A linear formulation with $O(n^2)$ variables for the quadratic assignment problem

Serigne Gueye · Philippe Michelon

Received: date / Accepted: date

Abstract We present an integer linear formulation that uses the so-called “distance variables” to solve the quadratic assignment problem (QAP). The model involves $O(n^2)$ variables. Valid equalities and inequalities are additionally proposed. We further improved the model by using metric properties as well as an algebraic characterization of the Manhattan distance matrices that Mittelman and Peng [28] recently proved for the special case of problems on grid graphs. We numerically tested the lower bound provided by the linear relaxation using instances of the quadratic assignment problem library (QAPLIB). Our results are compared with the best known lower bounds. For all instances, the formulation gives a very competitive lower bound in a short computational time, improving seven best lower bounds of QAPLIB instances for which no optimality proofs exist.

Keywords quadratic assignment problem, distance, integer programming, cutting planes.

1 Introduction

The quadratic assignment problem (QAP), first introduced by Koopmans and Beckmann [26] in 1957, consists in assigning $n$ entities to $n$ locations, which are denoted by $k$ and $l$, respectively, and separated by a distance of $d_{kl}$, which may differ from $d_{lk}$. Furthermore, entities $i$ and $j$ must exchange quantities of a given product $f_{ij}$ or $f_{ji}$. The cost of assigning $i$ to $k$ is denoted by $c_{ik}$. An assignment also induces a product routing cost, which is assumed proportional to the product quantities to be exchanged and to the distance that separates the entities. The standard mathematical program for the QAP is based on the binary variables $x_{ik}$:
$x_{ik} = \begin{cases} 1 & \text{if entity } i \text{ is assigned to location } k, \\ 0 & \text{otherwise}. \end{cases}$

With these variables, the problem consisting of

$$\min \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik} x_{ik} + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ij} d_{kl} x_{ik} x_{jl}$$

such that

$$\sum_{k=1}^{n} x_{ik} = 1 \quad \forall i = 1, \ldots, n$$

$$\sum_{i=1}^{n} x_{ik} = 1 \quad \forall k = 1, \ldots, n$$

$$x_{ik} \in \{0, 1\} \quad \forall i, k = 1, \ldots, n$$

belongs to the class of 0–1 quadratic problems. The QAP is NP-hard [17]; it is considered one of the most difficult problems in this category, especially for an exact solution. This difficulty is illustrated by the lack of optimality proofs for the best known feasible solutions of the 32 instances of the quadratic assignment problem library (QAPLIB) collected by Burkard, Çela, Karisch, and Rendl in 1997 [8].

Numerous methods have been used to address this problem; they may be roughly subdivided into metaheuristic methods providing suboptimal solutions, lower bounding techniques including linear or semidefinite programming (SDP) relaxations, and exact methods consisting in branch-and-bound schemes. The branch-and-bound and lower bounding techniques are highly interconnected because the former uses the bound provided by the latter.

Exact solution methods and bounding techniques may be classified into two groups: linearization techniques, which either exploit or are derived from a linear reformulation of the problem, and approaches that use a trace reformulation of the problem.

The projection bound method (PB) of Hadley, Rendl, and Wolkowicz [21] is a bounding technique that is based on the trace formulation. Karisch and Rendl [23] exploited the PB technique in their triangular decomposition method (TD) to compute a lower bound for the remaining part of a quadratic objective function’s decomposition into paths and triangles. The SDP relaxation methods studied in [24] [38] [39] also use the QAP trace formulation. SDP bounds are competitive. But they remain difficult to use as basic bounding procedures within branch-and-bound algorithms because they require considerable computational times.
Nevertheless, Mittelman and Peng [28] have recently achieved a breakthrough towards solving particular cases of \(QAP\) problems with SDP bounds. For \(QAP\) instances with Hamming or Manhattan distances, the authors were able to prove an important condition that is necessary to characterize any real matrix representing Manhattan or Hamming distances of a hypercube. Based on this condition, they propose two SDP formulations, which use a splitting process of the distance matrix. Their study gives some of the best currently known lower bounds—especially for large-size instances starting from \(n = 81—\)which are computed in reasonably short times (compared with other SDP bounds). Although we propose an integer linear formulation (not an SDP one) in this paper, we exploit Mittelman and Peng’s necessary condition to improve our model and, for some instances, to also improve the best known lower bounds.

Linearization techniques are based on the idea of replacing each quadratic term (or several terms) with new variables and then adding linear constraints to produce an integer linear reformulation that is equivalent to the original quadratic formulation. This idea was first implemented in the context of pseudo-boolean minimization by Fortet [14] [15], who introduces the variable \(z_{ikjl}\) to linearize \(x_{ik}x_{jl}\). Additional constraints make this equality true for any integer solution. These variables have been used by Lawler [27], Frieze & Yadegar [16], and Adams & Johnson [2], who applied Adams & Sherali’s [3] reformulation linearization technique (RLT) to \(QAPs\). The model resulting from a level-1 RLT proved superior to those by Lawler and Frieze-Yadegar, as well as to the Gilmore-Lawler bound [27][18]. The level-1 RLT involves \(O(n^4)\) variables. But, because of its particular structure, it is possible to obtain good bounds by using a dual ascent method on a Lagrangian relaxation scheme. The numerical experiments for level-1 RLTs are limited to Nugent instances [30] of up to size 20. Analogous to the level-1 RLT algorithm, the same technique has been applied to a level-2 [1] and, more recently, a level-3 RLT [22] in a distributed version [20]. The linear model, which involves \(O(n^6)\) variables for level 2 and \(O(n^9)\) variables for level 3, forbids a direct computation of the model and requires a Lagrangian relaxation (as in RLT 1) as well as distributed algorithms. For all levels, the maximal size of instances for which numerical results have been obtained does not exceed 30. The bounds are excellent for these instances. For example, optimal values are reached for four instances of QAPLIB. However, very large computational times are necessary and/or considerable computing resources are required. For example, to obtain a bound at 3% deviation of the optimal value of an instance of size 30 (\(nug30\) [8]), using the distributed version of a level-3 RLT [20], requires 100 host machines working in parallel with a total computational time of about 63 hours.

Recently (2011), Fischetti, Monaci, and Savagnin [12] proposed three powerful ideas that provide exact solutions for four previously unsolved instances (including an instance of size 128) of the Eschermann and Wunderlich [11] type. The first idea concerns symmetry properties derived from the concept of clone entities, which are a set of entities whose locations may be interchanged without changing the corresponding value. The authors then defined clone clusters and used them to reformulate the problem in a rectangular \(QAP\) where the number of locations (denoted by \(m\)) is greater than the number of entities (denoted by \(n\)). They then applied the Kaufman and Broeckx linearization [25] to this reformulation. Note
that the type of variable used in this linearization was also proposed by Glover [19] for non-linear problems. This implies $O(n^2)$ variables for square QAPs and $O(n \times m)$ variables for the rectangular case. They then used decomposition strategies of the flow matrix to solve the instance of size 128.

Our study aims to propose a linear formulation of the QAP that also induces additional $O(n^2)$ variables. The formulation is based on the so-called “distance variables” previously used by Caprara and Salazar-González [10] and by Caprara, Letchford, and Salazar-González [9] for the linear arrangement problem, a particular case of QAP. In the linear arrangement problem, the distance matrix corresponds to node distances in a simple graph representing a path with a unitary edge weight; the flow matrix is binary. We have extended the use of these variables to QAPs and are able to present extensive numerical results.

The paper is organized as follows: Section 2, which addresses a “general” quadratic assignment problem, presents the definition of the distance variables. In Section 3, we analyze the polyhedra induced by the formulation. As a result, two valid constraints are added to the model. Section 4 considers the special case of QAPs with Manhattan distance matrices. This particularity makes it possible to add stronger facets. The first facet is derived from the metric and grid properties; the second facet (shown in Section 7) is based on the necessary condition proved by Mittelman and Peng [28] for Manhattan distance matrices. All of these inequalities are generated by cutting plane algorithms. Concerning the quality of the resulting lower bounds, the numerical experiments we performed are described in Section 6 and Section 8.

2 A linear formulation with $O(n^2)$ variables

For all entities $i$ and $j$, the distance variables $D_{ij}$ are defined as

$$D_{ij} = \sum_{k=1}^{n} \sum_{l=1}^{n} d_{kl} x_{ik} x_{jl}, \quad \forall \ i, j = 1, 2, ..., n.$$  \hspace{1cm} (1)

Note that, for all fixed locations $k_0$ and $l_0$, taking $x_{ik_0} = 1$ and $x_{jl_0} = 1$ implies $D_{ij} = d_{k_0 l_0}$. Thus, $D_{ij}$ represents the distance between entities $i$ and $j$, which depends on their respective locations.

Alternatively, it is possible to introduce the variables

$$F_{k,l} = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} x_{ik} x_{jl}, \quad \forall \ k, l = 1, 2, ..., n,$$

representing the quantity of products from $k$ to $l$. This paper is limited to presenting the “distance” variable formulation.

