A revised reformulation-linearization technique for the quadratic assignment problem

Borzou Rostami *, Federico Malucelli
Dipartimento di Elettronica, Informazione e Bioingenieria, Politecnico di Milano, Milan, Italy

Received 10 April 2014
Received in revised form 2 July 2014
Accepted 21 August 2014

1. Introduction

Consider the general form of the Quadratic Assignment Problem (QAP) proposed by Lawler [1] as follows:

$$\text{QAP}: \min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} q_{ijkl}x_{ij}x_{kl} + \sum_{i=1}^{n}\sum_{k=1}^{n} c_{ik}x_{ik} \tag{1}$$

$$\text{s.t. } x \in X,\ x \text{ binary} \tag{2}$$

where

$$X = \left\{x \geq 0 : \sum_{j=1}^{n} x_{ij} = 1\ \forall i = 1,\ldots,n;\ \sum_{i=1}^{n} x_{ij} = 1\ \forall j = 1,\ldots,n\right\}. \tag{3}$$

The QAP was first introduced by Koopmans and Beckmann [2] in the context of facility location to deal with a one-to-one assignment of n facilities to n locations when the objective function accounts for interaction costs between pairs of facilities. Since then, the QAP has been used to model many applications including, among others, backboard wiring [3], typewriter keyboards and control panels design [4], scheduling [5], storage-and-retrieval [6]. It has been shown that the QAP is among the most difficult NP-hard combinatorial optimization problems and, in general, solving instances of size $n \geq 30$ in a reasonable time is impossible [7]. Due to its quadratic nature, many attempts have been made in the literature to linearize the objective function so that the resulting lower bound is strong enough to be used in a branch-and-bound algorithm. Among the best lower bounding approaches in the literature we can refer the reader to Frieze and Yadegar [8], Carraresi and Malucelli [9,10], Adams and Johnson [11], Karisch et al. [12], the level-1 RLT dual-ascent bound by Hahn and

* Corresponding author. Tel.: +39 022399 4029; fax: +39 022399 3412.
E-mail addresses: borzou.rostami@polimi.it, bo.rostami@gmail.com (B. Rostami), federico.malucelli@polimi.it (F. Malucelli).

Grant [13], the convex quadratic programming bound by Anstreicher and Brixius [14], the level-2 RLT by Adams et al. [15], and level-3 RLT by Hahn et al. [16].

In this paper we present revised versions of the RLT representations for the QAP. The main idea is to remove some set of constraints in each level of the RLT representation of the QAP so that the resulting problem remains equivalent to the original one, and the set of new constraints possesses the block-diagonal structure.

2. Reformulation-linearization technique

In this section we present the Reformulation-linearization technique (RLT) applied to the QAP. Based on the RLT technique for general zero–one polynomial programs by Adams and Sherali [17,18], the first RLT representation for the QAP was introduced by Adams and Johnson [11]. Consider problem QAP as presented in (1)–(3). The level-1 RLT representation is generated via the following two steps:

Reformulation: Multiply each of the $2n$ equations and each of the n^2 nonnegativity constraints defining X by each of the n^2 binary variables x_{kl}, and append these new constraints to the formulation. When the 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}^2 with x_{kl} throughout the constraints and set $x_{ij}x_{kl} = 0$ if $i = k$ and $j \neq l$ or $i \neq k$ and $j = l$.

Linearization: For all (i, j, k, l) with $i \neq k$ and $j \neq l$, substitute each product $x_{ij}x_{kl}$ with y_{ijkl}. Enforce the equality $y_{ijkl} = y_{klij}$ for all (i, j, k, l) with $i < k$ and $j \neq l$.

