ignored. We will refer to that form of the problem as a terse RLT1 model. A full sized RLT model would also include products of the inequality (4-2) by x_{ij} or its complement. Yet they are not necessary to prove the equivalence with the original model, as shown below.

The following theorem is adapted from (Zhu 2007, Sect. 4.1). Zhu's model contains an additional set of RLT constraints.

Property 4.1 *Problems (*GQAP*) and (*LIP*) are equivalent in the following sense:*

A. *Given any feasible solution x of* GQAP*, there exists a feasible solution (x, v) in (*LIP*) with the same objective value.*

B. *Conversely, given any feasible solution (x, v) of (*LIP*), the corresponding solution x is feasible in (*GQAP*) with the same objective value.*

Proof (A) For a given x, let $v_{ijkn} = x_{ij}x_{kn}$, it is trivial to show that (x, v) is a feasible solution of LIP and the objective function values of GQAP and LIP match as long as (4-2), (4-3) and (4-4) are part of LIP.

(B) Conversely, to show the other direction of the equivalence, given that LIP contains the constraints of GQAP with variables x, we have to show that a feasible solution (x, v) of LIP satisfies $v_{ijkn} = x_{ij}x_{kn}, \forall(i, j, k, n), i < k$, which guarantees that the objective values of LIP and GQAP are equal. In fact, given that the x variables are restricted to be 0 or 1, it suffices to show that if (x, v) is feasible to LIP, v_{ijkn} is 0 unless x_{ij} and x_{kn} are both equal to 1, in which case it is also 1.

- If $x_{ij} = 0$ for given i and j, then (4-8) and (4-11) together imply $v_{ijkn} = 0, \forall n, \forall k \neq i$.

If $x_{kn} = 0$ for given k and n, then (4-8) rewritten as

$$\sum_j v_{knij} = x_{kn} \quad \forall(k, n, i),\ k \neq i$$

and (4-11) together imply

$$v_{knij} = 0, \quad \forall j, \forall k \neq i.$$

By the symmetry constraint (4-10), one can therefore say that for any (i, j, k, n), with $i \neq k$, $v_{i,j,k,n}$ is 0 whenever either x_{ij} or x_{kn} or both are equal to 0.

- Now, one must show that $v_{ijkn} = 1$ if $x_{ij} = x_{kn} = 1$ for given i, j, n, and $k \neq i$.

From (4-3), $\sum_{j'} x_{ij'} = 1\ \forall i$, which implies that if $x_{ij} = 1$ then $x_{ij'} = 0 \quad \forall j' \neq j$.

If $x_{ij} = 1$ then from (4-8), for any $k \neq i$, $\sum_{n'} v_{ij'kn'} = x_{ij'} = 0 \forall j' \neq j$, which in turn implies that for any $k \neq i$, $v_{ij'kn'} = 0 \quad \forall j' \neq j,\ \forall n'$.
Thus, in particular for $n' = n$, we obtain that $v_{ij'kn} = 0\ \forall j' \neq j$.
Therefore, by (4-8) and the symmetry constraint (4-10),

$$x_{kn} = \sum_{j'} v_{knij'} = \sum_{j'} v_{ij'kn} = \sum_{j' \neq j} v_{ij'kn} + v_{ijkn} = v_{ijkn}$$

which implies that if $x_{kn} = 1$ then $v_{ijkn} = 1$. This completes the proof. □

Notice that the equivalence holds in spite of the fact that the inequality constraint (4-2) was not multiplied by an x term. Adding such constraints might tighten the model, but care must be taken to ensure that the ILP still holds. We will consider possible Lagrangean relaxation schemes satisfying the ILP in the next subsections.

Notice that a by-product of the proof is that whenever all x_{ij} are binary in LIP, so are all v_{ijkn}, thus LIP is really a pure integer programming problem. This may have implications when studying geometric interpretations of Lagrangean relaxations of LIP. After dualizing some constraints, the Lagrangean problem should not have the integrality property to allow the LR bound to dominate the LP bound.

5.1 A Tiny Lagrangean Relaxation of a small RLT formulation for the GQAP

Let us now consider relaxations of problem LIP with the potential of yielding strong bounds on the optimal value of (GQAP). First of all, obviously, the continuous relaxation $\overline{\text{LIP}}$ of LIP resulting from replacing the binary conditions on x by nonnegativity conditions [upper bounds are not necessary given (4-3)], will yield a valid lower bound on the optimum of LIP and thus on that of GQAP. A priori, stronger bounds for LIP may be obtained by Lagrangean relaxation if the Lagrangean subproblems do not have the Integrality Property.

Consider dualizing (4-10), the "symmetry" constraint, which only involves the v variables. The Lagrangean subproblem for Lagrangean multiplier $\lambda_{ijkn}, \forall i, j, k, n, i < k$, and with either (4-8) or (4-9), but not both, is, if we choose (4-8):

$$\text{LR}_\lambda: \quad \text{Min} \sum_{j,n,i,k:i\neq k} C_{ijkn} v_{ijkn} + \sum_{i,j} \left(C_{ijij} + B_{ij}\right) x_{ij} + \sum_{j,n,i,k:i<k} \lambda_{ijkn}\left(v_{knij} - v_{ijkn}\right) \tag{4-13}$$

subject to

$$\sum_i a_{ij} x_{ij} \leq S_j \quad \forall(j) \tag{4-2}$$

$$\sum_j x_{ij} = 1 \quad \forall(i) \tag{4-3}$$

$$\sum_n v_{ijkn} = x_{ij} \quad \forall(i, j),\ \forall k \neq i \tag{4-8}$$

$$v_{ijkn} \in \{0, 1\} \quad \forall(i, j, k, n),\ i \neq k \tag{4-11}$$

$$x_{ij} \in \{0, 1\} \quad \forall(i, j) \tag{4-4}$$

Instead of (4-8), one could use

$$\sum_n v_{knij} = x_{ij} \quad \forall(k, j),\ i \neq k \tag{4-9}$$

but not both, as the families of v variables that depend on a given x variable at 1 should not overlap, to allow LIP model decomposition with respect to the x_{ij}'s.

