Available online at www.sciencedirect.com

ScienceDirect

European Journal of Operational Research 180 (2007) 983–996

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

Discrete Optimization

A level-2 reformulation–linearization technique bound for the quadratic assignment problem

Warren P. Adams [a], Monique Guignard [b], Peter M. Hahn [c,*], William L. Hightower [d]

[a] *Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-0975, USA*
[b] *OPIM Department, The University of Pennsylvania, Philadelphia, PA 19104-6340, USA*
[c] *Department of Electrical and Systems Engineering, The University of Pennsylvania, Philadelphia, PA 19104-6315, USA*
[d] *Mathematics and Computer Science, High Point University, High Point, NC 27262, USA*

Received 16 October 2003; accepted 27 March 2006
Available online 30 June 2006

Abstract

This paper studies polyhedral methods for the quadratic assignment problem. Bounds on the objective value are obtained using mixed 0–1 linear representations that result from a reformulation–linearization technique (rlt). The rlt provides different "levels" of representations that give increasing strength. Prior studies have shown that even the weakest level-1 form yields very tight bounds, which in turn lead to improved solution methodologies. This paper focuses on implementing level-2. We compare level-2 with level-1 and other bounding mechanisms, in terms of both overall strength and ease of computation. In so doing, we extend earlier work on level-1 by implementing a Lagrangian relaxation that exploits block-diagonal structure present in the constraints. The bounds are embedded within an enumerative algorithm to devise an exact solution strategy. Our computer results are notable, exhibiting a dramatic reduction in nodes examined in the enumerative phase, and allowing for the exact solution of large instances.

Keywords: Combinatorial optimization; Assignment; Branch and bound; Quadratic assignment problem; Reformulation–linearization technique

* Corresponding author. Current address: 2127 Tryon Street, Philadelphia, PA 19146-1228, USA. Tel.: +1 215 546 3413; fax: +1 215 546 4043.

E-mail addresses: wadams@clemson.edu (W.P. Adams), guignard@wharton.upenn.edu (M. Guignard), hahn@seas.upenn.edu (P.M. Hahn), bhightower@linus.highpoint.edu (W.L. Hightower).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2006.03.051

1. Introduction

The standard mathematical formulation of the quadratic assignment problem is as follows

$$\text{QAP}: \quad \min\left\{\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}x_{ij} + \sum_{\substack{i=1\\ i\neq k}}^{n}\sum_{\substack{j=1\\ j\neq l}}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} C_{ijkl}x_{ij}x_{kl} : \mathbf{x}\in\mathbf{X}, \mathbf{x} \text{ binary}\right\}, \tag{1}$$

where

$$\mathbf{x}\in\mathbf{X} \equiv \left\{\mathbf{x} \geqslant \mathbf{0} : \sum_{i=1}^{n} x_{ij} = 1 \quad \text{for } j = 1,\ldots,n; \; \sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } i = 1,\ldots,n\right\}. \tag{2}$$

The problem is so named because the objective is to optimize a quadratic function of binary variables over the assignment polytope **X**. The objective contains no quadratic expressions $x_{ij}x_{kl}$ having $i = k$ or $j = l$ since the **x** binary restrictions force $x_{ij}x_{kl} = x_{ij}$ if $i = k$ and $j = l$, and the **x** binary and assignment restrictions together force $x_{ij}x_{kl} = 0$ otherwise. We use the abbreviations qap and QAP throughout the paper to refer to "quadratic assignment problem" and problem QAP above, respectively. For notational convenience, we henceforth let all summations run from 1 to n unless noted otherwise.

The qap is among the most difficult NP-hard combinatorial optimization problems. In theory, it can be solved by enumerating the n factorial feasible binary solutions, and by selecting one that yields a minimal value. But from a practical point of view, it is extremely challenging, with exact procedures tending to fail for problem sizes of about $n = 25$ to $n = 30$, i.e. for 625–900 variables.

Among exact methods, branch-and-bound approaches have been the most successful. Here, the intent is to implicitly enumerate over the set of solutions, using lower bounds to prune branches of the binary search tree. The key challenge has been to obtain tight bounds that permit effective pruning and that are not too expensive to compute. Interestingly, the qap has proven itself more challenging than other classes of NP hard problems in terms of the size instances that can be solved. A partial explanation is that the majority of test problems suffer from a homogeneous objective function which tends to hurt the pruning process.

Our prior research on the qap has led to computational advances, and pointed the way for the current study. These earlier efforts were based on the application of a reformulation–linearization-technique (rlt) to QAP. The rlt recasts QAP as a mixed 0–1 linear program via two steps. It first *reformulates* the problem by constructing redundant nonlinear restrictions, obtained by multiplying the equality constraints of **X** by product factors of the binary variables. Thereafter, it *linearizes* the objective and constraints by substituting a continuous variable for each distinct nonlinear term. Depending on the product factors used to compute the redundant restrictions, different formulations emerge. The result is an n-level hierarchy of mixed 0–1 linear representations of QAP. Each level of the hierarchy provides a program whose continuous relaxation is at least as tight as the previous level, with the highest level giving a convex hull representation.

The *weakest* level-1 rlt form, which follows from the work of [2,3], was shown [1,14] to subsume and unify alternate linear representations of QAP, and the resulting bounds to dominate the majority of published works in terms of relaxation strength. In addition, this form has a block-diagonal structure [1] that lends itself to efficient solution methods; in particular, to a Lagrangian relaxation with special structure in the subproblem as well as the dualized constraints. (Also see [10] for a different interpretation of this same decomposition and bound.) For QAP of size n, the subproblems consist of $n^2 + 1$ separate linear assignment problems, n^2 of size $n - 1$ and one of size n. The dualized equality constraints essentially set one family of variables equal to another, so that each such restriction has exactly two nonzero entries, one 1 and one -1. The overall approach motivates a monotonic increasing sequence of lower bounds, and is referred to as a *dual ascent strategy*.

Bounds from the level-1 rlt were strategically implemented within enumerative algorithms [11–13], resulting in marked success. Well-known test problems up to size $n = 22$ were solved in [11]. Other problems solved include [12] the size $n = 25$ instance from [16], and the Krarup 30a [13]. References to these classical test cases are found in QAPLIB [7]. The only competitive methods of which we are aware are due to Brixius and Anstreicher [6] and Anstreicher et al. [5], which use convex quadratic programming bounds relaxations as in [4].

Based on our successes with the level-1 rlt representation, we turn attention in this paper to the level-2 form. This program provides even tighter bounds than level-1, but at the price of increased size. The challenge is to take advantage of the additional strength without being hurt by the size. As with its level-1 counterpart, we will show how the level-2 form can be handled via a Lagrangian approach to obtain a subproblem with block-diagonal structure. This time, however, the dualized constraints are much more plentiful, and each subproblem block can itself be decomposed. A more sophisticated approach is needed to handle the nested structure, as well as the complicating constraints.

This effort is a natural extension of our earlier work. We compare the strength of the level-2 bounds with those available from level-1, as well as with other published approaches. We then devise a similar dual ascent strategy as for level-1 to efficiently solve the linear programming relaxation. The resulting bounds are embedded within an enumeration algorithm to solve problems larger than those from level-1.

As the underlying theoretical motivation for this work is the rlt, we provide a brief discussion in the following section, emphasizing the level-1 and 2 forms of QAP. We focus attention in Section 3 on a specially devised algorithm for solving level-2 via a Lagrangian approach. Section 4 gives computational experience. Section 4.1 examines the strength of the lower bounds afforded by level-2, and considers the merits of the Lagrangian approach as opposed to straightforward linear programming solvers. Section 4.2 demonstrates the effectiveness of level-2 within a branch-and-bound algorithm, focusing on CPU execution times and numbers of nodes enumerated within the binary search trees. Section 5 gives a brief summary of our conclusions and discusses ongoing research.

2. Reformulation–linearization technique (rlt) applied to the QAP

The rlt is a methodology for reformulating mixed 0–1 linear and polynomial programs in higher-variable spaces so that tight polyhedral approximations of the convex hull of solutions are obtained. The key operation, introduced in [2,3], is to compute products of binary variables and their complements with problem constraints, and to enforce the identity that $x^2 = x$ for binary x. Significant research has been conducted on rlt methods. We limit attention in this paper to the level-1 [1,2] and level-2 (see p. 104 and 105 of [20]) forms of QAP only, and refer the interested reader to [19–21] for more general and detailed discussions.