The level-1 RLT results as follows:

$$\text{RLT1}: \min \sum_i \sum_j \sum_{k\neq i} \sum_{l\neq j} q_{ijkl} y_{ijkl} + \sum_i \sum_j c_{ij} x_{ij} \tag{4}$$

$$\text{s.t.} \sum_{i\neq k} y_{ijkl} = x_{kl} \quad \forall (j, k, l),\ j \neq l \tag{5}$$

$$\sum_{j\neq l} y_{ijkl} = x_{kl} \quad \forall (i, k, l),\ i \neq k \tag{6}$$

$$y_{ijkl} = y_{klij} \quad \forall (i, j, k, l),\ i < k,\ j \neq l \tag{7}$$

$$y_{ijkl} \geq 0 \quad \forall (i, j, k, l),\ i \neq k,\ j \neq l \tag{8}$$

$$x \in X, \quad x \text{ binary}. \tag{9}$$

Note that for any feasible solution (x, y) to RLT1, the RLT theory enforces the equations $y_{ijkl} = x_{ij}x_{kl}$ for all $(i, j, k, l),\ i \neq k,\ j \neq l$ [17,19]. Thus we have the following:

Proposition 1. *Problems QAP and RLT 1 are equivalent.*

Eq. (7) is very important and says that if an element y_{ijkl}, $i \neq k, j \neq l$ is part of a solution (i.e., equal to 1) then it has a "complementary element" y_{klij} that is also in that solution. In general, the RLT1 representation has a large number of variables and constraints, which makes it computationally challenging, even for small QAP instances. Resende et al. [20] performed a computational test of the lower bounds generated by the LP relaxation of the RLT1. They reduced the numbers of variables and constraints in RLT1 by removing all variables y_{ijkl} with $i > k$ and $j \neq l$ and by making the substitutions suggested by (7) throughout the objective function and constraints. Then they solved the LP relaxation by using an experimental interior point method code, called ADP. To solve the RLT1, Adams and Johnson provide a Lagrangian relaxation which has a block-diagonal structure. More precisely they dualize constraints (7) on the complementary pairs and decompose the resulting problem into n^2 separate linear assignment problems of size $n-1$ and a linear assignment problem of size n. Hahn and Grant [13] gave a different interpretation of the same decomposition for lower bound calculation by using a dual-ascent strategy. Their dual-ascent procedure gives a bound very close to optimum of the LP relaxation of the RLT1, improving upon the computational results of Adams and Johnson [11], and requiring only a small fraction of the time of Resende et al. [20].

Based on the success of level-1 RLT representation to gain a tight bound for the QAP and also due to the block-diagonal structure of the problem which lends itself to efficient solution methods, the level-2 and level-3 RLT can be defined in the same way as the level-1 RLT via the reformulation and linearization steps. In the level-2 RLT representation, in addition to the operations done in the level-1, each binary variable in X is multiplied also by products $x_{kl}x_{pq}$ having $k \neq p$ and $l \neq q$. For more details concerning the reformulation and linearization step for the level-2 RLT we refer the reader to [15]. The level-2 RLT is called RLT2 and is written as follows:

$$\min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n q_{ijkl} y_{ijkl} + \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \tag{10}$$

$$\text{s.t.} \sum_{i\neq k,p} z_{ijklpq} = y_{klpq} \quad \forall (j, k, l, p, q),\ j \neq l \neq q,\ k \neq p \tag{11}$$

$$\sum_{j\neq l} z_{ijklpq} = y_{klpq} \quad \forall(i,k,l,p,q),\ i\neq k\neq p,\ j\neq q \tag{12}$$

$$z_{ijklpq} = z_{ijpqkl} = \cdots = z_{pqklij} \quad \forall(i,j,k,l,p,q),\ i<k<p,\ j\neq l\neq q \tag{13}$$

$$z_{ijklpq} \geq 0 \quad \forall(i,j,k,l,p,q),\ i\neq k\neq p,\ j\neq l\neq q \tag{14}$$

$$(5), (6), (7), (8), (9). \tag{15}$$

The linear relaxation of the RLT2 is increasingly large and highly degenerate. Ramakrishnan et al. [21] enforced the constraints (13) to combine complementary variables and reduce the number of constraints and variables, then used commercial linear programming package CPLEX to solve the linear relaxation of the RLT2. However, because of the problem size and limitations of CPLEX, they were only able to solve instances up to size 12. Following the idea of Hahn and Grant [13] to solve the RLT1 in an efficient way, Adams et al. [15] have presented a dual-ascent strategy that exploits the block-diagonal structure of constraints in the RLT2 form.

In order to get even tighter bounds for the QAP, Zhu presented the level-3 RLT in his Ph.D. dissertation [22]. The RLT3 formulation is significantly larger than the previous levels of RLT, but its continuous linear relaxation provides the tightest lower bound of all three RLT models. Hahn et al. [16] implemented a dual-ascent procedure similar to that employed in Adams et al. [15] for RLT2.