The Lagrangean dual is

$$\text{LR} \quad \underset{\lambda}{\text{Max}} \quad \text{v}(\text{LR}_\lambda),$$

its optimal value is what we call the Lagrangean bound on v(LIP) = v(GQAP). Notice that with the symmetry constraint (4-10) dualized, (4-8) and (4-9) are not equivalent any more, yet, with (4-10), one can keep only one of the two constraints, (4-8) or (4-9), without changing the optimal value. Problem LR_λ, essentially because of constraints (4-2), does not have the Integrality Property (Geoffrion 1974).

We can now make use of the ILP to solve LR_λ easily, if either (4-8) or (4-9) is not present in the model. Ignoring temporarily the constraints that are solely over x_{ij}, i.e., (4-2), (4-3) and (4-4), one can see that the only true constraints remaining are (4-8) or (4-9), and if only one of them is present in the model, the Lagrangean problem does decompose into one problem for each (i, j), i.e., in fact, for each x_{ij}, which plays the role of a right-hand-side parameter. The value of x_{ij} can only be 0 or 1. If it is 0, all associated v_{ijkn} are also zero by (4-8) (resp. all v_{knij} by (4-9)). If it is 1, the non-overlapping subproblem constraints are

$$\sum_n v_{ijkn} = 1 \quad \forall k \neq i \quad \text{if one chooses (4-8)}$$

$$\left(\text{resp. } \sum_n v_{knij} = 1 \quad \forall k \neq i \quad \text{if one chooses (4-9)}\right)$$

and for each k, one solves a multiple choice problem, i.e., one simply selects the v_{ijkn}(resp. the v_{knij}) with the smallest objective function coefficient (notice that one must set exactly one of them equal to 1 for each (i, j, k), given the equality constraint, even if some objective function coefficients is negative, which can happen given that the multipliers λ_{ijkn} may have any sign). Let β_{ij} be the optimal value of this trivial (i, j)th subproblem, then LR_λ is equivalent to

$$(\mathsf{GAP}_\lambda) \quad \text{Minimize:} \quad \sum_{i,j} (C_{ijij} + B_{ij} + \beta_{ij}) x_{ij}$$

$$\text{s.t.} \sum_i a_{ij} x_{ij} \leq S_j \quad \forall (j) \tag{4-2}$$

$$\sum_j x_{ij} = 1 \quad \forall (i) \tag{4-3}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j). \tag{4-4}$$

This is a *linear* generalized assignment problem (a GAP). Thus the solution of LR_λ requires the solution of exactly one GAP, and the final bound may be tighter than the LP bound v($\overline{\mathsf{LIP}}$). Given the current practical limits on GQAP problem sizes, it seems that current BB solvers can solve such linear generalized assignment problems to optimality within a few seconds.

5.2 A stronger Lagrangean relaxation of a larger RLT formulation for the GQAP

For linear problems, as mentioned earlier, the strength of a Lagrangean relaxation is a direct consequence of its geometric interpretation. We propose to strengthen model LIP by adding the following constraints that use the fact that for all i and j, $v_{ijij} = x_{ij}$ and $v_{ijin} = 0$ for $j \neq n$:

$$\sum_k a_{kn} v_{ijkn} \leq S_n x_{ij} \quad \forall (i, j, n),\ k \neq n \tag{4-14}$$

and

$$\sum_{k \neq i} a_{kj} v_{ijkj} \leq \left(S_j - a_{ij}\right) \mathrm{x}_{ij} \quad \forall (i, j) \tag{4-15}$$

as proposed in (Billionnet and Calmels 1996), and used by our co-authors and us in (Pessoa et al. 2010). Adding these constraints to the "kept" polyhedron tightens the relaxation, although it does make it more expensive to solve. Indeed instead of doing simple arithmetic for computing the β_{ij}, we now have to solve one problem in 0–1 variables for each pair (i, j). The ILP still holds, as the v_{ijkl} variables still decompose into non-overlapping families, one for each x_{ij}. In the end, one has to solve a 0–1 linear problem for each pair (i,j), and the same linking model GAP_λ, but with stronger β_{ij} values. Each Lagrangean relaxation submodel will take a little longer to solve than for the "tiny" model. The advantage is strongly improved bounds.