Consider problem QAP as presented in Section 1, (1) and (2). The level-1 rlt representation is generated via the following two steps, as in [1].

Construction of the level-1 rlt formulation of QAP

Reformulation: Multiply each of the $2n$ equations and each of the n^2 nonnegativity restrictions $\mathbf{x} \geqslant \mathbf{0}$ defining $\mathbf{X}$ in (2) by each of the n^2 binary variables x_{kl}, and append these new restrictions to (2). When a variable x_{ij} in a given constraint is multiplied by x_{kl}, express the resulting product as $x_{ij}x_{kl}$ in that order. Substitute $x_{kl} = x_{kl}^2$ throughout the constraints. Set $x_{ij}x_{kl} = 0$ if $i = k$ and $j \neq l$ or if $i \neq k$ and $j = l$.

Linearization: Linearize the resulting problem by substituting, for every occurrence of each product $x_{ij}x_{kl}$ with $i \neq k$ and $j \neq l$, the continuous variable y_{ijkl}. Enforce the trivial restrictions that $y_{ijkl} = y_{klij}$ for all (i,j,k,l) with $i < k$ and $j \neq l$.

Upon so doing, the below formulation results. Here, we have abbreviated the summation notation as earlier mentioned.

$$
\begin{aligned}
\text{RLT1:}\quad \min \Bigg\{ & \sum_i \sum_j b_{ij}x_{ij} + \sum_{i\neq k} \sum_{j\neq l} \sum_k \sum_l C_{ijkl}y_{ijkl} \\
& \sum_{i\neq k} y_{ijkl} = x_{kl} \quad \text{for all } (j,k,l),\ j \neq l, && (3)\\
& \sum_{j\neq l} y_{ijkl} = x_{kl} \quad \text{for all } (i,k,l),\ i \neq k, && (4)\\
& y_{ijkl} = y_{klij} \quad \text{for all } (i,j,k,l),\ i < k,\ j \neq l, && (5)\\
& y_{ijkl} \geqslant 0 \quad \text{for all } (i,j,k,l),\ i \neq k,\ j \neq l, \\
& \mathbf{x} \in \mathbf{X},\ \mathbf{x} \text{ binary} \Bigg\}.
\end{aligned}
$$

Two simple operations can be used to reduce the numbers of variables and constraints in RLT1. First, all variables y_{ijkl} with $i > k$ and $j \neq l$ can be eliminated by making the substitutions suggested by (5) throughout the objective function and constraints. Then (5) can be removed. Second, since $\mathbf{X}$ denotes an assignment polytope, any one of the $2n$ equations in (2) of QAP can be removed without changing the feasible set. This reduces the number of equations in $\mathbf{X}$ by one, and reduces the number of equations in either (3) or (4) by $n(n-1)$, since the rlt process effectively multiplies each equation in $\mathbf{X}$ by $n(n-1)$ different variables x_{kl}. Neither operation affects the strength of the continuous relaxation of RLT1, but it is preferable to not use these operations in order to exploit the problem structure.

The level-2 rlt form is constructed in a similar fashion to the level-1 case in that every restriction defining $\mathbf{X}$ is multiplied by each variable x_{kl}. Here, however, every such restriction is also multiplied by each product $x_{kl}x_{pq}$ having $k \neq p$ and $l \neq q$. The specifics are as follows.

Construction of the level-2 rlt formulation of QAP

Reformulation: Multiply each of the $2n$ equations and each of the n^2 nonnegativity restrictions $\mathbf{x} \geqslant \mathbf{0}$ defining $\mathbf{X}$ in (2) by each of the n^2 binary variables x_{kl}, and also by each of the $n^2(n-1)^2$ pairwise products of variables $x_{kl}x_{pq}$ having $k \neq p$ and $l \neq q$. Append these new restrictions to (2). When a variable x_{ij} in a given constraint is multiplied by x_{kl}, express the product as $x_{ij}x_{kl}$, and when it is multiplied by $x_{kl}x_{pq}$, express the resulting product as $x_{ij}x_{kl}x_{pq}$, preserving the order in both cases. Substitute $x_{kl} = x_{kl}^2$ throughout the constraints, reducing expressions of the form $x_{kl}x_{kl}x_{pq}$ and $x_{pq}x_{kl}\,x_{pq}$ to $x_{kl}x_{pq}$. Set $x_{ij}x_{kl} = 0$ if $i = k$ and $j \neq l$ or if $i \neq k$ and $j = l$, in all quadratic and cubic expressions. For the cubic expressions, this gives that $x_{ij}x_{kl}x_{pq} = 0$ if $i = k$ and $j \neq l$, if $i = p$ and $j \neq q$, if $i \neq k$ and $j = l$, or if $i \neq p$ and $j = q$.

Linearization: Linearize the resulting problem by substituting, for every occurrence of each product $x_{ij}x_{kl}$ with $i \neq k$ and $j \neq l$, the continuous variable y_{ijkl}, and for every occurrence of each product $x_{ij}x_{kl}x_{pq}$ with $i \neq k \neq p$ and $j \neq l \neq q$, the continuous variable z_{ijklpq}. Enforce the restrictions that $y_{ijkl} = y_{klij}$ for all (i,j,k,l) with $i < k$ and $j \neq l$, and also enforce the restrictions that $z_{ijklpq} = z_{klijpq} = z_{ijpqkl} = z_{klpqij} = z_{pqijkl} = z_{pqklij}$ for all (i,j,k,l,p,q), $i < k < p$, $j \neq l \neq q$.

The level-2 rlt form is below, where the coefficients D_{ijklpq} found in the objective are all 0. (Note that the below formulation allows nonzero D_{ijklpq} values, so that it generally handles cubic assignment problems.)

$$\text{RLT2:}\quad \min\Bigg\{\sum_i\sum_j b_{ij}x_{ij} + \sum_{i\neq k}\sum_{j\neq l}\sum_k\sum_l C_{ijkl}y_{ijkl} + \sum_{i\neq k,p}\sum_{j\neq l,q}\sum_{k\neq p}\sum_{l\neq q}\sum_p\sum_q D_{ijklpq}z_{ijklpq}$$

$$\sum_{i\neq k,p} z_{ijklpq} = y_{klpq} \quad \text{for all } (j,k,l,p,q),\ \ j\neq l\neq q,\ \ k\neq p, \tag{6}$$

$$\sum_{j\neq l,q} z_{ijklpq} = y_{klpq} \quad \text{for all } (i,k,l,p,q),\ \ i\neq k\neq p,\ \ l\neq q, \tag{7}$$

$$z_{ijklpq} = z_{klijpq} = z_{ijpqkl} = z_{klpqij} = z_{pqijkl} = z_{pqklij} \quad \text{for all } (i,j,k,l,p,q),\ \ i<k<p,\ \ j\neq l\neq q, \tag{8}$$

$$z_{ijklpq} \geqslant 0 \quad \text{for all } (i,j,k,l,p,q),\ \ i\neq k\neq p,\ \ j\neq l\neq q, \tag{9}$$

$$\sum_{i\neq k} y_{ijkl} = x_{kl} \quad \text{for all } (j,k,l),\ \ j\neq l, \tag{10}$$

$$\sum_{j\neq l} y_{ijkl} = x_{kl} \quad \text{for all } (i,k,l),\ \ i\neq k, \tag{11}$$

$$y_{ijkl} = y_{klij} \quad \text{for all } (i,j,k,l),\ \ i<k,\ \ j\neq l, \tag{12}$$

$$y_{ijkl} \geqslant 0 \quad \text{for all } (i,j,k,l),\ \ i\neq k,\ \ j\neq l, \tag{13}$$

$$\mathbf{x}\in\mathbf{X},\ \ \mathbf{x}\ \text{binary}\Bigg\}.$$

As with the case of RLT1, two operations can be used to reduce the size of RLT2 without affecting the relaxation strength. First, only those variables z_{ijklpq} with $i<k<p$ and $j \neq l \neq q$, and only those variables y_{ijkl} with $i<k$ and $j \neq l$ are needed, since Eqs. (8) and (12) allow us to remove all remaining variables. This makes (8) and (12) unnecessary. Second, and noting as before that any one of the $2n$ equality constraints defining $\mathbf{X}$ can be omitted without changing the feasible set, any single equation in $\mathbf{X}$, the $n(n-1)$ constraints of (10), (11) obtained by multiplying that equation by the x_{kl} variables, and the $n(n-1)^2(n-2)$ constraints of (6), (7) obtained by multiplying that equation by the $x_{kl}x_{pq}$ terms can all be removed from RLT2. As with RLT1, we will not perform such operations, but instead exploit a block-diagonal structure present within a Lagrangian subproblem.