With these variables, the QAP may be formulated as the following mixed-integer linear program:
A linear formulation with $O(n^2)$ variables for the quadratic assignment problem

\[ \text{(MIP)}: \text{Min} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik}x_{ik} + \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} f_{ij}D_{ij} \quad (2) \]

such that
\[ \sum_{i=1}^{n} x_{ik} = 1 \quad \forall i = 1, ..., n \quad (3) \]
\[ \sum_{k=1}^{n} x_{ik} = 1 \quad \forall k = 1, ..., n \quad (4) \]
\[ D_{ij} \geq d_{kl}(x_{ik} + x_{jl} - 1) \quad \forall i, j, k, l = 1, ..., n, \ i \neq j, \ k \neq l \quad (5) \]
\[ x_{ik} \in \{0, 1\} \quad \forall i, k = 1, ..., n \quad (6) \]
\[ D_{ij} \geq 0 \quad \forall i, j = 1, ..., n, \ i \neq j, \quad (7) \]

where the constraints (5) are a linearization of the following relaxation of Equation 1:
\[ D_{ij} \geq \sum_{k=1}^{n} \sum_{l=1}^{n} d_{kl}x_{ik}x_{jl}, \forall i, j = 1, 2, ..., n. \quad (8) \]

In fact, for any feasible solution, we can easily verify that the constraints (5) imply that $D_{ij}$ is greater than the distance between $i$ and $j$. Because we are minimizing and because $f_{ij} \geq 0$, $D_{ij}$ is precisely equal to this distance. Our linear model for the quadratic assignment problem has a relatively small number of $(O(n^2))$ variables; there are, however, two drawbacks. The first limitation (previously observed by Caprara, Letchford, and Salazar-González [9]) concerns the linear relaxation, which provides a poor quality lower bound.

**Theorem 1** If $c_{ik} = 0$ for all $i$ and $k$, then the linear relaxation of the problem above has an optimal value equal to 0.

**Proof** It is sufficient to observe that $x_{ik} = \frac{1}{n}$ is feasible for all $i$ and $k$. Constraints 5 then reduce to $D_{ij} \geq \frac{2n^2}{n} - d_{kl}$, so that $D_{ij} = 0$ because of (7). □

**Remark 1** The linear relaxation bound can be slightly improved by substituting $D_{ij} \geq 0$ for $D_{ij} \geq d$, where $d$ is the smallest distance between two locations.

The second limitation concerns the $(O(n^4))$ number of constraints (5) that should be reduced. In the following section, we strengthen the model by reducing the number of constraints and by finding valid inequalities. In fact, besides its low number of variables, the particular structure of our model makes it easy to derive some of these inequalities.

3 Polyhedral analysis

The set of feasible (MIP) solutions can be equivalently described as the set of vectors $P$ defined by
\[ \mathcal{P} = \{ (x, D) \mid \sum_{i=1}^{n} x_{ik} = 1, \forall i = 1, \ldots, n \quad (3), \]
\[ \sum_{k=1}^{n} x_{ik} = 1, \forall k = 1, \ldots, n \quad (4), \]
\[ x_{ik} \in \{0, 1\}, \forall i, k = 1, \ldots, n \quad (6), \]
\[ D_{ij} \geq \sum_{k=1}^{n} \sum_{l=1}^{n} d_{kl} x_{ik} x_{jl}, \forall i \neq j = 1, 2, \ldots, n \quad (8), \]

where \( x = \{ x_{ik} \}_{1 \leq i, k \leq n} \) and \( D = \{ D_{ij} \}_{1 \leq i \neq j \leq n} \).

Recall that the \( O(n^4) \) linear constraints (5) in \((MIP)\) are obtained from a linearization of the quadratic inequalities (8).

We present new valid \( O(n^3) \) inequalities for \( \text{conv}(\mathcal{P}) \) that are equivalent to the constraints (5). We are therefore able to substantially reduce the problem size while keeping the optimal integer solution unchanged. Some preliminary polyhedral results are needed to introduce these inequalities.

**Lemma 1** We have \( \dim(\text{conv}(\mathcal{P})) = (n-1)^2 + n(n-1) \).

**Proof.** Separate \( \text{conv}(\mathcal{P}) \) into two polyhedra: the assignment polytope \( \text{conv}(\mathcal{A}) \) and the distance polyhedron \( \text{conv}(\mathcal{D}) \).

\[ \mathcal{A} = \{ x \mid (3); (4); (6) \} \]

and

\[ \mathcal{D} = \{ D \mid \exists x \in \mathcal{A} \text{ such that } (8) \}. \]

By using linear algebra results of an affine space, we have \( \dim(\text{conv}(\mathcal{A}) \times \text{conv}(\mathcal{D})) = \dim(\text{conv}(\mathcal{A})) + \dim(\text{conv}(\mathcal{D})) \).

Yet \( \dim(\text{conv}(\mathcal{A})) = (n-1)^2 \) (see [6] [5]), and \( \text{conv}(\mathcal{D}) \) is a full-dimensional, unbounded polyhedron (i.e., \( \dim(\text{conv}(\mathcal{D})) = n(n-1) \), see [9]).

Because \( \text{conv}(\mathcal{P}) \subset \text{conv}(\mathcal{A}) \times \text{conv}(\mathcal{D}) \), it follows that the lemma can be proved by finding \((n-1)^2 + n(n-1) + 1\) affinely independent points of \( P \). Let \( m = (n-1)^2 \), and let \( x^0, x^1, \ldots, x^m \) be affinely independent points of \( \text{conv}(\mathcal{A}) \).

From each \( x^u \) \((u = 0, 1, \ldots, m)\), we can derive the point \( D^u = \{ D^u_{ij} \}_{1 \leq i \neq j \leq n} \) \((u = 0, 1, \ldots, m)\) corresponding to the exact distance between the entities \( i \) and \( j \). The points \( x^u \in \mathcal{A} \) can thus be extended to the points \( (x^u, D^u) \in \mathcal{A} \times \mathcal{D}, u = 0, 1, \ldots, m \).

Consider the point \( D^0 = \{ D^0_{ij} \}_{1 \leq i \neq j \leq n} \). Define \( D^{0v} \) \((v = 0, 1, \ldots, q-1)\), where \( q = n(n-1) \) for \( q \) points obtained from \( D^0 \) by sequentially fixing one coordinate of \( D^0 \) to a large value \( M \) without changing the remaining \( q-1 \) coordinates. As mentioned in [9], \( D^0 \) and \( D^{0v} \) \((v = 0, 1, \ldots, q-1)\) are affinely independent for a sufficiently large value of \( M \).
Now let $\alpha_0, \alpha_{0v} \ (v = 0, 1, \ldots, q)$, $\alpha_1, \ldots, \alpha_m$ be some scalars satisfying

$$\alpha_0(x^0, D^0) + \sum_{v=0}^{q-1} \alpha_{0v}(x^0, D^{0v}) + \sum_{u=1}^{m} \alpha_u(x^u, D^u) = 0 \quad (9)$$

$$\alpha_0 + \sum_{v=0}^{q-1} \alpha_{0v} + \sum_{u=1}^{m} \alpha_k = 0. \quad (10)$$

Note that $\beta_0 = \alpha_0 + \sum_{v=0}^{q-1} \alpha_{0v}$.

$$(10) \Rightarrow \beta_0 + \sum_{u=1}^{m} \alpha_u = 0.$$

$$(9) \Rightarrow \beta_0 x^0 + \sum_{u=1}^{m} \alpha_u x^u = 0.$$

Because $x^0, x^1, \ldots, x^m$ are affinely independent, we have $\beta_0 = \alpha_1 = \ldots = \alpha_m = 0$.

Yet $(9) \Rightarrow \alpha_0 D^0 + \sum_{v=0}^{q-1} \alpha_{0v} D^{0v} = 0$, and the affine independence of $D^0$ and $D^{0v}$ ($v = 0, 1, \ldots, q-1$) implies that $\alpha_0 = \alpha_{00} = \ldots = \alpha_{0(q-1)} = 0$.

Therefore, $(x^0, D^0); (x^0, D^{00}); \ldots; (x^0, D^{0(q-1)}); (x^1, D^1); \ldots; (x^m, D^m)$ are $m + q = (n - 1)^2 + n(n - 1) + 1$ affinely independent points.

Let $i$ be a given entity, and let $k$ be a given location. Consider the subdomain

$$\mathcal{P}_{ik} = \{(x, D) \mid (3); (4); (6); (8) \quad x_{ik'} = 0 \quad \forall k' \neq k\},$$

defining the feasible solutions in which $i$ is located on $k$. Using similar arguments as in the lemma above, we can prove that $\dim(\text{conv}(\mathcal{P}_{ik})) = (n - 2)^2 + n(n - 1)$, which shows that the inequalities below are facets of $\text{conv}(\mathcal{P}_{ik})$.

**Lemma 2** For any entity $i$ and any location $k$,

$$D_{ij} \geq \sum_{l=1}^{n} d_{kl}x_{jl}, \forall j = 1, 2, \ldots, n$$

are facets of $\text{conv}(\mathcal{P}_{ik})$.

**Proof**. First, note that $x_{ik'} = 0 \forall k' \neq k \Leftrightarrow x_{ik} = 1$. Taking the inequality (8) into account, it follows that, for any fixed $j$,

$$D_{ij} \geq \sum_{l=1}^{n} d_{kl}x_{jl}$$

is valid.
To prove the facet property, it suffices to show that the corresponding face is of a dimension \( \dim(\text{conv}(P_{ik})) - 1 \) by exhibiting \((n - 2)^2 + n(n - 1)\) affinely independent points of the face.