We are presenting one GQAP instance for which results have already appeared in the literature, and compare our results with previously published ones. One of the GQAP instances proposed by Lee and Ma is of dimension 7×16, it has 112 0–1 variables and 23 constraints. It is mentioned in the literature as Lee&Ma 16×7. There is a factor α in the data that can be used as a multiplier in the objective function, it is usually taken as 5, but was tested for our approach in Table 1 both with values 1 and 5. OV is the optimal value and Gap (%) is the relative percentage error between the lower bound LB and OV. The results in the column RLT1 + LR + ILP are taken from (Park 2014) using the subgradient method proposed by Beltran and Heredia in 2005, and the other results from (Pessoa and al. 2010) using the volume algorithm. As can be seen, the duality gap for RLT1 + LR + ILP is 0, the lower bound is exact and proves the optimality of the best feasible solution found. By contrast, other bounds are weaker. The improvement comes from having a tighter set of constraints in the Lagrangean relaxation, still decomposable via ILP, which allows solving the Lagrangean dual by solving a set of subproblems of dimension no larger than that of the original model.

5.3 Lagrangean relaxation with 0 multiplier and the LIP model without (4.10)

The two Lagrangean relaxations discussed above can be applied to the standard GQAP model and to any special case of it, such as the CDAP discussed in Sect. 6. Now that the general scheme RLT + LR + ILP has been described in detail for the GQAP, it may be important to bring up two issues that are important in practice.

5.3.1 The use of valid bounds in practice

The MIP literature is rich in approaches for IP problems. Among them, one finds in particular papers concerned with finding bounds on integer optima. Some concentrate on finding better, that is tighter types of bounds than those previously available, if any, see for instance Guignard and Kim (1987) for Lagrangean decomposition bounds that could be tighter than Lagrangean relaxation bounds, or Adams and Johnson (1994) proposing bounds for the QAP obtained from solving approximately a continuous RLT model. There are other types of papers that do not necessarily search for the strongest possible bound of a particular type, for instance an optimal Lagrangean relaxation bound, but use "some" bound to be used in BB methods for getting proven optimal solutions quickly. See for instance Caprara et al. (1999) for 0–1 quadratic knapsack problems, where the authors compute suboptimal Lagrangean multipliers that they do not update in the BB tree, because the relaxed problems yield already strong bounds and can be solved very quickly. Finally, there are papers that propose to compute tight Lagrangean bounds, possibly together with a heuristic approach, aimed at providing quickly solutions in a environment where speed is critical. This paper is trying to combine aspects of all three approaches. In the context of crossdock door assignment for example (see Sect. 6), one must be able to provide gate assignments quicky. It is then important to be able to come up with a quick solution together with a good bound (see for instance Guignard et al. 2012). The next subsection presents a justification for our selection of a single multiplier vector, in this case the 0 vector, in the rest of the paper, within our RLT + LR + ILP approach.

Table 1 Lower bound values for a small GQAP by different approaches

GQAP instance	Nmber of 0–1 variables	α	OV	RLT1 + LR + ILP (Park 2014)		Transformational only (Pessoa et al. 2010)		Volume algorithm only (Pessoa et al. 2010)		Volume + transformational (Pessoa et al. 2010)	
				LB	Gap (%)	LB	Gap (%)	LB	Gap (%)	LB	Gap (%)
Lee&Ma 16x7	112	1	1,031,150	1,031,150	0	NA	NA	NA	NA	NA	NA
Lee&Ma 16x7	112	5	2,809,870	2,809,870	0	817,500	70.9%	2,673,429.9	4.9%	2,716,593.6	3.3%

5.3.2 The use of 0 multipliers in RLT + LR + ILP[3]

Let us assume that the problem to be solved admits an RLT-like model whose Lagrangean relaxation of the symmetry constraint produces a model with the ILP property. Let us take the example of the GQAP. In the proof of Property 4.1, the symmetry constraint (4.10) plays a role at two places. Let us ignore the $i \neq k$ case, that is specific to this problem type. In part B, without (4.10), one cannot conclude that if (x, v) is feasible to LIP, v_{ijkn} is 0 unless x_{ij} and x_{kn} are both equal to 1, in which case it would also be 1. Still, the LR models of Sects. 5.1 and 5.2 both contain all the constraints on x, plus some constraints linking x and v. The LR relaxations with multipliers equal to 0 have all original constraints, plus all constraints linking x and v, but they do not have the symmetry constraints (4.10). In addition, if the symmetry constraints are close to being satisfied, the Lagrangean objective function value may be close to the original objective function value. The relaxed RLT model with 0 multipliers may thus be close to the original model. We will first explore the effect of this multiplier choice on a special case of the GQAP in the next section.

6 The Crossdock Door Assignment Problem (CDAP)

The CDAP is a special case of the GQAP. It arises in a crossdock, i.e., a building, often rectangular, that receives loaded incoming trucks on one side, the inbound side, and dispatches loaded outgoing trucks on the opposite outbound side. The goods from the incoming trucks are unloaded and sorted according to destinations, carried across the building to the outbound doors where they are loaded into outgoing trucks that will transport them to their final destinations. The origins of the goods can be manufacturer sites and destinations can be distribution centers. The advantage over the "old" system of storing goods in warehouses until they are needed is consolidation of shipments and just-in-time deliveries. The CDAP problem consists in assigning incoming trucks to inbound doors and outgoing trucks to outbound doors so as to minimize the cost of transporting the goods through the crossdock. This cost may be substantial as carts are often pushed manually trough the crossdock and transporting goods from a given incoming truck to outbound trucks may require a number of trips, and this often add a large component to the cost of labor.