Before proceeding to the Lagrangian relaxation in the following section, we briefly compare problems QAP, RLT1, and RLT2. Each of the latter two (linear) forms is equivalent to the first (quadratic) form when the $\mathbf{x}$ binary restrictions are enforced; each linear problem implies that $y_{ijkl}=x_{ij}x_{kl}$, while RLT2 also implies that $z_{ijklpq}=x_{ij}x_{kl}x_{pq}$. Hence, an optimal solution to either RLT1 or RLT2 will yield an optimal solution to QAP. However, when the $\mathbf{x}$ binary restrictions are relaxed in RLT1 and RLT2, the problems are no longer equivalent to QAP, with both providing lower bounds on the optimal objective value. Although the assignment set (2) defining the constraints of QAP has all binary extreme points, a cost of achieving linearity in RLT1 and RLT2 is that the extreme points forfeit this property. This is to be expected since otherwise we would have P = NP.

Continuing with the comparisons, observe that RLT2 simplifies to RLT1 when the variables z_{ijklpq}, and the associated constraints (6)–(9), are eliminated from the problem. This is directly attributable to the manner in which these representations are computed. Hence, bounds from the continuous relaxation of RLT2 are at least as tight as those from RLT1. And, as will be seen in Section 4, RLT2 can give considerably better bounds than RLT1. Even though the D_{ijklpq} objective coefficients are all 0 in RLT2, the z_{ijklpq} variables serve to tighten the relaxation of RLT2 by restricting the values that the y_{ijkl} and x_{ij} can realize.

3. Lagrangian relaxation of RLT2

The Lagrangian approach for RLT2 is in the same spirit as that used in [1] to solve RLT1. The key observation, repeatedly applied to RLT2, is found in the Lemma below.

Lemma. *Consider any feasible and bounded linear program of the form*

$$\text{LP}: \quad \hat{z}=\min\{\mathbf{c}^{\mathrm{t}}\mathbf{x}+\mathbf{g}^{\mathrm{t}}\mathbf{w} : \mathbf{B}\mathbf{w} \geqslant \mathbf{d}x_i \quad \textit{for some chosen } i,\ \mathbf{A}\mathbf{x} \geqslant \mathbf{b}\}, \tag{14}$$

where $\mathbf{B}\mathbf{w} \geqslant \mathbf{d}$ and $\mathbf{A}\mathbf{x} \geqslant \mathbf{b}$ denote feasible and bounded polyhedral sets, with $\mathbf{A}\mathbf{x} \geqslant \mathbf{b}$ enforcing $x_i \geqslant 0$. (The superscript t *is used to denote vector transpose.) Then an optimal solution $(\hat{\mathbf{x}},\hat{\mathbf{w}})$ to LP can be obtained by solving*

$$\bar{z}=\min\{(\mathbf{c}+\Delta\mathbf{e}_i)^{\mathrm{t}}\mathbf{x} : \mathbf{A}\mathbf{x} \geqslant \mathbf{b}\}, \tag{15}$$

where

$$\Delta=\min\{\mathbf{g}^{\mathrm{t}}\mathbf{w} : \mathbf{B}\mathbf{w} \geqslant \mathbf{d}\}, \tag{16}$$

and where $\mathbf{e}_i$ is the unit column vector having a 1 *in position i and zeroes elsewhere. Here, $\hat{\mathbf{x}}=\bar{\mathbf{x}}$ and $\hat{\mathbf{w}}=\bar{\mathbf{w}}\bar{x}_i$ with $\bar{\mathbf{x}}$ solving (15) and $\bar{\mathbf{w}}$ solving (16), so that $\hat{z}=\bar{z}$.*

Proof. The solution $(\hat{\mathbf{x}},\hat{\mathbf{w}})$ with $\hat{\mathbf{x}}=\bar{\mathbf{x}}$ and $\hat{\mathbf{w}}=\bar{\mathbf{w}}\bar{x}_i$ is primal feasible to LP because $\mathbf{A}\bar{\mathbf{x}} \geqslant \mathbf{b}$ by (15) and $\mathbf{B}\bar{\mathbf{w}} \geqslant \mathbf{d}$ by (16), with $\mathbf{B}\hat{\mathbf{w}} \geqslant \mathbf{d}\hat{x}_i$ since $\hat{x}_i \geqslant 0$. Also, $\hat{z}=\bar{z}$ since $\hat{z}=\mathbf{c}^{\mathrm{t}}\hat{\mathbf{x}}+\mathbf{g}^{\mathrm{t}}\hat{\mathbf{w}}=\mathbf{c}^{\mathrm{t}}\bar{\mathbf{x}}+\mathbf{g}^{\mathrm{t}}\bar{\mathbf{w}}\bar{x}_i=(\mathbf{c}+\Delta\mathbf{e}_i)^{\mathrm{t}}\bar{\mathbf{x}}=\bar{z}$. Thus, the proof reduces to finding a dual feasible solution $(\hat{\mathbf{u}},\hat{\mathbf{v}})$ to LP, with $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ corresponding to $\mathbf{B}\mathbf{w} \geqslant \mathbf{d}x_i$ and $\mathbf{A}\mathbf{x} \geqslant \mathbf{b}$ respectively that, together with $(\hat{\mathbf{x}},\hat{\mathbf{w}})$, satisfy complementary slackness. Toward this end, define $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ to be optimal dual solutions to (16) and (15) respectively. Dual optimality to (16) gives $\mathbf{B}^{\mathrm{t}}\hat{\mathbf{u}}=\mathbf{g}$, $\hat{\mathbf{u}}^{\mathrm{t}}(\mathbf{d}-\mathbf{B}\bar{\mathbf{w}})=0$, $\hat{\mathbf{u}} \geqslant \mathbf{0}$, and $\Delta=\hat{\mathbf{u}}^{\mathrm{t}}\mathbf{d}$ while dual optimality to (15) gives $\mathbf{A}^{\mathrm{t}}\hat{\mathbf{v}}=\mathbf{c}+\Delta\mathbf{e}_i$, $\hat{\mathbf{v}}^{\mathrm{t}}(\mathbf{b}-\mathbf{A}\bar{\mathbf{x}})=0$, and $\hat{\mathbf{v}} \geqslant \mathbf{0}$.The equations $\mathbf{B}^{\mathrm{t}}\hat{\mathbf{u}}=\mathbf{g}$ have $\hat{\mathbf{u}}$ satisfying dual feasibility with respect to the variables $\mathbf{w}$ in (14), while the equations $\mathbf{A}^{\mathrm{t}}\hat{\mathbf{v}}=\mathbf{c}+\Delta\mathbf{e}_i$ and $\Delta=\hat{\mathbf{u}}^{\mathrm{t}}\mathbf{d}$ have $(\hat{\mathbf{u}},\hat{\mathbf{v}})$ satisfying dual feasibility with respect to the variables $\mathbf{x}$ in (14).Also,$\hat{\mathbf{u}} \geqslant \mathbf{0}$ and $\hat{\mathbf{v}} \geqslant \mathbf{0}$. Complementary slackness to (14) is satisfied as $\hat{\mathbf{v}}^{\mathrm{t}}(\mathbf{b}-\mathbf{A}\hat{\mathbf{x}})=\hat{\mathbf{v}}^{\mathrm{t}}(\mathbf{b}-\mathbf{A}\bar{\mathbf{x}})=0$ and $\hat{\mathbf{u}}^{\mathrm{t}}(\mathbf{d}\hat{x}_i-\mathbf{B}\hat{\mathbf{w}})=\hat{\mathbf{u}}^{\mathrm{t}}(\mathbf{d}-\mathbf{B}\bar{\mathbf{w}})\bar{x}_i=0$. This completes the proof. □