As for \( \text{conv}(P) \), \( \text{conv}(P_{ik}) \) may be separated into an assignment polytope \( \text{conv}(A_{ik}) \) and a distance polyhedron \( \text{conv}(D_{ik}) \), in which the constraints \( x_{ik'} = 0 \) \( \forall k' \neq k \) are added.

In this case, \( \dim(\text{conv}(A_{ik})) = (n - 2)^2 \), and \( \text{conv}(D_{ik}) \) remains a full-dimensional, unbounded polyhedron (i.e., \( \text{conv}(D_{ik}) = n(n - 1) \)).

Let \( x^0, x^1, ..., x^m \) be affinely independent points of \( \text{conv}(A_{ik}) \), where \( m = (n - 2)^2 \). We again extend each \( x^u (u = 0, 1, ..., m) \) to the points \( (x^u, D^u) \), where \( D^u \) denotes the exact distance corresponding to the location \( x^u \). Note that \( D^u_{ij} = \sum_{l=1}^{n} d_{kl} x_{jl} \forall u \).

Let \( (x^0, D^{0v}) (v = 0, ..., q - 2) \), where \( q = n(n - 1) \), which is obtained from \( (x^0, D^0) \) by sequentially fixing one of the \( D^0 \) coordinates—other than \( D^0_{ij} \)—to a sufficiently large value \( M \) while keeping the others unchanged. As observed in the preceding proof, \( D^0 \) and \( D^{0v} (v = 0, ..., q - 2) \) are affinely independent, by which \( (x^0, D^0) \) and \( (x^u, D^u) (u = 1, ..., m) \) are \( m + q - 1 = (n - 2)^2 + n(n - 1) \) affinely independent points of the face.

We can now introduce the first valid inequalities of \( \text{conv}(P) \).

**Theorem 2** The following equalities are valid inequalities of \( \text{conv}(P) \):

\[
D_{ij} \geq \sum_{l=1}^{n} d_{kl} x_{jl} + \sum_{k'=1}^{n} \lambda_{kk'} x_{ik'} \quad \forall i \neq j, k,
\]

where \( \lambda_{kk'} = \min_{1 \leq k' \neq k' \leq n} d_{k'k'} - d_{kk} \).

**Proof.** For the given entities \( i \) and \( j \) \( (i \neq j) \) and a location \( k \), we know from Lemma 2 that

\[
D_{ij} \geq \sum_{l=1}^{n} d_{kl} x_{jl} \forall (x, D) \in P_{ik}
\]

is a facet for \( \text{conv}(P_{ik}) \). Let \( k' \neq k \) be any location, and

\[
P_{ikk'} = \{(x, D) | (3); (4); (6); \}
\]

\[
x_{ik'} = 1
\]

By using lifting techniques, we can find the best coefficient \( \lambda_{kk'} \) for which

\[
D_{ij} \geq \sum_{l=1}^{n} d_{kl} x_{jl} + \lambda_{kk'} x_{ik'} \forall x \in P_{ikk'}.
\]
We have
\[ \lambda_{kk'} = \text{Min} \ D_{ij} - \sum_{l=1}^{n} d_{kl}x_{jl} \]
such that \((x, D) \in P_{ikk'}\).

The optimal value of this problem is precisely \(\lambda_{kk'} = \text{Min}_{1 \leq k' \neq l \leq n} d_{k'l'} - d_{kl}\).

We obtain the corresponding valid inequalities by repeating the same lifting procedure for all variables \(x_{ik'}\) with \(k' \neq k\).

According to integer programming theory (see [29]), each lifting step increases the dimension of the resulting face by one unit. Because, following Lemma 2,
\[ D_{ij} \geq \sum_{l=1}^{n} d_{kl}x_{jl}, \forall j = 1, 2, ..., n \]
defines a facet of a dimension \((n - 2)^2 + n(n - 1) - 1\) of \(\text{conv}(P_{ik})\), it follows that
\[ D_{ij} \geq \sum_{l=1}^{n} d_{kl}x_{jl} + \sum_{k' \neq k} \lambda_{kk'}x_{ik'} \]
defines a face of dimension \((n - 2)^2 + (n + 1)(n - 1) - 1 = 2(n - 1)^2\).

But according to Lemma 1, \(\text{dim}(\text{conv}(P)) = (n - 1)^2 + n(n - 1)\). So the inequalities above define facets of \(\text{conv}(P)\) if and only if \(2(n - 1)^2 = (n - 1)^2 + n(n - 1) - 1\), which happens only when \(n = 2\). For greater sizes (i.e., \(n > 2\)), we obtain valid inequalities only.

Nevertheless, replacing the constraints (5) by (11) yields an equivalent formulation while reducing the constraint complexity by a factor \(n\) (i.e., \(O(n^3)\) instead of \(O(n^4)\)).

To further strengthen our formulation, we consider the following more precise set of integer feasible solutions instead of \(P\):
\[ P^w = \{(x, D) \mid \text{ (3); (4); (6); (12)} \} \]

Clearly, \(P^w \subset P\), by which \(\text{conv}(P^w) \subset \text{conv}(P)\). \(P^w\) is also included in another set described below.

**Theorem 3** Let \(d_k = \sum_{l=1}^{n} d_{kl}, \forall k = 1, 2, ..., n, \) and
\[ \sum_{j=1}^{n} D_{ij} = \sum_{k=1}^{n} d_kx_{ik}, \forall i \neq j = 1, 2, ..., n. \]

Defining \(F = \{(x, D) \mid \text{ (3); (4); (6); (12)} \}, we have \(P^w \subset F\).
Proof. Let \((x,D) \in \mathcal{P}^n\). It satisfies

\[ D_{ij} = \sum_{k=1}^{n} \sum_{l=1}^{n} d_{kl} x_{ik} x_{jl}, \forall i, j. \]

Thus

\[ \sum_{j=1}^{n} D_{ij} = \sum_{k=1}^{n} \left[ \sum_{l=1}^{n} d_{kl} \sum_{j=1}^{n} x_{jl} \right] x_{ik}. \]

It follows with constraint (4) that

\[ \sum_{j=1}^{n} D_{ij} = \sum_{k=1}^{n} \left[ \sum_{l=1}^{n} d_{kl} \right] x_{ik} = \sum_{k=1}^{n} d_k x_{ik}. \]

\[ \square \]

\[ \mathcal{P}^n \subset \mathcal{P} \cap \mathcal{F} \subset \mathcal{P} \Rightarrow \text{conv}(\mathcal{P}^n) \subset \text{conv}(\mathcal{P} \cap \mathcal{F}) \subset \text{conv}(\mathcal{P}). \]

Thus, adding the equalities in \(\mathcal{P}^n\) allows us to get closer to \(\text{conv}(\mathcal{P}^n)\), which significantly improves the formulation. This improved formulation is denoted by \((MIP^+)\).

\[(MIP^+) : \text{Min} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik} x_{ik} + \sum_{i=1}^{n} \sum_{j \neq i}^{n} f_{ij} D_{ij} \]

such that \((3), (4), (6), (7), (11), (12)\).

Up to this point, we have not made any assumptions concerning the structure of the distance matrix \(d = \{d_{kl}\}_{1 \leq k, l \leq n}\). We now consider that \(d\) represents Manhattan distances on a grid graph for the following reasons. The first reason concerns finding new facets (or valid inequalities) with the help of well-defined structures. This is a more difficult task if we consider a general problem. The second reason, mentioned in the beginning of this paper, concerns viewing the linear arrangement problem (studied by Caprara et al. [9]) as a special case of QAP in which the assignment has to be made on a line (a grid graph with one line). Thus, it seems logical to extend some of the known polyhedral results to any grid.

4 The Manhattan distance matrix

We assume that \(d\) represents Manhattan distances of a rectangular grid graph. Let \(P_r\) and \(P_s\) be two path graphs with \(r\) and \(s\) nodes respectively. The rectangular grid graph \(G_{(r,s)}\) of \(r \times s\) nodes is defined as the graph Cartesian product of \(P_r\) and \(P_s\) (i.e., \(G_{(r,s)} = P_r \times P_s\)). Here is an example of \(G_{(3,3)}\): 
A linear formulation with $O(n^2)$ variables for the quadratic assignment problem

Remark 2. The distance between any adjacent nodes of such graphs is 1. In addition, note that there are no elementary cycles of size 3.