This optimization problem can be viewed as a GQAP as follows. Index *i* (resp. *i'*) represents either an incoming (resp. an outgoing) truck that needs to be assigned to **exactly** one inbound door *j* (resp. outbound door *j'*, see constraint (4-3)). Each inbound door *j* (resp. outbound door *j'*) can only handle a certain volume (or weight, or a function of both) S_j (resp. $S_{j'}$) to be downloaded (inbound case) or uploaded (outbound case) during a shift (see constraint (4-2)). The goal is to minimize the total cost of transporting the goods from inbound doors to outbound doors using Manhattan distances. Let $C_{iji'j'}$ be the unit transportation cost, which may depend, among other things, on the distance between doors *j* and *j'* and the kind of goods transported. The cost of carrying across the dock the goods from truck *i* unloaded at door *j*, that must be delivered at destination *i'* by a truck assigned to door *j'* is $C_{iji'j'} x_{ij} x_{i'j'}$ since it depends on simultaneously assigning *i* to *j* and i' to j'. This is a static decision problem that is at the core of a number of dynamic or simulation approaches over time.

The "names" of the instances in the dataset can be a little misleading. The dataset assumes that the problem is symmetric, i.e., there are as many incoming as outgoing trucks, and there are as many receiving as departing doors. So for instance a problem named 20x10S*k* will

[3] We want to thank Jongwoo Park for suggesting this option.

have a total of 20 × 10, so 200, possible assignments *on each side* of the crossdock, and therefore 400 binary assignment variables, since an incoming truck cannot be assigned to an outbound door, nor an outgoing truck to an inbound door. The k in 20x10Sk refers to different ways of generating the coefficients of the instance. The smaller k, the more difficult it is to find feasible 0–1 solutions, which explains some of the variations in runtime for instances of the same size.

The CDAP problem is somewhat more difficult to solve than the general GQAP. One approach would consist of treating it just as a GQAP, with every variable x_{ij} or v_{ijkn} created, giving for instance a very large cost to assigning an incoming (resp. outgoing) truck to an outbound (resp. inbound) door. This was our first approach, but it created numerical instability. We used GAMS' index management facility to restrict the sets of pairs (in the case of x_{ij}) or 4-tuples (in the case of v_{ijkn}) of indices to those that make physical sense.

We demonstrate the potential bound improvement of using RLT + LR + ILP over the continuous RLT1 and RLT2 bounds for the CDAP by providing results for instances from a large dataset called SetA (and SetB for larger instances) that was generated for and used in (Guignard et al. 2012). Table 2 gives results for a few of the smaller, less interesting, instances, and for all instances with between 144 and 500 x variables. The table first lists the standard exact continuous RLT1 bounds computed by LP, in this case by GAMS/gurobi using the barrier option with no crossover, since we have no use for the optimal basis. We chose to compute these RLT1 bounds by linear programming for two reasons: (1) for these relatively small sizes these bounds are easy to compute exactly using LP software, and they are exact, while bounds from the literature, being computed by dual ascent heuristics, cannot be guaranteed to be optimal, and (2) we might want to use the optimal dual variables of the symmetry constraints as initial Lagrangean multipliers in the Lagrangean relaxation. We know by LR theory that the corresponding LR bound, computed by solving optimally an integer programming problem, will be at least as strong as the optimal LP relaxation bound. While most likely not optimal, this "initial" LR value provides a guaranteed lower bound on the possible increase from the continuous RLT1 bound to the optimal RLT1 + LR + ILP bound. The total running time should then be the sum of the time to solve the continuous RLT1 to optimality and the time to solve the Lagrangean relaxed problem. We also computed Lagrangean bounds corresponding to taking $\lambda = 0$. All Lagrangean values for $\lambda = 0$ were stronger and took less computing time than using the optimal LP multipliers (see Sect. 5.3 for a discussion on the potential advantages of using 0 multipliers). This being the case, we are not reporting the values based on the optimal LP multipliers. The bounds given are only iteration #1 of the optimization of the Lagrangean dual, to be used, if desired, as a starting point for the optimization of the Lagrangean bound.

We also list the RLT2 bounds and times provided by Peter M. Hahn.[4] These RLT2 bounds are computed by a sophisticated dual ascent method coded in Fortran using the GQ3AP model (see Zhu 2007, for details), in which bounds were systematically recorded after 20, then 40 iterations.

As expected, as the instance dimension increases, the ratio of the computation times needed by the method of this paper over those needed for computing continuous RLT2 bounds becomes increasingly better, and the approach becomes more and more attractive. As a reminder, if the original instance is of size n, the RLT1 model is of size n^2 and the RLT2 model of size n^3, but our problems in RLT1 + LR are all of size n^2, and if decomposed, one has many submodels, all only of size n. No guaranteed optimal solution is known for

[4] We want to thank Peter M. Hahn for graciously providing us with unpublished continuous RLT2 bounds and running times for the CDAP when available.