3. A revised RLT

As far as the tightness of the bounds is concerned, the RLT representations of the QAP are among the most successful lower bounding approaches. However, in the high level RLT representation of the QAP these bounds require much computational effort, which can be problematic within a branch-and-bound algorithm. In order to speed up the bound computation in the level-d RLT representation of the QAP, we construct a smaller reformulation for each level of the RLT based on the structure of the problem. Let us start with the level-1 RLT formulation presented in (4)–(9). The revised RLT1 (RRLT1) formulation is defined as follows:

$$\text{RRLT1}:\ \min \sum_i \sum_j \sum_{k\neq i} \sum_{l\neq j} q_{ijkl} y_{ijkl} + \sum_i \sum_j c_{ij} x_{ij}$$

s.t. (5), (6), (7), (8)

$x \in X'$, $\quad x$ binary

where

$$X' = \left\{ x \geq 0 : \sum_{j=1}^{n} x_{ij} = 1\ \forall i = 1,\ldots,n \right\}. \tag{16}$$

Note that the only difference with RLT1 is that constraints $\sum_{i=1}^{n} x_{ij} = 1, \forall j = 1,\ldots,n$ are missing.

Theorem 2. *The problems RLT*1 *and RRLT*1 *are equivalent.*

Proof. To prove the theorem we use the idea of [19]. Consider any feasible solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ to Problem RRLT1. We first show that the following equations must hold.

$$\hat{y}_{pqst} = \hat{x}_{pq}\hat{x}_{st} \quad \forall\ (p,q,s,t),\ p\neq s,\ q\neq t. \tag{17}$$

If $\hat{x}_{pq} = 0$, constraint $\sum_{i\neq k} \hat{y}_{ijpq} = \hat{x}_{pq}$ of (5), together with the nonnegativity restrictions $\hat{y}_{ijpq} \geq 0$ enforces that $\hat{y}_{ijpq} = 0$ for all (i,j), $i\neq p, j\neq q$. Consider the case $\hat{x}_{pq} = \hat{x}_{st} = 1$, and by contradiction assume that $\hat{y}_{pqst} = \hat{y}_{stpq} < 1$. The constraint $\sum_{j\neq q} \hat{y}_{sjpq} = \hat{x}_{pq}$ of (6), together with $\hat{y}_{stpq} < 1$ implies that there exists an index $l \neq t, q$ with $\hat{y}_{slpq} > 0$. By considering constraint (7), we have $\hat{y}_{pqsl} = \hat{y}_{slpq} > 0$, so that (6) implies $x_{sl} = 1$. The equalities $x_{sl} = x_{st} = 1$ for $l \neq t$, contradict the constraint $\sum_{j=1}^{n} x_{sj} = 1$ of X'. Consequently, $\hat{y}_{pqst} = \hat{x}_{pq}\hat{x}_{st}$ for binary $\hat{x}_{pq}$ and $\hat{x}_{qr}$.

Now we show that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is a feasible solution for the RLT1. Since constraints (5), (6), (7), and (8) together with the binary restriction are precisely the same in both models, the proof is to show that $\sum_{i=1}^{n} \hat{x}_{ij} = 1, \forall j = 1,\ldots,n$. Consider an index $s \in \{1,\ldots,n\}$ such that $\sum_{i=1}^{n} \hat{x}_{is} = \epsilon \neq 1$. Multiplying this equation by binary variable $\hat{x}_{kl}$ with $l \neq s$ to obtain

$$\sum_{i\neq k} \hat{x}_{is}\hat{x}_{kl} = \epsilon x_{kl} \quad \forall(k,l),\ l\neq s. \tag{18}$$

By (17) and (18), we have

$$\sum_{i\neq k} \hat{y}_{iskl} = \sum_{i\neq k} \hat{x}_{is}\hat{x}_{kl} = \epsilon x_{kl} \quad \forall(k,l),\ l\neq s. \tag{19}$$

Since $\epsilon \neq 1$, Eqs. (19) contradict (6) and the proof is complete. $\square$

The following theorem formally shows that the continuous relaxations of the RRLT1 (CRRLT1) is as tight as the continuous relaxation of the RLT1 (CRLT1), in that a linear combination of the constraints of CRRLT1 implies the constraints $\sum_{i=1}^{n} x_{ij} = 1\ \forall j = 1, \ldots, n$.