Using $D$ to denote the distance variable matrix, we can see that $D$ is symmetrical (i.e., $D_{ij} = D_{ji}$) with zero diagonals (i.e., $D_{ii} = 0$). Thus, the formulation ($MIP^+$) can be simplified as follows.

$$(MIP^+) : \text{Min} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik}x_{ik} + \sum_{i=1}^{n} \sum_{j=i+1}^{n} (f_{ij} + f_{ji})D_{ij}$$

such that (3), (4), (6)

$$D_{ij} \geq 0 \quad \forall \ 1 \leq i < j \leq n \quad (7)$$

$$D_{ij} \geq \sum_{l=1}^{n} d_{kl}x_{jl} + \sum_{k' \neq k} \lambda_{kk'}x_{ik}x_{ik'} \quad \forall \ 1 \leq i < j \leq n \quad (11)$$

$$\sum_{j=1}^{n} D_{ij} = \sum_{k=1}^{n} d_kx_{ik} \quad \forall \ 1 \leq i < j \leq n \quad (12)$$

Taking the symmetries into account, we may rewrite the QAP feasible set as

$$\mathcal{P} = \{(x,D) \mid (3), (4), (6) \text{ such that (3),(4),(6)} \}$$

$$\mathcal{D} = \{D \mid \exists x \in \mathcal{A}, D_{ij} \geq \sum_{k=1}^{n} \sum_{l=1}^{n} d_{kl}x_{ik}x_{jl}; \forall \ 1 \leq i < j \leq n \ (8)\}.$$
The assumption that $d$ represents the Manhattan distances of a rectangular grid graph implies that it respects the metric property, in particular the well-known triangular inequalities.

**Theorem 4** Let $i, j, h$ satisfy $1 \leq i < j < h \leq n$. The following triangular inequalities are facets of $\text{conv}(P)$:

\[
\begin{align*}
D_{ij} &\leq D_{ih} + D_{jh}, \\
D_{ih} &\leq D_{ij} + D_{jh}, \\
D_{jh} &\leq D_{ij} + D_{ih}.
\end{align*}
\]

**Proof.** The validity is directly derived from the metric property. We focus on showing that these inequalities are facets. For symmetry reasons, it is sufficient to show that (13) is a facet of $\text{conv}(P)$.

Because $\dim(\text{conv}(A)) = (n-1)^2$, there exist $m+1$ affinely independent points $x^0, x^1, \ldots, x^m \in A$, where $m = (n-1)^2$.

We define $i_0, j_0, h_0$ as three fixed entities with $i_0 < j_0 < h_0$. Let

$$
F = \{(x, D) \in \text{conv}(P) \mid D_{i_0j_0} = D_{i_0h_0} + D_{j_0h_0}\}.
$$

Because the variables $\{D_{ij}\}_{1 \leq i < j \leq n}$ are not upper bounded, we can always extend each point $x^k (k = 0, 1, \ldots, m)$ to a point $(x^k, D^k) \in F$ by fixing appropriate values to the $D^k$ components $D_{i_0h_0}, D_{i_0j_0}$, and $D_{j_0h_0}$.

Let us consider $D^0$. Define the points $D^{rs} = \{D_{ij}^{rs}\}_{1 \leq i < j \leq n} \in F$, where $1 \leq r < s \leq n$ and $(r, s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}$, as

$$
D_{ij}^{rs} = \begin{cases} 
D_{ij}^0, & \text{if } (i, j) \neq (r, s), \\
D_{ij}^0 + M, & \text{otherwise}
\end{cases}
$$

where $M$ is any positive real number.

Define the points $D^{m+1} = \{D^{m+1}_{ij}\}_{1 \leq i < j \leq n}$ as

$$
D^{m+1}_{ij} = \begin{cases} 
D_{ij}^0, & \text{if } (i, j) \notin \{(i_0, j_0); (i_0, h_0)\}, \\
D_{ij}^0 + M, & \text{otherwise}
\end{cases}
$$

and $D^{m+2} = \{D^{m+2}_{ij}\}_{1 \leq i < j \leq n}$ as

$$
D^{m+2}_{ij} = \begin{cases} 
D_{ij}^0, & \text{if } (i, j) \notin \{(i_0, j_0); (j_0, h_0)\}, \\
D_{ij}^0 + M, & \text{otherwise.}
\end{cases}
$$

The two points above are also contained in $F$.

In Appendix A, we show that
A linear formulation with $O(n^2)$ variables for the quadratic assignment problem

$D^0$, $D^{rs}$ \(1 \leq r < s \leq n, (r,s) \notin \{(i_0,j_0); (i_0,h_0); (j_0,h_0)\}$,
$D^{m+1}$, $D^{m+2}$

are \(n(n-1)/2\) affinely independent points.

As a consequence, the extended set of points of $\mathcal{F}$ is affinely independent:

\[
(x^0, D^0), \quad (x^0, D^{rs}), \quad (1 \leq r < s \leq n, (r,s) \notin \{(i_0,j_0); (i_0,h_0); (j_0,h_0)\}),
\]
\[
(x^0, D^{m+1}), \quad (x^0, D^{m+2}).
\]

And this is also true for

\[
(x^0, D^0), \quad (x^0, D^{rs}), \quad (1 \leq r < s \leq n, (r,s) \notin \{(i_0,j_0); (i_0,h_0); (j_0,h_0)\}),
\]
\[
(x^k, D^{m+1}), \quad (x^k, D^{m+2}),
\]
\[
(x^k, D^k) (1 \leq k \leq m).
\]

We exhibited \((n-1)^2 + n(n-1)/2\) affinely independent points of $\mathcal{F}$, showing that $\dim(\mathcal{F}) = (n-1)^2 + n(n-1)/2 - 1$. Thus, the corresponding face is a facet.

In addition to exhibiting these triangular facets, the grid structure facilitates an extension of the rank inequalities (or the clique inequalities) proposed by [9] for the linear arrangement problem.

**Theorem 5** Let $i, j, h$ satisfying $1 \leq i < j < h \leq m$. The following inequalities are facets of $\text{conv}(\mathcal{D})$:

\[
D_{ij} + D_{ih} + D_{jh} \geq 4. \quad (16)
\]

**Proof.** Starting with the validity, we first suppose, by contradiction, that $D_{ij} + D_{ih} + D_{jh} \leq 3$. Then, because $D_{ij} \geq 1$, we have $d_{ij} = d_{ih} = d_{jh} = 1$. It follows that the three entities are located in nodes of an elementary cycle of size 3. This is impossible because a grid graph does not contain any elementary cycles of size 3. Let us now prove that the inequalities are also facets.

We define $i_0, j_0, h_0$ as three fixed entities with $i_0 < j_0 < h_0$. Let

\[
\mathcal{F} = \{D \in \text{conv}(\mathcal{D}) \mid \exists x \in \mathcal{A} \text{ such that } (8), \quad D_{i_0j_0} + D_{i_0h_0} + D_{j_0h_0} = 4\}.
\]

Let $D^0$ be a point in $\mathcal{F}$, and let $M$ be a large positive real value. We may suppose, without loss of generality, that $D^0_{i_0j_0} = 1$, $D^0_{i_0h_0} = 1$, and $D^0_{j_0h_0} = 2$; otherwise we renumber the indices if necessary.

Define the points $D^{rs} = \{D^{rs}_{ij}\}_{1 \leq i < j \leq n}$ in $\mathcal{F}$, where $1 \leq r < s \leq n$ and $(r,s) \notin \{(i_0,j_0); (i_0,h_0); (j_0,h_0)\}$, as

\[
D^{rs}_{ij} = \begin{cases} D^0_{ij}, & \text{if } (i,j) \neq (r,s), \\ D^0_{ij} + M, & \text{otherwise.} \end{cases}
\]
Define $D_{ij} = \{D_{ij}^{i_{0}h_{0}}\}_{1 \leq i < j \leq n}$ as

$$D_{ij}^{i_{0}h_{0}} = \begin{cases} M, & \text{if } (i, j) \notin \{(i_{0}, j_{0}); (i_{0}, h_{0}); (j_{0}, h_{0})\}, \\ 1, & \text{if } (i, j) = (i_{0}, j_{0}), \\ 2, & \text{if } (i, j) = (i_{0}, h_{0}), \\ 1, & \text{if } (i, j) = (j_{0}, h_{0}). \end{cases}$$

and define $D_{ij} = \{D_{ij}^{i_{0}h_{0}}\}_{1 \leq i < j \leq n}$ as

$$D_{ij}^{i_{0}h_{0}} = \begin{cases} M, & \text{if } (i, j) \notin \{(i_{0}, j_{0}); (i_{0}, h_{0}); (j_{0}, h_{0})\}, \\ 2, & \text{if } (i, j) = (i_{0}, j_{0}), \\ 1, & \text{if } (i, j) = (i_{0}, h_{0}), \\ 1, & \text{if } (i, j) = (j_{0}, h_{0}). \end{cases}$$

By construction, $D^{0}, D^{rs}, D_{ij}^{i_{0}h_{0}}, D_{ij}^{i_{0}h_{0}} \in \mathcal{F}$.

Additionally, we show in Appendix B that the points are affinely independent. Thus, $\dim(\text{conv}(\mathcal{F})) = n(n - 1)/2 - 1$, and the corresponding face is a facet of $\text{conv}(\mathcal{D})$.

The theorem may be further generalized. Let $S$ be a subset of entities exchanging a unitary flow between each other (i.e., $f_{ij} = 1$ for all $i, j \in S$). Let $QAP(S)$ be the optimal value of a $QAP$ problem that consists in assigning the entities of $S$ on a grid. We have

$$\sum_{(i, j) \in S} D_{ij} \geq QAP(S),$$

which defines a rank inequality for the $QAP$ as proposed in [4] for the linear arrangement problem. The facets in Theorem 5 are particular cases corresponding to entity subsets of size 3. Some analogous inequalities can be defined with subsets of any size. But as the size increases, the number of inequalities grows exponentially, inducing more complex separation problems. For this reason, we limit our investigation to a size 3, as stated above. As described in Section 6, the generation of these inequalities as well as of the previous facets provides a good lower bound in a short time for both small- and large-size instances.