Table 2 Comparison of continuous RLT1 and RLT2 bounds with RLT1 + LR + ILP bounds (CDAP)

Instance	Nb. of 0–1 var.	Exact RLT1 conti. bound	Nb of 0–1 var In RLT1	Time in s	Best conti. RLT2 bound by Hahn (20 iter.)	Time in s (Hahn)	Best conti. RLT2 bound by Hahn (40 iter.)	Time in s (Hahn)	Initial Lagrangean bound using 0 dual vars.	Time in s using the non-decomp. model	Optimal value
8x4S5	64	5009	4160	0.36	5075	10	5075	20	5083	1.23	5174
10x4S5	80	6110	6480	0.64	6224	19	6237	39	6235	1.97	6518
10x5S5	100	6188	10,100	3.19	6319	102	6331	218	6273	3.1	6616
11x5S5	110	7240	12,210	3.63	7353	171	7366	353	7376	26.17	7812
12x5S10	120	7623	14,520	4.73	7712	217	7725	452	7637	3.44	7978
12x6S5	144	9853	20,280	7.1	10,061	785	10,077	1609	10,139	46.75	10,891
12x6S10	"	9823	20,880	7.87	10,011	404	10,027	842	9955	6.05	10,362
12x6S15	"	9801	"	6.77	9942	662	9964	1336	9922	1.84	10,362
12x6x20	"	9778	"	6.21	9862	840	9899	1757	9878	1.8	10,456
12x6x30	"	9736	"	6.93	9829	736	9851	1594	9840	5.15	10,228
15x6S5	180	13,074	32,580	16.43	13,067	2089	13,102	4606	13,140	32.20	13,927
15x6S10	"	13,026	"	15.04	13,007	1520	13,032	3183	13,066	8.79	13,803
15x6S15	"	12,986	"	14.23	12,934	677	12,947	1480	13,057	8.61	13,765
15x6S20	"	12,951	"	11.78	12,914	758	12,935	1610	13,045	8.66	13,720
15x6S30	"	12,892	"	11.44	12,805	743	12,817	1573	12,950	2.18	13,567
15x7S5	210	13,773	44,310	32.99	13,799	1875	13,816	3928	13,947	996.79	NA
15x7S10	"	13,709	"	48.17	13,722	2461	13,741	5079	13,815	30.17	NA
15x7S15	"	13,662	"	34.21	13,684	2420	13,702	4850	13,735	4.87	NA

Table 2 continued

Instance	Nb. of 0–1 var.	Exact RLT1 conti. bound	Nb of 0–1 var In RLT1	Time in s	Best conti. RLT2 bound by Hahn (20 iter.)	Time in s (Hahn)	Best conti. RLT2 bound by Hahn (40 iter.)	Time in s (Hahn)	Initial Lagrangean bound using 0 dual vars.	Time in s using the non-decomp. model	Optimal value
15x7S20	"	13,617	"	10.26	13,646	1997	13,669	4062	13,674	6.91	NA
15x7S30	"	13,538	"	10.22	13,530	1795	13,551	3813	13,613	6.05	NA
20x10S5	400	26,293	160,400	63	26,136	64,244	26,180	128,918	26,672	428	NA
20x10S10	"	26,161	"	1488	NA	NA	NA	NA	26,446	188	NA
20x10S15	"	26,043	"	1243	NA	NA	NA	NA	26,300	85.41	NA
20x10S20	"	25,932	"	770	NA	NA	NA	NA	26,161	72.37	NA
20x10S30	"	25,766	"	908	NA	NA	NA	NA	25,953	64	NA
25x10S5	500	43,094	250,500	218	NA	NA	NA	NA	43,140	118.4	NA
25x10S10	"	42,878	"	195	NA	NA	NA	NA	42,973	514	NA
25x10S15	"	42,701	"	470	NA	NA	NA	NA	42,811	438	NA
25x10S20	"	42,547	"	251	NA	NA	NA	NA	42,682	354	NA
25x10S30	500	42,279	"	252	NA	NA	NA	NA	42,435	232	NA

RLT2 bounds from Fortran code (Hahn)
Other bounds using GAMS 25.0.2 with gurobi 7.5.2
All run times on same department server

instances with more than 180 variables x_{ij}, that is, the largest instances solved to optimality are those called SetA_15x6Sxxx. All computations in Table 2 were performed on the same Linux department server. The RLT1 + LR + ILP runs used GAMS 25.0.2, and Gurobi library version 7.5.2, using the barrier option with no crossover. The server's characteristics are as follows: processors 2x Intel Xeon E5-2623 v3 3.0 GHz, storage 300 GB 15 K and 600 GB 10 K, RAM, OS RHEL 6.7.

As the size of the instance keeps increasing, the difference between the running times of the non-decomposed Lagrangean model and the decomposed one can be expected to become larger and larger. We asked Jongwoo Park[5] if he could share a few of the results of his implementation of the full ILP-decomposition of the LR model, also using 0 Lagrangean multipliers. His experiments used gams 24.9.2 and cplex 12.7.1.0, and in order to compare comparable results, Table 3 summarises runs all obtained using cplex on that same department server. It includes a few instances for which an RLT2 bound was not available and/or for which cplex could not get an optimal value for the non-decomposed model within a reasonable time. It shows the optimal value and time obtained by Park using the fully ILP-decomposed model. As expected, for larger instances, the time for solving this decomposed LR model is much smaller than that required for solving the full size decomposable, yet not fully decomposed, model. Why not critical for smaller instances, this is the way to go for a really efficient implementation for large instances. This will be presented in a separate paper with Park concentrating on the implementation of the full RLT1 + LR + ILP decomposition and on the optimization of the Lagrangean bound by adapting the *surrogate subgradient method* (Zhao et al. 1999; Bragin et al. 2015). This method was the only method among many tested that reliably produced improvement of the LR bound over the initial value.