Theorem 3. *Problems CRLT1 and CRRLT1 are equivalent.*

Proof. Since constraints (5), (6), (7), and (8) appear in both CRLT1 and CRRLT1, the proof reduces to show that constraints $\sum_{i=1}^{n} x_{ij} = 1\,\forall j = 1, \ldots, n$ of X defined in (3) can be computed as linear combinations of the constraints of CRRLT1. Consider any $(j, k, l), j \neq l$, and observe that constraints (6) enforce that $\sum_{i\neq k} y_{ijkl} = x_{kl}$. Summing this equations over all $(k, l), l \neq j$, we obtain

$$\sum_{l\neq j}\sum_{k}\sum_{i\neq k} y_{ijkl} = \sum_{l\neq j}\sum_{k} x_{kl} \quad \forall j = 1, \ldots, n. \tag{20}$$

By definition of X' in (16), the right hand side of (20) can be written as follows:

$$\sum_{l\neq j}\sum_{k} x_{kl} = \sum_{k}\sum_{l} x_{kl} - \sum_{k} x_{kj} = n - \sum_{k} x_{kj} \quad \forall j = 1, \ldots, n. \tag{21}$$

Now consider the left hand side of (20). By using (5) and (7), it can be written as:

$$\begin{aligned}\sum_{l\neq j}\sum_{k}\sum_{i\neq k} y_{ijkl} &= \sum_{i\neq k}\sum_{k}\sum_{l\neq j} y_{klij} = \sum_{i\neq k}\sum_{k} x_{ij} = \sum_{k}\sum_{i\neq k} x_{ij}\\ &= \sum_{k}\sum_{i} x_{ij} - \sum_{k} x_{kj} = n\sum_{i} x_{ij} - \sum_{k} x_{kj} \quad \forall j = 1, \ldots, n.\end{aligned} \tag{22}$$

By (21) and (22), we have $\sum_{i=1}^{n} x_{ij} = 1\ \forall j = 1, \ldots, n$. The proof is completed. $\square$

We now turn to the level-2 RLT representation of the QAP presented in (10)–(15). We give a smaller reformulation of the RLT2 called revised RLT2 (RRLT2) by substituting X defined in (3) with X' defined in (16) and removing constraints $\sum_{i\neq k} y_{ijkl} = x_{kl}\ \forall (j, k, l),\ j \neq l$ from the RLT2 formulation. The RRLT2 has the following form:

$$\begin{aligned}&\text{RRLT2}:\ \min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} q_{ijkl}y_{ijkl} + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}\\ &\text{s.t. (6)–(8), (11)–(14)}\\ &x \in X', \quad x \text{ binary}.\end{aligned}$$

Proposition 4. *Problems RLT2 and RRLT2 are equivalent.*

Proof. The proof is a trivial extension of the proof of Theorem 2. $\square$

Theorem 5. *Problems CRLT2 and CRRLT2 are equivalent.*

Proof. Following the same idea as the proof of Theorem 3 we show that constraints $\sum_{i=1}^{n} x_{ij} = 1\,\forall j = 1, \ldots, n$ of X defined in (3) and constraints (11) can be computed as linear combinations of the constraints of CRRLT2. Since the proof of the first part is the same as the one in Theorem 3, here we provide the proof for the second part. Consider any (j, k, l, p, q) with $j \neq l \neq q, k \neq p$, and observe that constraints (11) enforce that $\sum_{i\neq k,p} z_{ijklpq} = y_{klpq}$. Summing these equations over all (p, q) having $p \neq k$ and $q \neq j, l$, we obtain

$$\sum_{q\neq j,l}\sum_{p\neq k}\sum_{i\neq k,p} z_{ijklpq} = \sum_{q\neq j,l}\sum_{p\neq k} y_{klpq} \quad \forall (j, k, l),\ j \neq l. \tag{23}$$

By (6) and (7), the right hand side of (23) can be written as follows:

$$\sum_{q\neq j,l}\sum_{p\neq k} y_{klpq} = \sum_{p\neq k}\sum_{q\neq l} y_{klpq} - \sum_{p\neq k} y_{klpj} = (n-1)x_{kl} - \sum_{p\neq k} y_{pjkl} \quad \forall (j, k, l),\ j \neq l. \tag{24}$$