The stipulations in the Lemma that $\mathbf{Bw} \geqslant \mathbf{d}$ and $\mathbf{Ax} \geqslant \mathbf{b}$ are feasible and bounded will be satisfied for the sets of interest in this study. However, these stipulations, which give that (15) and (16) are feasible and bounded, are not restrictive. First consider feasibility. Since LP is feasible, (15) is trivially feasible and (16) must be feasible if there exists a solution to (14) with $x_i > 0$. If not, then $x_i = 0$ at every solution to (14), and $\mathbf{g}^t\mathbf{w} = 0$ at every optimal solution since LP is bounded. Then we can let $\mathbf{w} = \mathbf{0}$ so that LP reduces to minimizing $\mathbf{c}^t\mathbf{x}$ over $\mathbf{Ax} \geqslant \mathbf{b}$, $x_i = 0$. Next consider boundedness. There cannot exist an attractive direction to (16) since LP is bounded. To show that (16) is bounded, we need only consider the case where (14) has a solution with $x_i > 0$ since, as noted above, LP otherwise reduces to minimizing $\mathbf{c}^t\mathbf{x}$ over $\mathbf{Ax} \geqslant \mathbf{b}$, $x_i = 0$. But, as suggested in the second line of the above proof, for any optimal $\bar{\mathbf{w}}$ to (16) and any feasible $\tilde{\mathbf{x}}$ to (15), the point $(\tilde{\mathbf{x}}, \tilde{\mathbf{w}})$ with $\tilde{\mathbf{w}} = \bar{\mathbf{w}}\tilde{x}_i$ is feasible to (14) with objective value $\mathbf{c}^t\tilde{\mathbf{x}} + \mathbf{g}^t\tilde{\mathbf{w}} = \mathbf{c}^t\tilde{\mathbf{x}} + \mathbf{g}^t\bar{\mathbf{w}}\tilde{x}_i = (\mathbf{c} + \Delta\mathbf{e}_i)^t\tilde{\mathbf{x}}$. So (15) cannot be unbounded since (14) would be unbounded.

Observe also that any subset of the inequalities $\mathbf{Bw} \geqslant \mathbf{d}x_i$ and $\mathbf{Ax} \geqslant \mathbf{b}$ can be treated as equalities in (14) and the Lemma continues to hold, provided the associated restrictions in (15) and (16) are also treated as equalities. The sets we consider below have equality constraints in nonnegative variables.

Now let us return to problem RLT2. Suppose that Eqs. (8) and (12) are not present in RLT2, having been placed into the objective function using some Lagrangian multipliers. Let $\bar{b}_{ij}$, $\overline{C}_{ijkl}$, and $\overline{D}_{ijklpq}$ denote the objective coefficients on x_{ij}, y_{ijkl}, and z_{ijklpq} respectively, perhaps adjusted from the original objective coefficients b_{ij}, C_{ijkl}, and D_{ijklpq} values by the dualized (8) and (12). Then RLT2 with these adjusted values looks as follows, where we have the scalar $K = 0$.

$$K + \min\left\{\sum_i \sum_j \bar{b}_{ij}x_{ij} + \sum_{i\neq k} \sum_{j\neq l} \sum_k \sum_l \overline{C}_{ijkl}y_{ijkl} + \sum_{i\neq k,p} \sum_{j\neq l,q} \sum_{k\neq p} \sum_{l\neq q} \sum_p \sum_q \overline{D}_{ijklpq}z_{ijklpq} \right.$$
$$\left. : (6), (7), (9), (10), (11), (13), \mathbf{x} \in \mathbf{X}\right\}. \tag{17}$$

We have temporarily relaxed the $\mathbf{x}$ binary restrictions in (17), and will explain in the forthcoming "Observation 1" how these restrictions are readily accommodated. The decomposition of (17) is given in the theorem below.

Theorem. *The optimization problem (17) can be solved via the assignment problem*

$$K + \min\left\{\sum_p \sum_q (\bar{b}_{pq} + \upsilon_{pq})x_{pq} : \mathbf{x} \in \mathbf{X}\right\}, \tag{18}$$

where, for each (p,q) pair, the value υ_{pq} is computed as

$$\upsilon_{pq} = \min\left\{\sum_{k\neq p} \sum_{l\neq q} (\overline{C}_{klpq} + \eta_{klpq})y_{klpq} : \sum_{k\neq p} y_{klpq} = 1 \quad \text{for } l \neq q,\ \sum_{l\neq q} y_{klpq} = 1 \right.$$
$$\left. \text{for } k \neq p, y_{klpq} \geqslant 0 \quad \text{for } k \neq p,\ \ l \neq q\right\} \tag{19}$$

and where, for each (k,l,p,q) having $k \neq p$ and $l \neq q$, the value η_{klpq} is computed as

$$\eta_{klpq} = \min\left\{\sum_{i\neq k,p} \sum_{j\neq l,q} \overline{D}_{ijklpq}z_{ijklpq} : \sum_{i\neq k,p} z_{ijklpq} = 1 \quad \text{for } j \neq l,q,\ \sum_{j\neq l,q} z_{ijklpq} = 1 \quad \text{for } i \neq k,p,\ z_{ijklpq} \geqslant 0 \right.$$
$$\left. \text{for } i \neq k,p,\ \ j \neq l,q\right\}. \tag{20}$$

Proof. For any (k,l,p,q) with $k \neq p$ and $l \neq q$, treat the $2(n-2)$ associated equality constraints in (7) and (8), and the $(n-2)^2$ nonnegativity restrictions in (9) of (17) as $\mathbf{Bw} \geqslant \mathbf{d}x_i$ of (14), with x_i of (14) represented by y_{klpq} and $\mathbf{w}$ of (14) represented by the $(n-2)^2$ variables z_{ijklpq} having $i \neq k \neq p$ and $j \neq l \neq q$, and treat the remaining variables and constraints of (17) as $\mathbf{x}$ and $\mathbf{Ax} \geqslant \mathbf{b}$ respectively. Then apply the Lemma so that the resulting problem of the form (15) contains no z_{ijklpq} term for the chosen (k,l,p,q). Denote Δ of (16) as η_{klpq}, changing the objective coefficient of y_{klpq} from $\overline{C}_{klpq}$ to $\overline{C}_{klpq} + \eta_{klpq}$. Now repeatedly apply the Lemma in the same manner, once for each (k,l,p,q) with $k \neq p$ and $l \neq q$, to reduce (17) to

$$K + \min\left\{\sum_p \sum_q \bar{b}_{pq} x_{pq} + \sum_{k \neq p} \sum_{l \neq q} \sum_p \sum_q (\overline{C}_{klpq} + \eta_{klpq}) y_{klpq} : (11), (12), (14),\ \mathbf{x} \in \mathbf{X}\right\}. \tag{21}$$

(For the remainder of this proof, we assume that all the subscripts i, j, k, and l in constraints (10), (11), and (13) of (21) are replaced by k, l, p, and q respectively, as this change of notation does not affect the feasible region.) Next, repeatedly apply the Lemma to (21), once for each (p,q) in the following manner. For each (p,q), treat the $2(n-1)$ associated equality constraints in (10) and (11), and the $(n-1)^2$ nonnegativity restrictions of (13) as $\mathbf{Bw} \geqslant \mathbf{d}x_i$ of (14), with x_i of (14) represented by x_{pq} and $\mathbf{w}$ of (14) represented by the $(n-1)^2$ variables y_{klpq} having $k \neq p$ and $l \neq q$, and treat the remaining variables and constraints of (18) as $\mathbf{x}$ and $\mathbf{Ax} \geqslant \mathbf{b}$ respectively. Then apply the Lemma, effectively removing all y_{klpq} with $k \neq p$ and $l \neq q$ from the problem. Denote Δ of (16) as υ_{pq} so that the objective coefficient of x_{pq} becomes $\bar{b}_{pq} + \upsilon_{pq}$. Upon repeatedly applying the Lemma, once for each x_{pq}, (21) is reduced to (18) as desired. This completes the proof. □

The Lemma and Theorem show how to decompose RLT2, less restrictions (8) and (12), into the assignment problem (18) of size n, the n^2 assignment problems (19) of size $n-1$, and the $n^2(n-1)^2$ assignment problems (20) of size $n-2$. This decomposition motivates a Lagrangian approach for determining an optimal set of dual multiplier values for (8) and (12), and hence for obtaining the optimal objective value to the continuous relaxation of RLT2. We propose below a dual ascent method, similar to that used in [1,10] for RLT1, which produces a monotonic nondecreasing sequence of lower bounds on this relaxation value to RLT2. Three crucial observations are used.