The formulation denoted by $(MIP^{++})$ is improved by the previous facets; it is summarized below:

$$(MIP^{++}) : \text{Min} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik} x_{ik} + \sum_{i=1}^{n} \sum_{j=i+1}^{n} (f_{ij} + f_{ji}) D_{ij}$$

such that

$$(3), (4), (6), (7), (11), (12), (13), (14), (15), (16)$$

In the following, $(MIP^{++})$ denotes the linear (i.e., continuous) relaxation of $(MIP^{++})$ and $V(MIP^{++})$ denotes its corresponding optimal value (which is the
A linear formulation with $O(n^2)$ variables for the quadratic assignment problem

The constraint complexity of ($MIP^{++}$) is $O(n^3)$, whose largest families are constraints (11), (13), (14), (15), and (16). Because of the polynomial number of constraints, ($MIP^{++}$) may be solved by directly introducing all of the constraints without memory limitation. Nevertheless, substantial gains in processing time can be obtained by using cutting plane algorithms, especially for the constraints in cubic numbers. These kinds of algorithms are used in our numerical experiments; they are described in detail in the following section.

5 Cutting plane algorithms

A cutting plane algorithm is generally used to progressively generate unsatisfied constraints. We have implemented two cutting plane algorithms, both are dedicated to constraints in cubic numbers (11), (13), (14), (15), and (16). Constraints in linear or quadratic numbers were directly integrated into the model. Subsections 5.1 and 5.2 provide additional details regarding these algorithms.

5.1 Algorithm for constraints (11) and (16)

The algorithm for these constraints is the same for each family. The resulting greedy procedure may be summarized as follows:

Algorithm 1: Constraint Generation($F$)

```plaintext
/* $F$ denotes any family of cuts */
while stoppingCriteria == false do
    Solve the linear relaxation of ($MIP^{++}$);
    Find $F' \subseteq F$, a set of unsatisfied cuts;
    Add $F'$ to $MIP^{++}$;
end
```

The “stopping criteria” become true when there are no violated inequalities, or when a maximal number of iterations is reached, or when the difference between two successive optimal values of the linear relaxation ($MIP^{++}$) is below a given threshold.

5.2 Algorithm for triangular inequalities

Although it is possible to apply Algorithm 1 to generate triangular inequalities, we have experimentally observed that it is faster to proceed as follows:

Let $(\pi, D)$ be the current solution of ($MIP^{++}$). Consider the simple complete graph $G = (V, E, D)$, where the nodes represent the entities $i = 1, 2, ..., n$, and each edge $(i, j)$ is weighted by $D_{ij}$ (the components of $D$). To find some violated triangular inequalities, we compute the shortest path between all node couples. For each couple $(i, j)$, let $D'_{ij}$ denote the value corresponding to the shortest path
$i_1, i_2, ..., i_k$, where $k$ is the size of the path, $i_1 = i$, and $i_k = j$.

If $D_{ij} > D_{ij}^\ast$, then the inequality

$$D_{ij} \leq D_{i_1 i_2} + D_{i_2 i_3} + ... + D_{i_{k-1} i_k}$$

is a cut that could be added to the model.

When $k = 3$, this is a triangular inequality. In fact, as long as there is at least one violated triangular inequality, the shortest path computation will find it. In addition, it can also find a path of a size greater than 3.

We can therefore deduce a simple cutting plane algorithm. At each iteration, $(MIP^{++})$ is solved, which gives a current solution $(\pi, \mathcal{D})$. By using the Floyd-Warshall algorithm [13] [36], we compute all of the shortest paths, which gives us a distance matrix $D^\ast$. We compare each component of $D^\ast$ with each component of $\mathcal{D}$. And cuts similar to (17) are added if necessary. The process is iterated until it satisfies a stopping criterion, as in Algorithm 1 above. We refer to this algorithm as the triangular inequalities generation.

5.3 Global algorithm

Recall that Algorithm 1 is applied to each of the constraints (11) and (16), whereas the triangular inequalities are generated as stated above. Reaching the stopping criterion of a given algorithm for a given constraint family does not necessarily imply that no further violated cuts can be found. This is especially true when the process stops because a maximal number of iterations have been reached or because the lower bound did not increase above the threshold. It is therefore advantageous to alternate the elementary cutting plane algorithms described above. This is done in a global (cutting plane) algorithm using the same stopping criteria as above.

Algorithm 2: Global Algorithm

```plaintext
while stoppingCriteria == false do
    Triangular Inequalities Generation;
    Constraint Generation(11);
    Constraint Generation(16);
end
```

The global algorithm sequentially runs each cutting plane algorithm until its own stopping criterion is reached. In the following, we present the first numerical results obtained with this algorithm.

6 Numerical experiments on $(MIP^{++})$

Our aim is to evaluate the quality of the lower bound corresponding to the linear relaxation of $(MIP^{++})$. We conducted an initial series of experiments based en-
A linear formulation with $O(n^2)$ variables for the quadratic assignment problem entirely on the global Algorithm 2 (Table 1).

Nonetheless, we compare our results with the currently published, best known lower bounds obtained with QAPLIB instances [8] for which the distance matrix is given by the shortest path in a grid graph. The corresponding problems are the instances of Nugent, Vollman, and Ruml [30] (nug), Scriabin & Vergin [33] (scr), Skorin-Kapov [34] (sko), Thonemann & Bölte [35] (tho), and Wilhelm & Ward [37] (wil). Each instance is denoted by the first three letters of the first author, followed by its size. For example, nug12 is an instance of Nugent et al. [30] of size 12, corresponding to a QAP that consists in assigning 12 entities in a $3 \times 4$ grid graph.

In the following, we arbitrarily use the word “small” to refer to instances of a size smaller than or equal to 30; “medium” refers to instances of a size strictly larger than 30 and smaller than or equal to 64, and “large” refers to a size that is strictly larger than 64.

For each problem, a best feasible solution and the best lower bound of the optimal value are known for the current standard instances. The equality between these two values leads to an optimality proof. When the two values differ, a branch-and-bound scheme is necessary whose size and computational time depend on the relative deviation between the lower and the upper bound at the root node. No solution was proved optimal for the Skorin-Kapov [34] and the Wilhelm & Ward [37] instances, and only one Thonemann & Bölte [35] instance was solved.

Results are reported in Table 1, where Prob denotes the instance name, $n$ is the number of nodes of the grid, $UB$ is the best known upper bound, and $V(MIP^{++})$ is our lower bound with its corresponding computational time $CPU$ (sec.). We solved $MIP^{++}$ with the IBM Ilog Cplex 12.2 on a DELL R510 server equipped with 125GB of memory and an Intel® Xeon® 64-bit processor with two cores of 2.67GHz each.

We compared our bound with a large set of other bounds:

- SDP bounds by Mittelman and Peng ($SDP_{MittelmanPeng}$) [28], Rendl & Sotirov ($SDP_{RendlSotirov}$) [31], and Zhao et al. ($SDP_{Zhaoetal}$) [39],
- the triangular decomposition method ($TD$) [23],
- level-1, level-2, and level-3 reformulation linearization technique bounds (resp. $RLT_1$ [2], $RLT_2$ [1], and $RLT_3$ [22]),
- a level-3 RLT, performed by parallelization in a distributed environment using up to 100 host machines ($RLT_3Dist$) [20],
- the lift and project approach ($L - P$) by Burer & Vandenbussche [7],
- the interior point method ($IP$) by Resende, Ramakrishnan, and Drezner [32],
- the Gilmore & Lawler bound ($GLB$) [18] [27],
- and the projection method bound ($PB$) [21].

The papers from which these bounds were extracted do not necessarily contain results for all the instances considered here. Medium- or large-size instances may induce memory limitations or very large processing times for the proposed methods. This is the case for all $RLT$ schemes as well as for $L - P$, $IP$, and some $SDP$
bounds. Our investigation is limited to a maximum processing time of 1 day (i.e., 86400 sec). A dash (“−”) indicates that no result is known for a given instance and a given method.

Table 1 shows that, for nug instances, RLT Dist provides the best lower bounds that are optimal for most cases. Nevertheless, our bound is competitive in this class of problems. It provides better results than those obtained for the SDP MittelmanPeng, SDP Zhaoetal, IP, GLB, and PB columns. For Skorin-Kapov [34] (sko∗) instances, our bound ranks third up to sko56, the position just behind the SDP bound of Mittelman and Peng [28]. Note that we obtain a better bound than SDP MittelmanPeng for the sko42 instance. Recall also that the sizes of these instances are currently a limitation for RLT as well as for some SDP bounds.

For the scr and ste instances, the V (MIP ++) technique performed better than any other existing technique. Computational times remained reasonable for short and medium instances, but increased significantly for large ones.

To reduce these computing times, we performed the cutting plane algorithm without constraints (11). Our choice of these constraints was experimentally motivated. Corresponding results are reported in Table 2.