7 The quadratic 0–1 Knapsack Problem

The Quadratic Knapsack Problem (QKP), see for instance Caprara et al. (1999) and Létocart et al. (2012) is a 0–1 knapsack problem with a quadratic objective function:

$$\text{(QKP)} \quad \text{Minimize } Z = \sum_i B_i \cdot x_i + \sum_{i,k} C_{ik} \cdot x_i \cdot x_k \tag{6-1}$$

subject to

$$\sum_i a_i x_i \leq b, \tag{6-2}$$

$$x_i \in \{0, 1\} \quad (i = 1, 2, \ldots, M).$$

Let us replace $x_i x_k$ by v_{ik}, and let us write the RLT1 model:

$$\text{(LIP)}: \quad \text{Minimize } Z = \sum_i B_i \cdot x_i + \sum_{i,k} C_{ik} \cdot v_{ik} \tag{6-11}$$

$$\text{subject to:} \quad \sum_{i \neq k} a_i v_{ik} \leq (b - a_k) x_k, \forall k \tag{6-12}$$

$$\sum_{i \neq k} a_i v_{ki} \leq (b - a_k) x_k, \forall k \tag{6-12'}$$

[5] We want to thank Jongwoo Park for making available for this paper a small sample of his computational experiment on the CDAP using RLT1 + LR + ILP. See Park and Guignard (2018).

Table 3 Comparison of continuous RLT1, RLT2 and RLT1 + LR + ILP bounds without and with full decomposition (CDAP)

Instance	# of 0–1 var.	Exact RLT1 Conti. bound	Number of variables of RLT1	Time in s	Best continuous. RLT2 bound by Hahn (40 iter.)	Time in s	Initial Lagrangean bound using 0 dual vars. w/o ILP decomp.	Time in s	Initial Lagrangean bound using 0 dual vars. with ILP decomp. (Park 2018)	Time in s	Opt. value
15x6S5	180	13,074	32,580	4.0	13,102	4606	13,137	Time limit	13,140	7	13,927
15x7S5	210	13,773	44,310	9.8	13,816	3928	13,931	Time limit	13,947	21	NA
20x10S5	400	26,293	160,400	379	26,180	128,918	26,672	1569	26,672	32	NA
20x10S10	"	26,161	160,000	262	NA	NA	26,446	395	26,446	23	NA
20x10S15	"	26,043	"	227	NA	NA	26,300	339	26,300	20	NA
20x10S20	"	25,932	"	648	NA	NA	26,161	218	26,161	19	NA
20x10x30	"	25,766	"	174	NA	NA	25,953	295	25,953	27	NA

RLT2 bounds from Fortran code (Hahn)
All other bounds using GAMS 24.9.2, Cplex 12.7.1.0
All runs on same department server

$$v_{ik} = v_{ki} \forall(i,k),\ i \neq k \tag{6-13}$$

$$\sum_i a_i x_i \leq b \tag{6-14}$$

$$v_{ik} \in \{0,1\} \quad \forall(i,k),\ i \neq k \tag{6-16}$$

$$x_i \in \{0,1\} \quad \forall(i) \tag{6-17}$$

Ignoring (6-12) or (6-12′), then dualizing the symmetry constraint (6-13) with Lagrangean multiplier λ, one obtains a Lagrangean model LR $_\lambda$. In this paper, we require the Lagrangean model to be decomposable into a submodel for each x_i, and we solve it with 0 multipliers, i.e., in fact, we ignore the symmetry constraint. Contrary to the approach of Caprara, Pisinger and Toth, we require that the Lagrangean problem be solved with all variables x and v constrained to be 0–1. Based on the arguments given in (4-3), we used 0 multipliers and the values obtained are presented in Table 4.

The bound improvement between the continuous RLT1 and the integer Lagrangean relaxation of RLT1 advocated above is substantial, while the increase in computer time is a small multiple. Using the fully ILP-decomposed model instead would most likely yield a time reduction comparable to that seen in the CDAP case. This is left for future research.

8 Conclusion and further research

This paper investigates the theoretical and computational aspects of using RLT1 + LR + ILP decomposition for some 0–1 quadratic models with linear constraints. The results reported in the paper for the CDAP, with bounds comparable to unpublished RLT2 results provided by Peter M. Hahn for some smaller instances, are very promising. For the smallest instances, the time to obtain our RLT1 + LR + ILP bounds may be a little larger than that of the RLT2 method, but as problem dimensions increase, the trend is for the continuous RLT2 code to require a much larger number of variables and as a consequence a much larger computation time and space than the RLT1 + LR + ILP approach, even without using the full ILP-decomposition. The point seems to be that the particular type of RLT1 model chosen, fully decomposable into subproblems of the size of the original model, yields strong Lagrangean relaxation bounds.

The results on the quadratic 0–1 knapsack problem show perhaps even more clearly that spending a little more time than for the continuous RLT bound by solving instead a single non-decomposed Lagrangean subproblem with 0 multipliers already produces much stronger bounds on the integer optimum. Decomposing the Lagrangean model would obviously produce the same bound, but would reduce this time even further, and it may be a good choice for the design of an efficient Branch-and-Bound code.

Another extension of the research will be to construct a Branch-and-Bound code for the CDAP, similar to that of Hahn that produced optimal solutions for the CDAP instances with up to 180 0–1 variables, but using bounds from RLT1 + LR + ILP in 0–1 variables instead of from RLT2 models solved as continuous models.