Observation 1

The Theorem and proof, together with the Lemma, indicate how an optimal (binary) solution to (17) can be obtained via decomposition. Let $\bar{\mathbf{x}}$, $\bar{\mathbf{y}}$, and $\bar{\mathbf{z}}$ denote computed optimal (extreme point) solutions to (18), (19) for all (p,q), and (20) for all (k,l,p,q), $k \neq p$, $l \neq q$, respectively. Then the Lemma gives us that $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ solves (21), where $\hat{\mathbf{x}} = \bar{\mathbf{x}}$ and $\hat{y}_{klpq} = \bar{y}_{klpq}\bar{x}_{pq}$ for all (k,l,p,q) with $k \neq p$ and $l \neq q$. Also by the Lemma, given any $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ optimal to (21), we have $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ with $\tilde{\mathbf{x}} = \bar{\mathbf{x}}$, $\tilde{\mathbf{y}} = \bar{\mathbf{y}}$, and $\tilde{\mathbf{z}}$ defined by $\tilde{z}_{ijklpq} = \bar{z}_{ijklpq}\hat{y}_{klpq}$ for all (i,j,k,l,p,q) with $i \neq k \neq p$ and $j \neq l \neq q$ is optimal to (17). Combining these two statements, we get that an optimal $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ to (17) is defined in terms of $(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{z}})$ as

$$\tilde{\mathbf{x}} = \bar{\mathbf{x}}, \quad \tilde{y}_{klpq} = \bar{y}_{klpq}\bar{x}_{pq} \quad \text{for all } (k,l,p,q),\ k \neq p,\ l \neq q, \quad \text{and}$$
$$\tilde{z}_{ijklpq} = \bar{z}_{ijklpq}\bar{y}_{klpq}\bar{x}_{pq} \quad \text{for all } (i,j,k,l,p,q),\ i \neq k \neq p,\ j \neq l \neq q.$$

Now, since (18)–(20) are assignment problems, the extreme points are binary so that $\bar{\mathbf{x}}$, $\bar{\mathbf{y}}$, and $\bar{\mathbf{z}}$ are binary. This gives $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ as an optimal binary solution to (17).

Observation 2

It is not important in an algorithmic framework to explicitly keep track of the dual values for (8) and (12) each time an adjustment is made. But it is important to record the adjusted coefficients $\overline{C}_{ijkl}$ and $\overline{D}_{ijklpq}$ on y_{ijkl} and z_{ijklpq} of (17) respectively, every time multiples of (8) and (12) are placed into the objective function. Given any fixed (i,j,k,l,p,q) with $i < k < p$ and $j \neq l \neq q$, the effect of placing equations (8) into the objective is to make pairwise adjustments of the six objective coefficients corresponding to the associated $\mathbf{z}$ variables, increasing one such coefficient by the same amount that another is decreased. Similarly, for any fixed (i,j,k,l) with $i < k$ and $j \neq l$, the effect of placing (12) into the objective is to increase the objective coefficient on y_{ijkl} by the same quantity with which the objective on y_{klij} is decreased.

Observation 3

Given a representation of the form (17) to be solved, it is possible to get a tighter lower bound on the optimal objective value to RLT2 than that provided by (17). This is accomplished by taking advantage of the sequential solving of the $n^2(n-1)^2 + n^2 + 1$ assignment problems needed to optimize (17), as described in the Theorem. Specifically, suppose for a chosen (k,l,p,q) with $k \neq p$ and $l \neq q$, that the assignment problem of (20) to compute η_{klpq} is solved and that some variable, say z_{rsklpq}, turns out nonbasic with positive reduced cost ρ. Then the objective coefficient $\overline{D}_{rsklpq}$ can be decreased by ρ without affecting the value η_{klpq}. But by the second observation above, any one of the five associated variables from (8), say z_{klrspq}, can then have its cost $\overline{D}_{klrspq}$ increased by ρ through a dual adjustment to the constraint $z_{rsklpq} = z_{klrspq}$. This leads to a potential increase in η_{rspq}, if this value has not yet been computed. Similar adjustments can be made relative to the variables y_{klpq} and y_{pqkl}, potentially increasing the values υ_{kl} and υ_{pq}.

Our dual ascent strategy is below. Specifics were obtained via computational experiments.

Dual-ascent strategy

Step 1. Initialize (17) by assigning $\overline{D}_{ijklpq} = D_{ijklpq} = 0$ for all (i,j,k,l,p,q) with $i \neq k \neq p$ and $j \neq l \neq q$, $\overline{C}_{ijkl} = C_{ijkl}$ for all (i,j,k,l) with $i \neq k$ and $j \neq l$, $\bar{b}_{ij} = b_{ij}$ for all (i,j), and $K = 0$, where the D_{ijklpq}, C_{ijkl}, and b_{ij} coefficients are taken from RLT2. Set the iteration counter to 0.

Step 2a. For each (k,l), distribute the coefficient $\bar{b}_{kl}$ amongst the $(n-1)^2$ coefficients $\overline{C}_{ijkl}$ for all $i \neq k$ and $j \neq l$ by increasing each such $\overline{C}_{ijkl}$ by $\bar{b}_{kl}/(n-1)$ and decreasing $\bar{b}_{kl}$ to 0. This is equivalent, for each (k,l), to adding $\bar{b}_{kl}/(n-1)$ times each of the $n-1$ equations $\sum_{j \neq l} y_{ijkl} - x_{kl} = 0$ for all $i \neq k$ found in (11) to the objective of (17).

Step 2b. For each (k,l,p,q) with $k \neq l$ and $p \neq q$, distribute the updated coefficient $\overline{C}_{klpq}$ amongst the $(n-2)^2$ coefficients $\overline{D}_{ijklpq}$ for all $i \neq k,p$ and $j \neq l,q$ by increasing each such $\overline{D}_{ijklpq}$ by $\overline{C}_{klpq}/(n-2)$ and decreasing $\overline{C}_{klpq}$ to 0. This is equivalent, for each (k,l,p,q) with $k \neq l$ and $p \neq q$, to adding $\overline{C}_{klpq}/(n-2)$ times each of the $n-2$ equations $\sum_{j \neq l,q} z_{ijklpq} - y_{klpq} = 0$ for all $i \neq k,p$ found in (7) to the objective of (17).

Step 3. Use the Theorem to solve (17) as $n^2(n-1)^2 + n^2 + 1$ assignment problems, exploiting the sequential solving as discussed in Observation 3. This is accomplished as follows.

Step 3a. Solve the $n^2(n-1)^2$ assignment problems (20) of size $n-2$ to obtain $\bar{\mathbf{z}}$ and the values η_{klpq} as follows. Sequentially consider all (k,l,p,q), $k \neq p$, $l \neq q$, beginning with those (k,l,p,q) for which $\overline{C}_{klpq}$ prior to Step 2b was 0. For a selected (k,l,p,q), change the coefficient $\overline{D}_{ijklpq}$ for each $i \neq k,p$ and $j \neq l,q$ to a percentage of the sum of $\overline{D}_{ijklpq}$, $\overline{D}_{klijpq}$, $\overline{D}_{ijpqkl}$, $\overline{D}_{klpqij}$, $\overline{D}_{pqijkl}$, and $\overline{D}_{pqklij}$, and equally adjust the latter five values so that the sum stays constant. (For each (i,j,k,l,p,q) with $i \neq k \neq p$ and $j \neq l \neq q$, we found it most effective to use 35 as the percentage for the first two such z variables encountered in (8), then use 50% for the next two such z variables, and finally 100% for the last two. Observation 2 describes the associated dual adjustments to (8).) Upon solving this assignment problem, place the corresponding equations (6) and (7) into the objective function with the optimal dual multipliers, effectively readjusting the $\overline{D}_{ijklpq}$ values for $i \neq k,p$ and $j \neq l,q$ and increasing $\overline{C}_{klpq}$ by η_{klpq}. Proceed through all such (k,l,p,q) values where $k \neq p$, $l \neq q$.