Evidently, eliminating the proposed valid inequalities does not significantly degrade the bound quality even though it significantly reduces the processing times. In fact, we were able to attain the same bound in a shorter computational time for medium- and large-size instances, enabling us to address problems of all sizes (up to 150) within our time limit. For large-size instances, our bound is dominated by other existing methods.

Our processing times nevertheless compare favorably with those of other methods. Table 3 summarizes the processing times (in seconds) of other methods in comparison with our bound V (MIP ++) given in Table 2. Processing times for these bounds are not reported in any uniform manner in QAP literature. Some papers report the time for each instance (for example, [28]), whereas others either indicate some selected times, or give a rough estimate (as for TD). To obtain a fair comparison, these times should be adjusted to compensate for the computing environment, especially for older techniques such as the triangular decomposition method. The processing time for V (MIP ++) is comparatively short and requires fewer computing resources than the distributed algorithms providing the best bounds for the nug instances. Therefore, the ratio between the attained bound and the time required to obtain it may be considered satisfactory.

The V (MIP ++) bound can be further improved by taking advantage of the Manhattan distance matrix characterization by Mittelman et al. In the following section, we describe another cutting plane algorithm that uses this recent theoretical result.
### Global algorithm results

<table>
<thead>
<tr>
<th>Problem</th>
<th>n</th>
<th>V (MP)</th>
<th>CPU (sec)</th>
<th>SDP_MehboubPan</th>
<th>SDP_RendlRendlvonStengel</th>
<th>SDP_BlekhermanParrilo</th>
<th>TD</th>
<th>RLT2</th>
<th>RLT3</th>
<th>RLT4</th>
<th>RLT and - P</th>
<th>LP</th>
<th>GLB</th>
<th>DB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td>n6-g22</td>
<td>12</td>
<td>540.3</td>
<td>0.0</td>
<td>509</td>
<td>557</td>
<td>547</td>
<td>578</td>
<td>568</td>
<td>536</td>
<td>584</td>
<td>594</td>
<td>536</td>
<td>358</td>
<td>594</td>
<td>578</td>
</tr>
<tr>
<td>n6-g25</td>
<td>12</td>
<td>1083.1</td>
<td>0.0</td>
<td>1044</td>
<td>1122</td>
<td>1075</td>
<td>1083</td>
<td>1041</td>
<td>1120</td>
<td>1149</td>
<td>1150</td>
<td>1141</td>
<td>1041</td>
<td>963</td>
<td>1120</td>
</tr>
<tr>
<td>n6-g55</td>
<td>16</td>
<td>1153.8</td>
<td>0.0</td>
<td>1102</td>
<td>1188</td>
<td>1132</td>
<td>1150</td>
<td>1141</td>
<td>1041</td>
<td>963</td>
<td>1240</td>
<td>1141</td>
<td>1041</td>
<td>963</td>
<td>1240</td>
</tr>
<tr>
<td>n6-g56</td>
<td>20</td>
<td>2397.6</td>
<td>0.1</td>
<td>2299</td>
<td>2451</td>
<td>2326</td>
<td>2396</td>
<td>2152</td>
<td>2508</td>
<td>2569</td>
<td>2570</td>
<td>2152</td>
<td>2182</td>
<td>2507</td>
<td>2570</td>
</tr>
<tr>
<td>n6-g57</td>
<td>25</td>
<td>3475.0</td>
<td>0.6</td>
<td>3331</td>
<td>3535</td>
<td>-</td>
<td>3475</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>n6-g58</td>
<td>30</td>
<td>5687.4</td>
<td>1.8</td>
<td>5460</td>
<td>5803</td>
<td>-</td>
<td>5687</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s5-42</td>
<td>12</td>
<td>14592.9</td>
<td>14.5</td>
<td>13593</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s5-49</td>
<td>15</td>
<td>21145.6</td>
<td>34.2</td>
<td>21649</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s5-66</td>
<td>16</td>
<td>41082.7</td>
<td>36.0</td>
<td>37131</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s5-72</td>
<td>20</td>
<td>6270.6</td>
<td>306.4</td>
<td>45013</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s5-82</td>
<td>25</td>
<td>8194.7</td>
<td>651.2</td>
<td>61764</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s5-12</td>
<td>30</td>
<td>10343.3</td>
<td>0.0</td>
<td>28130</td>
<td>20324</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s6-48</td>
<td>12</td>
<td>38013.0</td>
<td>0.1</td>
<td>98181</td>
<td>94968</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s6-57</td>
<td>15</td>
<td>6270.6</td>
<td>306.4</td>
<td>45013</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s6-70</td>
<td>16</td>
<td>8194.7</td>
<td>651.2</td>
<td>61764</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s6-82</td>
<td>20</td>
<td>10343.3</td>
<td>0.0</td>
<td>28130</td>
<td>20324</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s6-12</td>
<td>25</td>
<td>12481.8</td>
<td>306.4</td>
<td>45013</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s6-15</td>
<td>30</td>
<td>15229.4</td>
<td>36.0</td>
<td>37131</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s7-40</td>
<td>12</td>
<td>20955.0</td>
<td>3.9</td>
<td>209622</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s7-53</td>
<td>15</td>
<td>2374.8</td>
<td>59.3</td>
<td>40526</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s7-81</td>
<td>15</td>
<td>70102.2</td>
<td>2259.1</td>
<td>85194</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s7-90</td>
<td>18</td>
<td>10066.8</td>
<td>10181.1</td>
<td>108196</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s8-40a</td>
<td>100</td>
<td>130062.4</td>
<td>23964.7</td>
<td>143038</td>
<td>14298</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s8-40b</td>
<td>100</td>
<td>131767.3</td>
<td>18253.6</td>
<td>144826</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s9-40a</td>
<td>100</td>
<td>126655.8</td>
<td>18105.9</td>
<td>139019</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s9-40b</td>
<td>100</td>
<td>127248.7</td>
<td>19561.2</td>
<td>140577</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s9-40c</td>
<td>100</td>
<td>127574.8</td>
<td>20541.9</td>
<td>140553</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s9-40d</td>
<td>100</td>
<td>127186.1</td>
<td>20703.1</td>
<td>140288</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 1:** Global algorithm results.
<table>
<thead>
<tr>
<th>Proj</th>
<th>n</th>
<th>V</th>
<th>CPU (sec)</th>
<th>SDFMittelmannPeng</th>
<th>SDFMittelmannPeng</th>
<th>TD</th>
<th>RLT1</th>
<th>RLT2</th>
<th>RLT3</th>
<th>RLT3Dist</th>
<th>L - and - P</th>
<th>EP</th>
<th>GLR</th>
<th>PB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td>A12</td>
<td>12</td>
<td>580.3</td>
<td>0.0</td>
<td>508</td>
<td>508</td>
<td>547</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>578</td>
<td>578</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B15</td>
<td>15</td>
<td>1083.1</td>
<td>0.0</td>
<td>1044</td>
<td>1122</td>
<td>1075</td>
<td>1083</td>
<td>1041</td>
<td>1150</td>
<td>1150</td>
<td>1150</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C16</td>
<td>16</td>
<td>1153.8</td>
<td>0.0</td>
<td>1102</td>
<td>1148</td>
<td>1132</td>
<td>1150</td>
<td>1141</td>
<td>1240</td>
<td>1240</td>
<td>1240</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D20</td>
<td>20</td>
<td>2387.6</td>
<td>0.1</td>
<td>2299</td>
<td>2451</td>
<td>2326</td>
<td>2394</td>
<td>2152</td>
<td>2508</td>
<td>2569.00</td>
<td>2570</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E25</td>
<td>25</td>
<td>3475.0</td>
<td>0.4</td>
<td>3531</td>
<td>3535</td>
<td></td>
<td>3535</td>
<td>3535</td>
<td>3535</td>
<td>3535</td>
<td>3535</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F30</td>
<td>30</td>
<td>5687.8</td>
<td>1.4</td>
<td>5590</td>
<td>5603</td>
<td></td>
<td>5772</td>
<td>5772</td>
<td>5772</td>
<td>5772</td>
<td>5772</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G42</td>
<td>42</td>
<td>15092.9</td>
<td>10.2</td>
<td>14383</td>
<td>14383</td>
<td></td>
<td>15092</td>
<td>14383</td>
<td>15092</td>
<td>15092</td>
<td>15092</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H50</td>
<td>50</td>
<td>8243.2</td>
<td>2.5</td>
<td>7168</td>
<td>6897</td>
<td></td>
<td>7168</td>
<td>6897</td>
<td>7168</td>
<td>7168</td>
<td>7168</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I60</td>
<td>60</td>
<td>130296.4</td>
<td>1.0</td>
<td>128815</td>
<td>130569</td>
<td></td>
<td>130296</td>
<td>130569</td>
<td>130296</td>
<td>130296</td>
<td>130296</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J72</td>
<td>72</td>
<td>6770.6</td>
<td>106.0</td>
<td>45513</td>
<td>3721</td>
<td></td>
<td>6770.6</td>
<td>45513</td>
<td>6770.6</td>
<td>6770.6</td>
<td>6770.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K84</td>
<td>84</td>
<td>58194.7</td>
<td>296.6</td>
<td>61764</td>
<td>62691</td>
<td></td>
<td>58194</td>
<td>61764</td>
<td>58194</td>
<td>58194</td>
<td>58194</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L96</td>
<td>96</td>
<td>29018.0</td>
<td>0.1</td>
<td>86181</td>
<td>49988</td>
<td></td>
<td>29018</td>
<td>86181</td>
<td>29018</td>
<td>29018</td>
<td>29018</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M100</td>
<td>100</td>
<td>26273.0</td>
<td>1.9</td>
<td>260622</td>
<td>246707</td>
<td></td>
<td>26273</td>
<td>260622</td>
<td>26273</td>
<td>26273</td>
<td>26273</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N120</td>
<td>120</td>
<td>26273.0</td>
<td>2.9</td>
<td>258300</td>
<td>253509</td>
<td></td>
<td>26273</td>
<td>258300</td>
<td>26273</td>
<td>26273</td>
<td>26273</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O150</td>
<td>150</td>
<td>611691.6</td>
<td>6.4</td>
<td>7537980</td>
<td>7620626</td>
<td></td>
<td>611691</td>
<td>7537980</td>
<td>611691</td>
<td>611691</td>
<td>611691</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A linear formulation with $O(n^2)$ variables for the quadratic assignment problem