Another attractive possibility is to re-examine the symmetry constraints in the RLT1 model. As mentioned in Sect. 3.3, these symmetry constraints are *not* copy constraints, in the sense of Lagrangean decomposition (Guignard and Kim 1987). Indeed the symmetric variables are variables of the original model, not artificial copies. But it is possible to introduce copies of the original variables, giving them another name, say, w_{ji}, satisfying the additional constraint

Table 4 QKP

Instance # 0–1 var.	Density (%)	Original number of 0–1 var.	Exact RLT1 continuous bound	Nb. of 0–1 var. in RLT1	Time in s	Initial Lagrangean bound using 0 dual vars.	Time in s using the non-decomposed model	Optimal value
50	25	50	31,115	2500	0.01	23,184	0.03	17,260
	50		41,071		0.02	35,380	0.17	24,804
	75		76,379		0.01	67,806	0.25	55,007
	100		120,979		0.01	117,638	0.19	114,593
100	25	100	118,686	10,000	0.08	80,539	0.06	52,998
	50		197,275		0.07	163,373	0.37	108,439
	75		373,290		0.06	363,595	0.14	354,120
	100		409,931		0.07	382,226	0.48	343,337
150	25	150	279,461	22,500	0.43	196,406	0.08	140,710
	50		565,757		0.41	475,539	0.12	400,743
	75		209,682		0.40	186,630	0.80	137,672
	100		702,740		0.38	650,227	2.79	571,718
200	25	200	229,071	40,000	0.46	177,777	0.79	85,771

Table 4 continued

Instance # 0–1 var.	Density (%)	Original number of 0–1 var.	Exact RLT1 continuous bound	Nb. of 0–1 var. in RLT1	Time in s	Initial Lagrangean bound using 0 dual vars.	Time in s using the non-decomposed model	Optimal value
	50		762,765		0.43	640,714	0.87	427,428
	75		112,497		0.36	99,254	1.15	75,588
	100		787,075		1.32	724,394	11.41	623,145
250	25	250	795,863	62,500	0.53	748,470	0.26	704,437
	50		1,582,432		0.47	1,566,528	0.33	1,550,630
	75		1,912,292		0.53	1,714,501	11.25	1,374,797
	100		3,142,059		0.73	3,089,685	60.75	3,036,287
300	25	300	1,133,276	90,000	0.79	1,124,825	0.39	1,116,454
	50		366,020		0.76	312,198	1.67	194,556
	75		1,391,377		3.24	1,235,209	16.66	924,906
	100		4,082,998		2.16	3,844,934	60.13	3,539,021

Comparison of continuous RLT1 bounds with RLT1 + LR + ILP bounds
Bounds using GAMS 25.0.2 with gurobi 7.5.2
All run times on same department server

$$v_{ij} = w_{ji}, \quad \forall i, j,$$

and to copy carefully selected constraints in which one has replaced v_{ij} by w_{ji}. The original symmetry constraint is erased. Ignoring first the constraints over x alone, one can decompose the remaining problem into one problem per x_i, and remembering that we need only keep the problems with $x_i = 1$, this problem itself decomposes into a subproblem in variables v_{ij} and one in variables w_{ji}. Their optimal values are then used in the coefficient of x_i in the overall objective function subject to the constraints over x. If one chooses to split the constraints, as well as duplicate some, between the v_{ij} and the w_{ji} subproblems, so as to have two subproblems without the integrality property, one may get bounds strictly better than by plain Lagrangean relaxation. One will have to find which value to give to the Lagrangean decomposition multipliers to obtain a strong initial bound. Another computational issue is that if one wants to get the best possible Lagrangean decomposition bound, it is in general difficult to get the subgradient algorithm to converge, most likely because the subgradient has only components $+1$, 0 and -1. It is possible that the surrogate subgradient method (Zhao et al. 1999) will be better behaved.

The approach described above for the GQAP, the CDAP and the QKP should be applicable to other quadratic 0–1 problems with linear constraints. It might not always be possible to find a version of the RLT model with the 0–1 ILP property that is *equivalent* to the original problem in 0–1 variables. What is essential is that (1) this model must be tight enough, and (2) it must contain (2a) the original model in terms of the original variables, and (2b) additional constraints that each have one original 0–1 variable as right hand side, and only associated higher-dimension variables in the left hand side, with no such variable appearing in more than one of these subsets of constraints. The whole Lagrangean problem will then be decomposable into a linear version of the original 0–1 quadratic problem, and families of simple 0–1 subproblems, one per original 0–1 variable. This can mean a significant improvement over the standard continuous RLT1 approach, as one avoids the significant increase in the number of variables and in the size of the RLT model each time one moves up one level in the RLT scheme.

An intriguing question is whether it might be possible to construct a similar approach for RLT2 bounds instead of RLT1 bounds. Can one decompose an RLT2 model into smaller submodels that would be relatively easy to solve and would yield stronger bounds than the continuous RLT2 continuous bounds, and maybe as good as RLT3 bounds? A starting point for this analysis might be Adams et al. (2007).

Future research will also concentrate on identifying other important nonconvex 0–1 quadratic problem types that could benefit from the approach. Computational experiments using the fully decomposed Lagrangean model should tell us how large the models can become before the approach becomes too expensive. Already the decomposable, but not fully decomposed, Lagrangean relaxation models for the RLT1 bound on the CDAP and the QKP provide a significant computational improvement, in bound quality and, at least for larger instances, solution time, over the continuous RLT1 model, and at times, when available, even continuous RLT2 bounds, as shown in the computational results of this paper.