Step 3b. Solve the n^2 assignment problems (19) of size $n-1$ to obtain $\bar{\mathbf{y}}$ and the values υ_{pq} as follows. Sequentially consider all (p,q), beginning with those for which $\bar{b}_{pq}$ prior to Step 2a was 0. For a selected (p,q), change the coefficient $\overline{C}_{klpq}$ for each $k \neq p$ and $l \neq q$ to a percentage of the sum of $\overline{C}_{klpq}$ and $\overline{C}_{pqkl}$, and then adjust $\overline{C}_{pqkl}$ so that the sum stays constant. (For each (k,l,p,q) with $k \neq p$ and $l \neq q$, we use 70 as the percentage for the first such y variable encountered in (12), and 100% for the second. Observation 2 describes the associated dual adjustments to (12).) Upon solving this assignment problem, place the associated equations (10) and (11) into the objective function with the optimal dual multipliers, effectively readjusting the $\overline{C}_{klpq}$ values for $k \neq p$ and $l \neq q$ and increasing $\bar{b}_{pq}$ by υ_{pq}. Proceed through all such (p,q) values where $k \neq p$ and $l \neq q$.

Step 3c. Solve the assignment problem (18) of size n to obtain $\bar{\mathbf{x}}$. Upon so doing, place the equality constraints of $\mathbf{X}$ into the objective function with the optimal dual multipliers, adjusting the values of $\bar{b}_{pq}$ and the scalar K. Here, K is increased by the nonnegative objective value to the minimization problem of (18). Proceed to Step 4.

Step 4. If the binary optimal solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$, computed as in Observation 1, to (17) is feasible to RLT2, i.e., if it satisfies (8) and (12), stop with $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ optimal to problem QAP. If it is not feasible to RLT2, stop if some predetermined number of iterations has been performed. Otherwise, increase the iteration counter by 1 and return to Step 2a.

Four remarks are warranted. First, there are choices in the implementation of the algorithm, including the sequence in which the assignment problems are solved and the assignment routines employed. We found that the primal–dual Hungarian algorithm produces better results than simplex-based approaches. The reason is presently unknown, though we conjecture that the dual solutions obtained via the Hungarian approach are preferable in the given context to those provided by the network simplex method. Also, a dual feasible solution is readily available at each step since all objective coefficients are nonnegative, providing an advanced start. Second, the algorithm produces a nondecreasing sequence of lower bounds since Step 1 is entered with all variables having nonnegative reduced costs. Third, Steps 3a and 3b begin with variables having reduced costs of 0 because adjustments to these values are most important in increasing the bounds. If, for example, all reduced costs were adjusted to be positive at some iteration, then the bound is assured of increasing in the next iteration. Finally, although there is no established theoretical convergence for this algorithm, it significantly outperforms standard subgradient approaches in terms of bound quality.

Bound performance is examined in the next section.

4. Computational experience

Problem RLT1 provides tight lower bounds on the qap [1,10], and problem RLT2 provides theoretically superior bounds to RLT1. But there are two important questions relative to RLT2. First, how useful is this formulation in efficiently providing tight bounds? Second, how will it perform when placed within a branch-and-bound routine for optimally solving the qap? These two questions are addressed in Sections 4.1 and 4.2, respectively.

4.1. Lower bound provided by RLT2

There are two key issues regarding the efficiency of RLT2 in providing tight bounds. The first is the precise strength, as opposed to RLT1 and other bounding mechanisms. The second is the computational effort required to solve this large form, particularly when using the dual ascent strategy of the previous section in lieu of other approaches.

To address the issue of bound strength, we chose the three classical problems of sizes $n = 12$, 15, and 20 from the test set of Nugent et al. [16]. These problem sizes are found in the first column of Table 1. We then compare different bounds on the optimal objective values of these problems in the remaining columns. From left to right, the second column gives the classical lower bounds of Gilmore and Lawler [8,15], followed by the objective values to RLT1 computed using an interior point method in [18], the objective values to RLT1 obtained by the dual ascent strategy of [10], the optimal objective values to RLT2 for $n = 12$ and $n = 15$, the objective values to RLT2 computed using the dual ascent strategy of the previous section, and the optimal integer values. The values in columns 3 and 5 are exact for RLT1 and RLT2 respectively, while those in columns 4 and 6 are approximate dual-based values. Here, we terminated the dual ascent strategy for both RLT1 and RLT2 after 2000 iterations.

The following observations can be made. Gilmore–Lawler is the weakest bound, and was shown in [1] to be the initialization value of the dual ascent strategy for RLT1. The RLT1 bound of the third column reduces the gap between the Gilmore–Lawler bound and the optimal integer value of the last column, but the bound weakens as n increases. The dual ascent strategy tends to give near-optimal solutions to RLT1, as the values in column four closely approximate those of column three. The RLT2 $n = 12$ bound of column five was

Table 1
Comparison of lower bound calculations

Problem size n	Gilmore–Lawler	RLT1 bound	RLT1 dual ascent	RLT2 bound	RLT2 dual ascent	Optimal value
12	493	523	523	578	578	578
15	963	1041	1039	1150[a]	1150	1150
20	2057	2182	2179	N/A	2508	2570

[a] Deduced from the RLT dual ascent bound of column 6.

obtained [17] using the "approximate dual projective" (ADP) algorithm [17,18], but the bounds could not be computed for the sizes $n = 15$ and 20 instances [18], due to the problem size (as later explained). The bounds from the dual ascent strategy applied to RLT2 are in column six. These bounds are exact for problem sizes $n = 12$ and $n = 15$ as they equal the optimal objective values to QAP (implying that the RLT2 bound for size $n = 15$ in column five is 1150). There is, however, a modest gap between the RLT2 dual ascent and the integer optimum for the size $n = 20$ problem. Overall, the table indicates that the dual ascent strategies for RLT1 and RLT2 perform well in obtaining near-optimal linear programming solutions to the respective problems, and also indicates that RLT2 provides a significant improvement over RLT1, reducing the gaps between the linear programming and optimal integer values for the sizes $n = 12$ and $n = 15$ problems by 100%, and reducing the gap for the size $n = 20$ problem by 84%. Notably, the RLT2 bounds of column six for each of these three problems is significantly tighter than all the alternatives found in Table 1 of [4], with the best reported values of 521, 1033, and 2290, respectively.

Relative to computational effort, the most efficient method of [17], the ADP algorithm, takes 6504.2 seconds on a 150 MHz Silicon Graphics Challenge computer (private communication with the second author), to calculate the size $n = 12$ RLT2 bound of 578. As the 150 MHz challenge is an old machine, it was difficult to get a direct speed comparison with our Sun Ultra 10. Fortunately, we were able to find that the ratio of SPECint ratings between the DEC Alpha w/333 MHz 21164 CPU and the 150 MHz SGI Challenge w/ R4400 CPU is 4.5. We already knew that the 360 MHz Sun Ultra 10 is about 1.47 times faster that the 333 MHz DEC Alpha on single processor integer programs. Thus, we were able to ascertain that the 360 MHz Sun Ultra 10 is about 6.62 times faster than the 150 MHz SGI Challenge. So, the ADP runtime for the Nugent 12 instance on our Sun Ultra 10 would have been about 983 seconds. In comparison, the same RLT2 bound was obtained using the dual ascent algorithm on the Sun Ultra 10 in just 93.4 seconds. The paper [17] does not consider problems having $n > 12$ due to the size of the linear programs RLT2, despite the fact that they essentially use (8) and (12) to reduce the problem to have only variables z_{ijklpq} with $j < l < q$ and variables y_{ijkl} with $j < l$ (and then remove (8) and (12)).