Now consider the left hand side of (20). By using (12) and (13), it can be written as:

$$\begin{aligned}\sum_{q\neq j,l}\sum_{p\neq k}\sum_{i\neq k,p} z_{ijklpq} &= \sum_{i\neq k,p}\sum_{p\neq k}\sum_{q\neq j,l} z_{pqijkl} = \sum_{p\neq k}\sum_{i\neq k,p} y_{ijkl}\\ &= \sum_{p\neq k}\sum_{i\neq k} y_{ijkl} - \sum_{p\neq k} y_{pjkl} = (n-1)\sum_{i\neq k} y_{ijkl} - \sum_{p\neq k} y_{pjkl} \quad \forall (j, k, l),\ j \neq l.\end{aligned} \tag{25}$$

By (24) and (25), we have $\sum_{i\neq k} y_{ijkl} = x_{kl}\ \forall (j, k, l), j \neq l$. The proof is completed. $\square$

The idea of removing some set of constraints in level-1 and level-2 RLT representation of the QAP can be generalized for level-d RLT formulation with $1 \leq d \leq n$. In each level-d RLT representation we can eliminate one set of constraints generated in level-$(d-1)$ RLT, where level-0 RLT represent the original QAP. More precisely for each d, $2 \leq d \leq n-1$, the total number of constraints in the level-d RLT can be reduced by

$$f(d) = f(d-1) + (n-d+1)\prod_{i=0}^{d-2}(n-i)^2,$$

where $f(1) = n$.

4. Computational experiments

In this section we present the computational results of comparing the level-d RLT with the level-d RRLT for $d = 1, 2$, in terms of bound tightness and computational time. To obtain a lower bound for the QAP we solve the continuous relaxation of RRLT1 and RRLT2 by applying the Lagrangian relaxation. We implemented the algorithms in C++ language and run on an Intel Xeon CPU E5335 (2 quad core CPUs 2GH). Since the solution methods for RRLT1 and RRLT2 are quite similar, we restrict our attention to explain the solving process to the RRLT2. We first place constraints (7) and (13) into the objective function. The Lagrangian function is then defined as:

$$K + \min\left\{\sum_i\sum_j\sum_k\sum_{p\neq i}\sum_{q\neq j}\sum_{l\neq k}\bar{D}_{ijkpql}z_{ijkpql} + \sum_i\sum_j\sum_{k\neq i}\sum_{l\neq j}\bar{B}_{ijkl}y_{ijkl}\right.$$
$$\left. + \sum_i\sum_j\bar{c}_{ij}x_{ij} : \text{(6), (8), (11), (12), (14)},\ x \in X'\right\} \tag{26}$$

where $\bar{B}_{ijkl}$ and $\bar{c}_{ij}$ are the adjusted values for q_{ijkl} and c_{ij} respectively, after placing constraints (7) into the objective function, $\bar{D}_{ijkpql}$ are the coefficients corresponding to the variables z_{ijkpql} after placing constraints (13) into the objective function, and $K = 0$.

Theorem 6. *An optimal solution (x^*, y^*, z^*) for problem (26) can be obtained via solving the semi-assignment problem*

$$K + \min\left\{\sum_p\sum_q(\bar{c}_{pq} + \rho_{pq})x_{pq} : x \in X'\right\}, \tag{27}$$

where for each (p, q), ρ_{pq} is obtained by solving the following semi-assignment problem:

$$\rho_{pq} = \min \sum_{k\neq p}\sum_{l\neq q}(\bar{B}_{klpq} + \gamma_{klpq})y_{klpq}$$
$$s.t. \sum_{l\neq q} y_{klpq} = 1 \quad \forall k = 1, \ldots, n,\ k \neq p$$
$$y_{klpq} \geq 0 \quad \forall(k, l),\ k \neq p,\ l \neq q$$

and where for each (k, l, p, q) with $p \neq k$ and $q \neq l$

$$\gamma_{klpq} = \min \sum_{i\neq p,k}\sum_{j\neq q,l}\bar{D}_{ijklpq}z_{ijklpq}$$
$$s.t. \sum_{i\neq p,k} z_{ijklpq} = 1 \quad \forall j = 1, \ldots, n,\ j \neq q, l$$
$$\sum_{j\neq q,l} z_{ijklpq} = 1 \quad \forall i = 1, \ldots, n,\ i \neq p, k$$
$$z_{ijklpq} \geq 0 \quad \forall(i, j),\ i \neq p, k,\ j \neq q, l.$$