**Table 3** Lower bounds: times

<table>
<thead>
<tr>
<th>End</th>
<th>1/MP</th>
<th>SDP/AffineChordal</th>
<th>SDP/GN</th>
<th>RLT</th>
<th>RLT3Dist</th>
<th>L - P IP</th>
<th>RLT2IP</th>
<th>MIP++</th>
<th>SDPMittelmanPeng</th>
<th>SDPZhaoetal TD RLT</th>
</tr>
</thead>
<tbody>
<tr>
<td>9hosts, QuadCore</td>
<td>7197</td>
<td>11854</td>
<td>11801</td>
<td>6296</td>
<td>13092</td>
<td>13302</td>
<td>30129</td>
<td>7577</td>
<td>127011</td>
<td>28819</td>
</tr>
<tr>
<td>8hosts, QuadCore</td>
<td>2143</td>
<td>3143</td>
<td>3237</td>
<td>5374</td>
<td>6492</td>
<td>5314</td>
<td>6275</td>
<td>1543</td>
<td>31082</td>
<td>6</td>
</tr>
<tr>
<td>20host, quadcore</td>
<td>7577</td>
<td>127011</td>
<td>28819</td>
<td>100</td>
<td>5718</td>
<td>5718</td>
<td>229583</td>
<td>127011</td>
<td>28819</td>
<td>7469</td>
</tr>
<tr>
<td>25hosts, quadcore</td>
<td>31082</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**7 Eigenvector cuts**

Let $E$ be an $n \times n$ real matrix of ones, and let $D = \{D_{ij}\}_{1 \leq i,j \leq n}$ define the real matrix composed of the variables $D_{ij}$, where $D_{ji} = D_{ij}$ for all $i < j$. Because $D$ represents Manhattan distances in a grid graph, the following theorem holds.

**Theorem 6** Let $M = (n - 1)E - D$. We have $M \succeq 0$. 
Proof. See [28].

Mittelman et al. [28] used this result to introduce SDP relaxations based on splitting procedures of the distance matrix \( d = \{d_{kl}\}_{1 \leq k < l \leq n} \). In this paper, we propose a simplified application of this theorem by which new linear cuts may be introduced into the model \((MIP^{++})\).

Because of the semidefinite property of \( M \), we know that if \( v \) is an eigenvector of \( M \) and \( \lambda \) is its associated eigenvalue, then

\[
v^tMv = \lambda \|v\|^2 \geq 0.
\]

This remark leads to a cutting plane algorithm: Let \( \overline{D} \) denote the distance matrix obtained by solving \((MIP^{++})\), and let \( \overline{M} = (n - 1)E - \overline{D} \). Let \( \overline{v} \) denote the eigenvector of \( \overline{M} \) corresponding to the eigenvalue \( \overline{\lambda} \). We have the following theorem.

**Theorem 7** If \( \overline{\lambda} < 0 \), then

\[
\overline{v}^t\overline{M}\overline{v} \geq 0 \quad (18)
\]

is a cut.

*Proof.* We know that \( \overline{v}^t\overline{M}\overline{v} = \overline{\lambda} \|\overline{v}\|^2 \geq 0 \). If \( \overline{\lambda} < 0 \), then the inequality above is trivially violated by the current solution and hence is a cut. 

We refer to these cuts as **eigenvector cuts**. Note that \( \overline{v}^t\overline{M}\overline{v} \geq 0 \) is a linear inequality because \( \overline{v} \) is known. The eigenvector cuts are iteratively added in the following cutting plane algorithm for which the stopping criterion is identical to the previous algorithms.

**Algorithm 3: Eigenvector Cutting Plane Algorithm**

```
while stoppingCriteria == false do
    solve \( MIP^{++} \); 
    compute all eigenvectors and eigenvalues of \( \overline{M} \); 
    introduce all eigenvector cuts corresponding to negative eigenvalues ;
end
```

8 Numerical experiments

To evaluate the quality of the procedure without excessively increasing the processing times, we have replaced the constraint generation of the valid inequalities (11) in the global Algorithm 2 with the cutting plane Algorithm 3. The new procedure is as follows:
A linear formulation with $O(n^2)$ variables for the quadratic assignment problem

Algorithm 4: Global Algorithm

```
while stoppingCriteria == false do
  triangular inequalities generation;
  eigenvector cutting plane algorithm;
  constraint generation(16);
end
```

At each iteration, the eigenvectors are computed using the Gnu Scientific Library (GSL). The computing environment and time limitations are identical to those used previously. We first performed a test series with three iterations. (Table 4).

Table 4 illustrates the improvements we observed for all instances, with a more significant impact for medium- and large-size instances. For the sko instances, $V(MIP^{++})$ may generally be ranked as the second best bound; it continually performs better than $SDP_{MittelmannPeng}$. In fact, for the sko42, tho40, src12, src20, ste36a, and tho30 instances, $V(MIP^{++})$ outperformed all other methods. Although some $V(MIP^{++})$ improvement was also observed for larger size instances, our bound remained below the values attained by $SDP_{MittelmannPeng}$ and $TD$.

To further improve $V(MIP^{++})$, we performed additional numerical experiments with seven (instead of three) iterations. Because an increase in the number of iterations results in significantly increased computing times, these tests were limited to a maximum instance size of 72. Results reported in Table 5 show that, in addition to the previous good results, we were able to obtain best results for all sko instances of up to size 72 as well as for the wil50 instance.

Overall, we were able to improve 11 bounds of the literature corresponding to the following instances: sko42, sko49, sko56, sko64, sko72, scr12, scr20, ste36a, tho30, tho40, and wil50. The best heuristic solutions for seven of these instances have no optimality proofs (indicated in bold). We hope that our results contribute to finding an exact solution for the corresponding instances. Note, however, that the computational times increase significantly with the number of iterations. Considering the current state of the method, this makes it difficult to improve the bounds obtained in Table 4 for large-size instances.

9 Conclusion

The formulation we have presented uses distance variables for general quadratic assignment problems (QAPs). The original linear formulation, which had a poor bound, was improved by adding valid inequalities. The model was applied to the particular case of QAPs on grid graphs. For these cases, we were able to show that the formulation can be strengthened with facets that consider the metric property, the grid structure, and the algebraic property of a Manhattan distance matrix. The numerical experiments we performed to evaluate the quality of the lower bound show that the model is very competitive: It allowed us to compute lower bounds that were close to the best known upper bounds in less time. We
| Problem | n | $V_{MVF}$ | CPU/sec | $\bar{ZDP}_{MVF}$ | $\bar{ZDP}_{Raul}$ | $\bar{ZDP}_{Settinc}$ | $\bar{ZDP}_{MVF}_{TMDL}$ | TD | RL1 | RL2 | RL3 | RL4 | Dist | L - F | IP | GLB | PB | UB |
|---------|---|-----------|---------|-----------------|-----------------|-----------------|-------------------|----|------|------|------|------|-------|-----|-----|-----|-----|
| Test 1  | 10 | 14989.9   | 1.1     | 109100          | 109100          | 109100          | 109100            | ----| ---- | ---- | ---- | ---- | ----- | ----| ----| ----| ----|
| Test 2  | 10 | 14989.9   | 1.1     | 109100          | 109100          | 109100          | 109100            | ----| ---- | ---- | ---- | ---- | ----- | ----| ----| ----| ----|
| Test 3  | 10 | 14989.9   | 1.1     | 109100          | 109100          | 109100          | 109100            | ----| ---- | ---- | ---- | ---- | ----- | ----| ----| ----| ----|

Table 4: Lower bounds: three locations.
A linear formulation with $O(n^2)$ variables for the quadratic assignment problem