To gain a better understanding of the manner in which the dual ascent strategy progresses, we ran the $n = 20$ problem of [16] for 2000 iterations, and recorded the lower bound at different intervals. The results are found in Table 2. As might be expected, the majority of progress takes place during the first few hundred iterations, increasing the Gilmore–Lawler bound of 2057–2480.8 in the first 300 iterations, and then ultimately increasing to 2508.4. This is typical of the majority of problems we tested, exhibiting rapid initial gains and then slower tail-end improvement.

4.2. Performance of RLT2 in a branch-and-bound algorithm

We embedded RLT2 in a branch-and-bound algorithm to determine whether the tighter bounds will lead to improved solution methods. The level-2 form is considerably larger than the level-1, so a critical concern is whether the extra effort required to obtain these bounds is justified in the pruning process.

Two different algorithmic approaches using RLT2 are considered. Both begin by solving RLT1 with the dual ascent strategy, stopping with a maximum of 500 iterations, as suggested by Table 2, to obtain an objective representation of the form

$$K + \sum_i \sum_j \bar{b}_{ij} x_{ij} + \sum_{i \neq k} \sum_{j \neq l} \sum_k \sum_l \overline{C}_{ijkl} y_{ijkl} + \sum_{i \neq k,p} \sum_{j \neq l,q} \sum_{k \neq p} \sum_{l \neq q} \sum_p \sum_q \overline{D}_{ijklpq} z_{ijklpq}, \tag{22}$$

as found in (17). Here, K is the lower bound on QAP available at the termination of the dual ascent strategy, and $\bar{b}_{ij}$, $\overline{C}_{ijkl}$, and $\overline{D}_{ijklpq}$ are the nonnegative reduced costs. At any given node encountered in the binary tree

Table 2
Bound values and runtimes vs. iterations for size $n = 20$ problem of [16]

No. iterations	200	300	1000	1300	2000
Bound value	2473.2	2480.8	2486.2	2486.7	2508.4
Runtime (seconds)	2897.5	4327.8	14,310.2	18,633.4	27,862.9

search, an objective function of the form (22) is available. At such a node, the first approach performs an (abbreviated) dual ascent strategy on RLT2 to (22) until the lower bound available at some step is less than k% larger than the previous step, where k is some predetermined termination scalar. If the lower bound exceeds the incumbent value, the node is fathomed. Otherwise, a variable x_{ij}, which has not yet been fixed to a binary value, is set to 1 (using the look-ahead incrementing strategy of [12]). The second approach is similar to the first, but it computes a lower bound in two steps. It begins by relaxing (22) so that the coefficients $\overline{D}_{ijklpq}$ are all 0, and applies the dual ascent strategy on RLT1 using the same k% improvement criterion to terminate. If the acquired lower bound does not permit fathoming, then it reverts to the first strategy of computing the dual ascent value on RLT2. The idea is to save effort in bound computations by considering RLT1 instead of RLT2. Experimentation suggests setting $k = .008$ when the problem size n has $n \leqslant 28$ and $k = .0015$ when $n \geqslant 29$.

For our test cases, we consider seven challenging problems of sizes $n = 20$ to $n = 30$ from [16]. The problem sizes are listed in the first column of Table 3. Problems 20, 25, and 30 are taken directly from [16], with the remaining four instances obtained by truncating the data from the size $n = 30$ instance.

To determine the relative efficiency of our algorithms, we compare them with the most successful published exact methods. We are interested in both the number of nodes enumerated within the branch-and-bound routine and the CPU execution times. The first two columns of Table 3 give the problem size n and the measurement under study: these measurements are the numbers of nodes enumerated and the CPU execution times. The competing methods, listed from left to right in Table 3 starting with the third column, include the method of Anstreicher et al. [5] that uses quadratic programming bounds on QAP, the algorithm of Hahn et al. [12] that uses RLT1 bounds computed as in [1,10], and the two algorithmic approaches of this paper that use RLT2 bounds and the explained combination of RLT1 and RLT2 bounds respectively. The problems having $n = 28$ and $n = 30$ were not attempted using methods RLT1 and RLT2 since the combined method RLT1/RLT2 outperformed these approaches for sizes $n = 24$, 25, and 27, and since RLT1/RLT2 took 151,038 and 999,775 minutes (approximately 105 and 694 days) respectively.

To make a meaningful comparison in terms of CPU execution times, all minutes in Table 3 were normalized to a single CPU Dell 7150 PowerEdge server. The runtimes of [5,6,12], with [12] as corrected in [9], were reported normalized for a single CPU HP9000 C3000 workstation, and the runtimes for RLT2 and RLT1/RLT2 were taken from our Dell machine. The Dell is approximately 1.37 times faster than the HPC3000. Consequently, the minutes in columns 3 and 4 were obtained by dividing the times reported in [5,6,12] by 1.37. The best time for each of the seven problems is italicized.

Table 3
Comparison of competing branch-and-bound algorithms

Size	Measurement	Quad. program	RLT1	RLT2	RLT1/RLT2
20	No. of nodes	1,040,308	181,073	2257	1123
	Minutes	106	*43*	169	79
22	No. of nodes	1,225,892	1,354,837	2381	1609
	Minutes	*98*	417	260	109
24	No. of nodes	31,865,440	16,710,701	13,995	7632
	Minutes	4255	8687	2775	*1798*
25	No. of nodes	71,770,751	27,409,486	12,920	11,796
	Minutes	8698	23,727	5391	*4236*
27	No. of nodes	402,000,000	297,648,966	46,315	33,754
	Minutes	69,057	334,980	27,474	*21,073*
28	No. of nodes	2,230,000,000	–[a]	–[a]	202,295
	Minutes	337,612	–[a]	–[a]	*151,038*
30	No. of nodes	11,900,000,000[b]	–[a]	–[a]	543,061
	Minutes	2,662,528	–[a]	–[a]	*999,775*

[a] Not attempted.
[b] The authors cite 11,892,208,412 nodes in a private communication.

The four methods are competitive in terms of CPU execution times, though methods RLT2 and RLT1/RLT2 of columns five and six are much more efficient for the larger problems of sizes $n \geqslant 24$. For the three largest instances having $n = 27$, $n = 28$ and $n = 30$, the RLT1/RLT2 algorithm takes less than half the execution time of the quadratic programming alternative, as noted by comparing columns three and six. This is striking given the size and complexity of these problems, as evidenced by the following statement found in [5], where nug30 refers to the size $n = 30$ problem. "For the solution of nug30 an average of 650 worker machines were utilized over a one-week period, providing the equivalent of almost 7 years of computation on a single HP9000 C3000 workstation. ... To our knowledge these are among the most extensive computations ever performed to solve discrete optimization problems to optimality."

Relative to nodes explored, it is evident that the bounding strength afforded by RLT2 is very useful in fathoming nodes within the enumeration. In fact, the relatively few numbers of nodes needed with RLT2 is striking. For the problem sizes $n = 28$ and $n = 30$, for example, the quadratic programming algorithm evaluated approximately 2.2 billion and 11.9 billion nodes respectively. In contrast, the RLT1/RLT2 algorithm examined approximately two hundred thousand and 543,000 nodes respectively for these same problems. As might be expected, however, the effort required to solve RLT2 does not appear justified for problems having $n \leqslant 22$.

5. Conclusions and ongoing challenges

We developed exact solution methods for the quadratic assignment problem using the level-2 rlt of [19–21]. The resulting linear representation, problem RLT2, is large in size and highly degenerate. In order to solve it, we devised a dual ascent strategy that exploits a block-diagonal structure of the constraints. This strategy is an extension of that found in [1].

Problem RLT2 provides sharp lower bounds, as shown in Table 1, and consequently leads to very competitive exact solution strategies. A striking outcome, documented in Table 3, is the relatively few numbers of nodes considered in the binary search tree to verify optimality. This leads to efficient solution strategies for the more difficult problems having $n \geqslant 24$.