We applied the dual ascent algorithm proposed in Adams et al. [15] to this new problem. The procedure consists of updating the constant term K and the cost matrices $\bar{D}$, $\bar{B}$ and $\bar{c}$, in such a way that the cost of any (integer) feasible solution with respect to the modified objective function remains unchanged, while maintaining nonnegative coefficients. As a consequence of this property, the value of K at any moment of the execution is a valid lower bound on the optimal solution cost for the QAP.

For computational testing of comparing the RLT and RRLT, we used a representative set of instances from the QAPLIB [23] and some instances from the test set of Drugan [24]. This new test set currently was introduced by Drugan in [24] and

Table 1
Comparison of the level-1, 2 RLT and RRLT lower bounds and CPU times for the instances of QAPLIB. The best results are in boldface.

Instance	Opt.	RLT1		RRLT1		RLT2		RRLT2	
		Lb	Time	Lb	Time	Lb	Time	Lb	Time
Chr20a	2 192	2 161	54	2 159	52	**2 192**	1 689	**2 192**	**1 619**
Chr22a	6 156	6 077	74	6 085	74	**6 156**	1 837	**6 156**	**1 811**
Chr25a	3 796	3 565	131	3 553	132	**3 796**	6 541	**3 796**	**5 808**
Had16	3 720	3 525	19	3 525	19	**3 720**	407	**3 720**	**381**
Had18	5 358	5 036	30	5 035	31	**5 358**	19 044	**5 358**	**14 448**
Had20	6 922	6 504	47	6 507	47	**6 922**	56 791	**6 922**	**48 714**
Rou12	235 528	219 365	8	219 329	6	**235 528**	**33**	**235 528**	**33**
Rou15	354 210	318 496	14	318 413	14	351 106	**11 300**	**351 537**	12 167
Rou20	725 520	632 453	47	632 346	47	687 320	80 052	**688 391**	**77 251**
Bur26a	5 426 670	5 203 976	179	5 186 671	170	5 276 411	19 903	**5 277 443**	**16 911**
Bur26b	3 817 852	3 641 582	171	3 627 822	177	3 707 590	18 153	**3 798 123**	**17 233**
Bur26c	5 426 795	5 188 444	172	5 176 906	173	5 266 012	19 456	**5 269 391**	**18 346**
Bur26d	3 821 225	3 634 960	172	3 626 710	170	3 703 214	17 873	**3 706 253**	**17 252**
Nug12	578	512	6	512	6	**578**	774	**578**	**364**
Nug15	1 150	1 002	15	1 001	14	1 141	11 067	**1 142**	**11 015**
Nug18	1 930	1 623	30	1 621	31	1 860	40 480	**1 863**	**39 379**
Nug20	2 570	2 240	47	2 138	47	2 450	64 496	**2 455**	**63 858**
Tai15a	388 214	346 896	14	347 086	14	377 805	11 244	**378 347**	**10 037**
Tai15b	51 765 268	51 459 245	14	51 443 801	14	**51 765 268**	240	**51 765 268**	**208**
Tai20a	703 482	608 846	46	608 199	47	662 750	88 497	**663 855**	**78 890**
Tai20b	122 455 319	88 400 570	54	87 855 727	57	**122 455 319**	3 039	**122 455 319**	**2 791**

Table 2
Comparison of the level-1, 2 RLT and RRLT lower bounds and CPU times for the cQAP data set. The best results are in boldface.