<table>
<thead>
<tr>
<th>Problem</th>
<th>VMP++</th>
<th>CPUtime</th>
<th>SDP initialization</th>
<th>SDPD initialization</th>
<th>SDP relaxation</th>
<th>RLT</th>
<th>RLT$_2$</th>
<th>L - P</th>
<th>IP</th>
<th>GLB</th>
<th>PB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td>u.g12</td>
<td>1.5</td>
<td>545.3</td>
<td>557</td>
<td>1.1</td>
<td>527</td>
<td>538</td>
<td>157.15</td>
<td>572</td>
<td>415</td>
<td>730</td>
<td>350</td>
<td>600</td>
</tr>
<tr>
<td>u.g15</td>
<td>1.5</td>
<td>1093.3</td>
<td>0.1</td>
<td>1044</td>
<td>1122</td>
<td>1075</td>
<td>1083</td>
<td>1041</td>
<td>1143</td>
<td>963</td>
<td>1350</td>
<td>0</td>
</tr>
<tr>
<td>u.g16</td>
<td>1.5</td>
<td>1163.9</td>
<td>0.1</td>
<td>1102</td>
<td>1189</td>
<td>1132</td>
<td></td>
<td>1240</td>
<td>1240</td>
<td>1240</td>
<td>1240</td>
<td>0</td>
</tr>
<tr>
<td>u.g20</td>
<td>2.4</td>
<td>2249.9</td>
<td>0.4</td>
<td>2299</td>
<td>2451</td>
<td>2326</td>
<td>2394</td>
<td>2182</td>
<td>2506</td>
<td>2182</td>
<td>2506</td>
<td>2570</td>
</tr>
<tr>
<td>u.g25</td>
<td>2.1</td>
<td>3474.7</td>
<td>1.7</td>
<td>3331</td>
<td>3335</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.g30</td>
<td>2.4</td>
<td>5825.2</td>
<td>5.3</td>
<td>5490</td>
<td>5803</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s.k12</td>
<td>1.5</td>
<td>15120.3</td>
<td>68.8</td>
<td>14593</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s.k25</td>
<td>2.4</td>
<td>22546.2</td>
<td>205.5</td>
<td>21449</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s.k56</td>
<td>5.4</td>
<td>32937.1</td>
<td>1157.5</td>
<td>31711</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s.k64</td>
<td>5.4</td>
<td>46365.8</td>
<td>3325.3</td>
<td>45013</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s.k72</td>
<td>7.3</td>
<td>63525.7</td>
<td>12635.7</td>
<td>61764</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s.k12</td>
<td>1.5</td>
<td>30597.4</td>
<td>0.0</td>
<td>28110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s.k20</td>
<td>1.5</td>
<td>97009.6</td>
<td>0.4</td>
<td>8681</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s.k36</td>
<td>1.5</td>
<td>83332.9</td>
<td>13.1</td>
<td>7108</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s.k42</td>
<td>1.5</td>
<td>38097.0</td>
<td>68.5</td>
<td>46526</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Lower bounds: seven iterations
successfully improved numerous (11) best known lower bounds.

To obtain better bounds, even for large-size instances of up to 150, requires a methodology that allows us to obtain the same bounds as above, but in a shorter timespan. Future research should consider applying decomposition techniques, such as the Lagrangian relaxation scheme, on the coupling constraints (the constraints linking the variables \( D_{ij} \) and \( x_{ij} \)). This relaxation divides the problem into a standard linear assignment and a secondary sub-problem that depends entirely on the distance variables. According to numerical observations, this sub-problem seems to be solved more rapidly. Once such a procedure is implemented, a branch-and-bound algorithm will be possible.

Acknowledgements We thank Judith Bordt for her careful reading and useful comments.

A Proof of Theorem 4

Proof. Let us prove that the points defined in Theorem 4,

\[ D^0, D^r_s, D^{m+1}, D^{m+2}, \]

with \( 1 \leq r < s \leq n \), and \( (r, s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\} \), are \( n(n-1)/2 \) affinely independent points.

Let \( \alpha^0, \alpha^r_s, \alpha^{m+1}, \) and \( \alpha^{m+2} \) be some associated scalars satisfying

\[
\begin{align*}
\alpha^0 D^0 + \sum_{1 \leq r < s \leq n, (r, s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} \alpha^r_s D^r_s + \alpha^{m+1} D^{m+1} + \alpha^{m+2} D^{m+2} &= 0 \quad (19) \\
\alpha^0 + \sum_{1 \leq r < s \leq n, (r, s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} \alpha^r_s + \alpha^{m+1} + \alpha^{m+2} &= 0 \quad (20)
\end{align*}
\]

For any fixed values \( r_0 \) and \( s_0 \) such that \( 1 \leq r_0 < s_0 \leq n \), \( (r_0, s_0) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\} \), the equation \((19)\) gives

\[
\begin{align*}
\alpha^0 D^0_{r_0s_0} + \alpha^{r_0s_0} (D^0_{r_0s_0} + M) + \sum_{1 \leq r < s \leq n, (r, s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} \alpha^r_s D^0_{r_0s_0} + \alpha^{m+1} D^0_{r_0s_0} + \alpha^{m+2} D^0_{r_0s_0} &= 0 \\
\Rightarrow D^0_{r_0s_0}[\alpha^0 + \sum_{1 \leq r < s \leq n, (r, s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} \alpha^r_s + \alpha^{m+1} + \alpha^{m+2}] + \alpha^{r_0s_0} (D^0_{r_0s_0} + M) &= 0
\end{align*}
\]

Equation \((20)\) implies

\[
\begin{align*}
\Rightarrow D^0_{r_0s_0}[-\alpha^{r_0s_0}] + \alpha^{r_0s_0} (D^0_{r_0s_0} + M) &= 0 \\
\Rightarrow \alpha^{r_0s_0} &= 0
\end{align*}
\]

It follows that

\[
\begin{align*}
\alpha^0 D^0 + \alpha^{m+1} D^{m+1} + \alpha^{m+2} D^{m+2} &= 0 \\
\alpha^0 + \alpha^{m+1} + \alpha^{m+2} &= 0.
\end{align*}
\]
Thus
\[ a^0 D^0_{i_0 j_0} + a^m D^0_{i_0 j_0} + a^{m+1} D^0_{i_0 j_0} + a^{m+2} D^0_{i_0 j_0} + M = 0 \]
\[ a^0 D^0_{i_0 h_0} + a^m D^0_{i_0 h_0} + a^{m+1} D^0_{i_0 h_0} + a^{m+2} D^0_{i_0 h_0} = 0 \]
\[ a^0 D^0_{j_0 h_0} + a^m D^0_{j_0 h_0} + a^{m+1} D^0_{j_0 h_0} + a^{m+2} D^0_{j_0 h_0} + M = 0. \]

These equations together with equation (22) imply that \( a^0 = a^m = a^{m+2} = 0 \). Thus the points are, by definition, affinely independent.

\[ \square \]

B Proof of Theorem 5

Proof. Let us prove that the points defined in Theorem 5,
\[ D^0, D^{rs}, D^{i_0 h_0}, D^{j_0 h_0}, \]
with \( 1 \leq r < s \leq n, (r,s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\} \), are \((n-1)/2\) affinely independent points.

Let \( a^0, a^{rs}, a^{i_0 h_0}, \) and \( a^{j_0 h_0} \) with \( 1 \leq r < s \leq n \) and \( (r,s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\} \), be some scalars satisfying
\[ a^0 D^0 + \sum_{1 \leq r < s \leq n, (r,s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} a^{rs} D^{rs} + a^{i_0 j_0} D^{i_0 j_0} + a^{i_0 h_0} D^{i_0 h_0} = 0 \]  
(23)
\[ a^0 + \sum_{1 \leq r < s \leq n, (r,s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} a^{rs} + a^{i_0 j_0} + a^{i_0 h_0} = 0. \]  
(24)

For \( i_0 \) and \( j_0 \), the equation (23) gives
\[ a^0 D^0_{i_0 j_0} + \sum_{1 \leq r < s \leq n, (r,s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} a^{rs} D^{rs}_{i_0 j_0} + a^{i_0 j_0} D^{i_0 j_0} + a^{i_0 h_0} D^{i_0 h_0} = 0 \]
\[ \Rightarrow a^0 + \sum_{1 \leq r < s \leq n, (r,s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} a^{rs} + a^{i_0 j_0} + a^{i_0 h_0} = 0. \]

And (24) \( \Rightarrow a^{i_0 j_0} = 0. \)

Similarly,
\[ a^0 D^0_{i_0 h_0} + \sum_{1 \leq r < s \leq n, (r,s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} a^{rs} D^{rs}_{i_0 h_0} + a^{i_0 j_0} D^{i_0 j_0} + a^{i_0 h_0} D^{i_0 h_0} = 0 \]
\[ \Rightarrow a^0 + \sum_{1 \leq r < s \leq n, (r,s) \notin \{(i_0, j_0); (i_0, h_0); (j_0, h_0)\}} a^{rs} + a^{i_0 j_0} + a^{i_0 h_0} = 0 \]
\[ \Rightarrow a^{i_0 h_0} = 0. \]

Because \( a^{i_0 h_0} = a^{j_0 h_0} = 0 \), and because the vectors \( D^0 \) and \( D^{rs} \) are affinely independent for a sufficient value of \( M \), the scalars \( a^0 \) and \( a^{rs} \) are also null, which concludes the proof.

\[ \square \]
References

A linear formulation with $O(n^2)$ variables for the quadratic assignment problem