There are various avenues for future research. A shortcoming of our algorithms with RLT2 is the large RAM requirement. Fig. 1 is a plot of the total RAM used as a function of problem size n. The horizontal axis gives the problem size n and the vertical axis gives the memory in KBytes. We estimate that, with each unit increase in problem size n, the required memory grows by an approximate factor of 1.36. A major challenge

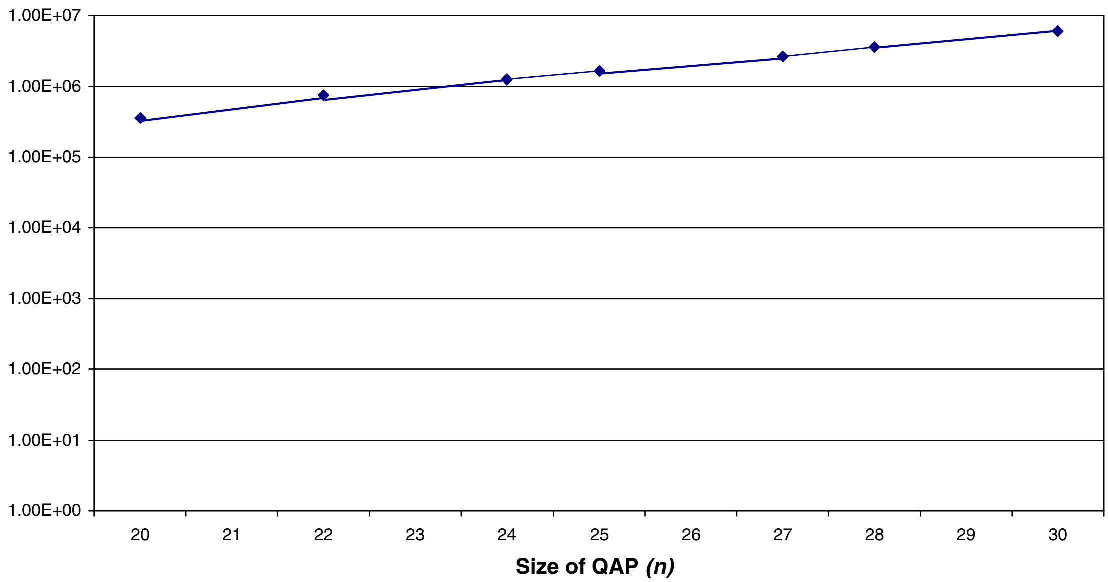

Fig. 1. Memory required for RLT2 branch-and-bound algorithm.

is to reduce this requirement. An emphasis of future research will be to develop methods for circumventing the large memory requirements, perhaps through innovative storage and retrieval methods and through strategic implementations of the bounding method.

The RAM requirements can also be reduced through a redesign of the enumerative process. Currently, as we progress through the search tree, we maintain the updated objective coefficients to RLT2 at each active node. For a problem of size n, there are n^2 variables x_{ij}, $n^2(n-1)^2$ variables y_{ijkl}, and $n^2(n-1)^2(n-2)^2$ variables z_{ijklpq}, and every variable has an associated coefficient. Specifically, every time a node is not fathomed, our depth-first strategy proceeds a level deeper into the tree, and maintains all relevant information for the active nodes, as well as for the newly encountered node. We can redesign the algorithm so that relevant information is kept only at the new node. This approach will reduce memory requirements but necessitate a recalculation of information every time a pruning takes place. Combination strategies that limit the number of active nodes for which information is stored may be most useful here.

If one succeeds in using level-2 rlt formulations for solving the qap for instances larger than size 30, it is conceivable that level-3 representations will work even better. In addition, the methods in this paper can be extended beyond the qap. The same type rlt representations can be constructed for general mixed 0–1 programs, with a similar decomposition approach available by the Lemma. Our future efforts will consider such problems.

Acknowledgements

The authors thank Professors Bakhtier Farouk and Nihat Bilgutay of the College of Engineering, Drexel University, and Professor Preston B. Moore of the Department of Chemistry, University of Pennsylvania, who graciously provided the computing resources necessary for this work. The work was supported in part by an international travel grant INT-9900376, as well as research grants DMI-0400155 and DMI-0423415 from the National Science Foundation.

References

[1] W.P. Adams, T.A. Johnson, Improved linear programming-based lower bounds for the quadratic assignment problem, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 16 (1994) 43–75.
[2] W.P. Adams, H.D. Sherali, A tight linearization and an algorithm for zero–one quadratic programming problems, Management Science 32 (10) (1986) 1274–1290.
[3] W.P. Adams, H.D. Sherali, Linearization strategies for a class of zero–one mixed integer programming problems, Operations Research 38 (2) (1990) 217–226.
[4] K.M. Anstreicher, N.W. Brixius, A new bound for the quadratic assignment problem based on convex quadratic programming, Mathematical Programming 89 (3) (2001) 341–357.
[5] K. Anstreicher, N. Brixius, J.-P. Goux, J. Linderoth, Solving large quadratic assignment problems on computational grids, Mathematical Programming 91 (3) (2002) 563–588.
[6] N.W. Brixius, K.M. Anstreicher, Solving quadratic assignment problems using convex quadratic programming relaxations, Optimization Methods and Software 16 (1–4) (2002) 49–68.
[7] R.E. Burkard, S.E. Karisch, F. Rendl, QAPLIB – A quadratic assignment problem library, European Journal of Operational Research 55 (99) (1991) 115–119, QAPLIB is found on the web at <http://www.seas.upenn.edu/qaplib/>.
[8] P.C. Gilmore, Optimal and suboptimal algorithms for the quadratic assignment problem, Journal of the Society for Industrial and Applied Mathematics. 10 (2) (1962) 305–313.
[9] P.M. Hahn, Corrigendum to tree elaboration strategies in branch-and-bound algorithms for solving the quadratic assignment problem, Yugoslav Journal of Operations Research 12 (2) (2002) 271–272.
[10] P.M. Hahn, T.L. Grant, Lower bounds for the quadratic assignment problem based upon a dual formulation, Operations Research 46 (6) (1998) 912–922.
[11] P.M. Hahn, T.L. Grant, N. Hall, A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method, European Journal of Operational Research 108 (3) (1998) 629–640.
[12] P.M. Hahn, W.L. Hightower, T.A. Johnson, M. Guignard-Spielberg, C. Roucairol, Tree elaboration strategies in branch-and-bound algorithms for solving the quadratic assignment problem, Yugoslav Journal of Operations Research 11 (1) (2001) 41–60.
[13] P.M. Hahn, J. Krarup, A hospital facility problem finally solved, The Journal of Intelligent Manufacturing 12 (5/6) (2001) 487–496.
[14] T.A. Johnson, New Linear Programming-Based Solution Procedures for the Quadratic Assignment Problem. Ph.D. Dissertation, Clemson University, Clemson, SC, 1992.
[15] E.L. Lawler, The quadratic assignment problem, Management Science 9 (4) (1963) 586–599.

[16] C.E. Nugent, T.E. Vollman, J. Ruml, An experimental comparison of techniques for the assignment of facilities to locations, Operations Research 16 (1) (1968) 150–173.
[17] K.G. Ramakrishnan, M.C.G. Resende, B. Ramachandran, J.F. Pekny, in: P.M. Pardalos, A. Migdalas, R. Burkard (Eds.), Tight QAP Bounds via Linear Programming. Combinatorial and Global Optimization, World Scientific Publishing, Singapore, 2002, pp. 297–303.
[18] M.G.C. Resende, K.G. Ramakrishnan, Z. Drezner, Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming, Operations Research 43 (5) (1995) 781–791.
[19] H.D. Sherali, W.P. Adams, A hierarchy of relaxations between the continuous and convex hull representations for zero–one programming problems, SIAM Journal on Discrete Mathematics 3 (3) (1990) 411–430.
[20] H.D. Sherali, W.P. Adams, A hierarchy of relaxations and convex hull characterizations for mixed-integer zero–one programming problems, Discrete Applied Mathematics 52 (1) (1994) 83–106.
[21] H.D. Sherali, W.P. Adams, A Reformulation–Linearization Technique for Solving Discrete and Continuous Nonconvex Problems, Kluwer Academic Publishers, Dordrecht/Boston/London, 1999.