Instance	Opt.	RLT1		RRLT1		RLT2		RRLT2	
		Lb	Time	Lb	Time	Lb	Time	Lb	Time
cqap20-0	238 754	215 117	48	215 045	48	**238 754**	541	**238 754**	**393**
cqap20-1	233 690	204 845	46	204 837	46	**233 690**	707	**233 690**	**513**
cqap20-2	230 750	204 996	47	204 970	48	**230 750**	668	**230 750**	**443**
cqap20-3	235 432	211 780	47	211 781	48	**235 432**	452	**235 432**	**450**
cqap20-4	242 392	212 641	47	212 928	47	**242 392**	584	**242 392**	**461**
cqap20-5	236 894	205 933	47	205 995	46	**236 894**	748	**236 894**	**459**
cqap20-6	241 720	210 392	48	210 639	48	**241 720**	651	**241 720**	**525**
cqap20-7	242 388	217 115	48	217 151	48	**242 388**	600	**242 388**	**545**
cqap20-8	236 546	210 846	47	210 863	48	**236 546**	726	**236 546**	**413**
cqap20-9	239 180	209 644	47	209 660	47	**239 180**	884	**239 180**	**607**
Average	237 775	210 331	47	210 287	47	**237 775**	656	**237 775**	**481**
cqap24-0	312 308	264 043	98	263 875	99	**312 308**	1749	**312 308**	**1373**
cqap24-1	305 074	269 989	99	270 009	100	**305 074**	1848	**305 074**	**1474**
cqap24-2	310 154	269 105	97	268 987	97	**310 154**	**1556**	**310 154**	1576
cqap24-3	307 622	266 878	98	266 758	99	**307 622**	**1745**	**307 622**	1765
cqap24-4	313 614	274 491	98	274 371	97	**313 614**	1558	**313 614**	**1179**
cqap24-5	308 634	268 792	98	268 766	98	**308 634**	1941	**308 634**	**1572**
cqap24-6	301 196	256 687	98	256 581	98	**301 196**	2157	**301 196**	**1640**
cqap24-7	301 742	262 322	98	262 242	98	**301 742**	2128	**301 742**	**1428**
cqap24-8	303 516	265 041	99	265 383	99	**303 516**	2116	**303 516**	**1715**
cqap24-9	309 774	277 585	99	277 553	101	**309 774**	1426	**309 774**	**1425**
Average	307 363	267 493	98	267 452	99	**307 363**	1822	**307 363**	**1515**

are called composite QAPs (cQAPs). Tables 1 and 2 report the lower bounds and required CPU times of the dual ascent strategy (terminated in 2500 iterations) applied to level-1 and level-2 of RLT and RRLT for the instances of QAPLIB and cQAPs, respectively. Since after only a small number of iterations, the bound grows to a significant percentage of its final value, and also due to the tradeoff between bound strength and CPU execution time we terminated the dual-ascent algorithms for the RLT2 and RRLT2 at iteration 100 if $n > 20$. In both tables the instance names and the corresponding dimensions are found in the first column. In the second column, there are the optimal values for each instance. The third and fourth columns give the RLT1 lower bound and its CPU time, followed by the lower bound values and CPU times of RRLT1 in columns fifth and sixth, the RLT2 in columns seventh and eighth, and RRLT2 in columns ninth and tenth. The best overall lower bounds and CPU times are indicated in boldface.

For the tested instances from the QAPLIB, the dual ascent strategy applied to RLT1 and RRLT1 almost provide the same bounds with the same computational times, while the dual ascent strategy applied to RLT2 and RRLT2 provides the different results. More precisely, as we can observe from Table 1, the bounds from the dual ascent strategy applied to RLT2 and RRLT2 for problems Chr20a, Chr22a, Chr25a, Had16, Had18, Had20, Rou12, Tai15b, and Tai20b are exact, as they are equal to the optimal objective values of the QAP. However the CPU time required by RLT2 to obtain these bounds is longer than the CPU time required by RRLT2. The bounds of the RRLT2 for the remaining problems are slightly tighter than the RLT2 bounds, but demand less computational effort except problem Rou15 for which RLT2 requires less CPU time. It should be noted, however, that using RRLT2 the bound 351 106 for rou15 achieved in less than 1500 s. For the cQAP instances, as we can observe from Table 2, the RLT1, on average, is slightly better than the RRLT1 in both the bound tightness and required CPU time, but for all 20 instances, the lower bound obtained from the RLT2 and RRLT2 are equal to the objective value of the cQAPs. However, in terms of the required CPU time the RRLT2 outperforms RLT2 for almost all instances.

5. Conclusions

In this paper we proposed a revised form of the Reformulation Linearization Technique for the Quadratic Assignment Problem without destroying the problem's structure. Our experimental results show that, by increasing the level of the RLT, solving the revised RLT representation provides a lower bound as strong as the bound obtained by the RLT, but with less computational effort.