Acknowledgements The author would like to thank the three anonymous referees whose many insightful comments and suggestions helped to significantly improve the organization and contents of the paper.

References

Adams, W. P., Guignard, M., Hahn, P. M., & Hightower, W. L. (2007). A Level-2 Reformulation-Linearization technique bound for the quadratic assignment problem. *European Journal of Operational Research, 180*(3), 983–996.

Adams, W. P., & Johnson, T. A. (1994). Improved linear programming-based lower bounds for quadratic assignment problems. *Discrete Mathematics & Theoretical Computer Science, 16,* 43–77.

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

Beltran, C., & Heredia, F. J. (2005). An effective line search for the subgradient method. *Journal of Optimization Theory and Applications, 125,* 1–18.

Billionnet, A., & Calmels, F. (1996). Linear programming for the 0–1 quadratic knapsack problem. *European Journal of Operational Research, 92,* 310–325.

Bragin, M. A., Luh, P. B., Yan, J. H., Yu, N., & Stern, G. A. (2015). Convergence of the surrogate Lagrangian relaxation method. *Journal of Optimization Theory and Applications, 164,* 173–201.

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

Drezner, Z., Hahn, P., & Taillard, E. (2005). Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods. *Annals of Operations Research, 139,* 65–94.

Fortet, R. (1959). L'Algèbre de Boole et ses applications en recherche opérationnelle. *Cahiers Centre Etudes Rech Opér, 4,* 5–36.

Geoffrion, A. M. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Study, 2,* 82–114.

Geoffrion, A. M., & McBride, R. (1978). Lagrangean relaxation applied to capacitated facility location problems. *AIIE Transactions, 10*(1), 40–47.

Guignard, M. (2003). "Lagrangean relaxation", invited survey. *TOP, 11*(2), 151–228.

Guignard, M. (2006). *RLT1 and LR for the GQAP*. OPIM Department Research Report 06-06-01, the Wharton School, University of Pennsylvania.

Guignard, M., Hahn, P., Pessoa, A. A., & DaSilva, D. C. (2012). Algorithms for the cross-dock door assignment problem. In *Proceedings of 4th international workshop model-based metaheuristics, Angra dos Reis, Brazil* (pp. 1–12).

Guignard, M., & Kim, S. (1987). Lagrangean decomposition: A model yielding stronger bounds. *Mathematical Programming, 39*(2), 215–228.

Hahn, P. M., Kim, B.-J., Guignard, M., Smith, J., & Zhu, Y.-R. (2008). An algorithm for the generalized quadratic assignment problem. *Computational Optimization and Applications, 40,* 351–372.

Hahn, P. M., Zhu, Y.-R., Guignard, M., & Hightower, W. L. (2012). A level-3 reformulation-linearization technique bound for the quadratic assignment problem. *INFORMS Journal on Computing, 24*(2), 202–209.

Kim, B.-J. (2006). *Investigation of methods for solving new classes of quadratic assignment problems (QAPs)*. Doctoral Dissertation, University of Pennsylvania, ESE Department.

Lee, C.-G., & Ma, Z. (2004). *The generalized quadratic assignment problem*. Research Report, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, M5S 3G8, Canada.

Létocart, L., Nagih, A., & Plateau, G. (2012). Reoptimization in Lagrangian methods for the 0–1 quadratic knapsack problem. *Computers & OR, 39*(1), 12–18.

Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P. M., & Querido, T. (2007). A survey for the quadratic assignment problem. *European Journal of Operational Research, 176,* 657–690.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part 1 Convex underestimating problems. *Mathematical Programming, 10,* 147–175.

Padberg, M., & Rijal, M. P. (1996). *Location, scheduling, design and integer programming*. Norwell, MA: Kluwer Academic.

Park, J. (2014). *Generalized quadratic assignment problem: Combining level 1 Lagrangean decomposition and reformulation-linearization technique*. Independent Study Report, University of Pennsylvania, ESE Department.

Park, J., & Guignard, M. (2018). *Computing bounds for CDAP problems using RLT1 + LR + ILP decomposition*. Research Report, OID Department, University of Pennsylvania, 2018.

Pessoa, A. A., Hahn, P. M., Guignard, M., & Zhu, Y.-R. (2010). Algorithms for the generalized quadratic assignment problem combining Lagrangean decomposition and the Reformulation-Linearization Technique. *European Journal of Operational Research, 206*(1), 54–63.

Sherali, H. D., & Adams, W. P. (1990). A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics, 3*(3), 411–430.
Tsui, L. Y., & Chang, C.-H. (1992). An optimal solution to a dock door assignment problem. *Computer and Industrial Engineering, 23*(1), 283–286.
Zhao, X., Luh, P. B., & Wang, J. (1999). Surrogate gradient algorithm for Lagrangian Relaxation. *Journal of Optimization Theory and Applications, 100*(3), 699–712.
Zhu, Y.-R. (2007). *Recent advances and challenges in quadratic assignment and related problems*. Doctoral Dissertation, University of Pennsylvania, ESE Department.

Affiliations

Monique Guignard[1]

[1] OID Department, The Wharton School, University of Pennsylvania, 3730 Walnut Street, Philadelphia, PA 19104, USA