References

[1] E.L. Lawler, The quadratic assignment problem, Manage. Sci. 9 (1963) 586–599.
[2] T.C. Koopmans, M. Beckmann, Assignment problems and the location of economic activities, Econometrica (1957) 53–76.
[3] L. Steinberg, The backboard wiring problem: a placement algorithm, SIAM Rev. 3 (1961) 37–50.
[4] M. Pollatschek, N. Gershoni, Y. Radday, Optimization of the typewriter keyboard by simulation, Angew. Inform. 17 (1976) 438–439.
[5] A. Geoffrion, G. Graves, Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/LP approach, Oper. Res. 24 (1976) 595–610.
[6] G.G. Polak, On a special case of the quadratic assignment problem with an application to storage-and-retrieval devices, Ann. Oper. Res. 138 (2005) 223–233.
[7] E.M. Loiola, N.M.M. De Abreu, P.O. Boaventura-Netto, P. Hahn, T. Querido, A survey for the quadratic assignment problem, European J. Oper. Res. 176 (2007) 657–690.
[8] A. Frieze, J. Yadegar, On the quadratic assignment problem, Discrete Appl. Math. 5 (1983) 89–98.
[9] P. Carraresi, F. Malucelli, A new lower bound for the quadratic assignment problem, Oper. Res. 40 (1992) S22–S27.
[10] P. Carraresi, F. Malucelli, A reformulation scheme and new lower bounds for the QAP, in: Quadratic Assignment and Related Problems, 1994, pp. 147–160.
[11] W.P. Adams, T.A. Johnson, Improved linear programming-based lower bounds for the quadratic assignment problem, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 16 (1994) 43–77.
[12] S.E. Karisch, E. Cela, J. Clausen, T. Espersen, A dual framework for lower bounds of the quadratic assignment problem based on linearization, Computing 63 (1999) 351–403.
[13] P. Hahn, T. Grant, Lower bounds for the quadratic assignment problem based upon a dual formulation, Oper. Res. 46 (1998) 912–922.
[14] K.M. Anstreicher, N.W. Brixius, A new bound for the quadratic assignment problem based on convex quadratic programming, Math. Program. 89 (2001) 341–357.
[15] W.P. Adams, M. Guignard, P.M. Hahn, W.L. Hightower, A level-2 reformulation-linearization technique bound for the quadratic assignment problem, European J. Oper. Res. 180 (2007) 983–996.
[16] P.M. Hahn, Y.R. Zhu, M. Guignard, W.L. Hightower, M.J. Saltzman, A level-3 reformulation-linearization technique-based bound for the quadratic assignment problem, INFORMS J. Comput. 24 (2012) 202–209.
[17] W.P. Adams, H.D. Sherali, A tight linearization and an algorithm for zero–one quadratic programming problems, Manag. Sci. 32 (1986) 1274–1290.
[18] W.P. Adams, H.D. Sherali, Linearization strategies for a class of zero–one mixed integer programming problems, Oper. Res. 38 (1990) 217–226.
[19] W.P. Adams, R.J. Forrester, F.W. Glover, Comparisons and enhancement strategies for linearizing mixed 0–1 quadratic programs, Discrete Optim. 1 (2004) 99–120.
[20] M.G. Resende, K. Ramakrishnan, Z. Drezner, Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming, Oper. Res. 43 (1995) 781–791.
[21] K.G. Ramakrishnan, M.G.C. Resende, B. Ramachandran, J.F. Pekny, Tight QAP bounds via linear programming, in: P.M. Pardalos, A. Migdalas, R.E. Burkard (Eds.), Combinatorial and Global Optimization, World Scientific Publishing, Singapore, 2002, pp. 297–303 (Chapter 19).
[22] Y.R. Zhu, Recent advances and challenges in quadratic assignment and related problems (Ph.D. thesis), University of Pennsylvania, 2007.
[23] R. Burkard, S. Karisch, F. Rendl, QAPLIB—a quadratic assignment problem library, J. Global Optim. 10 (1997) 391–403.
[24] M. Drugan, Generating QAP instances with known optimum solution and additively decomposable cost function, J. Comb. Optim. (2013) 1–35.