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

Serigne Gueye and Philippe Michelon
Université d’Avignon et des Pays du Vaucluse,
Laboratoire d’Informatique d’Avignon (LIA),
339, chemin des Meinajaries,
Agroparc BP 91228,
84911 Avignon Cedex 9, France

July 14, 2014

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 [15] 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.

1 Introduction

The quadratic assignment problem (QAP), first introduced by Koopmans and Beckmann [13] 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 QAP is NP-hard [7]; 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 [4].

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.

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 [6] and by Caprara, Letchford, and Salazar-González [5] for the linear arrangement problem, a particular case of QAP. We have extended the use of these variables to QAPs and are able to present extensive numerical results.

This paper is actually a shorter version of an article submitted to the European Journal Of Computational Optimization. As a consequence, we do not give the proofs of lemmas and theorems, and parts of the initial paper have been omitted.

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

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

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

Note that, for all fixed locations $k_0$ and $l_0$, taking $x_{i0} = 1$ and $x_{j0} = 1$ implies $D_{ij} = d_{k0l0}$. Thus, $D_{ij}$ represents the distance between entities $i$ and $j$, which depends on their respective locations.

With these variables, the QAP may be formulated as the following mixed-integer linear program:
(MIP) : 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

$\sum_{i=1}^{n} x_{ik} = 1 \quad \forall \ i = 1, \ldots, n \quad (3)$

$\sum_{k=1}^{n} x_{ik} = 1 \quad \forall \ k = 1, \ldots, n \quad (4)$

$D_{ij} \geq d_{kl}(x_{ik} + x_{jl} - 1) \quad \forall \ i, j, k, l = 1, \ldots, n, i \neq j, k \neq l \quad (5)$

$x_{ik} \in \{0, 1\} \quad \forall \ i, k = 1, \ldots, n \quad (6)$

$D_{ij} \geq 0 \quad \forall \ i, j = 1, \ldots, n, i \neq j \quad (7)$

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, ($O(n^4)$) 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 Valid Inequalities

We can now introduce the first valid inequalities of conv($P$).

**Theorem 1.** The following equalities are valid inequalities for the above formulation:

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

where $\lambda_{kk'} = \min_{1 \leq k' \neq l \leq n} d_{kk'} - d_{kl}$.

**Theorem 2.** Let $d_k = \sum_{l=1}^{n} d_{kl}, \forall \ k = 1, 2, \ldots, n$ are valid equalities.

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
4 The Manhattan distance matrix

We now assume that \( d \) represents Manhattan distances of a rectangular grid graph.

**Theorem 3.** Let \( i, j, h \) satisfy \( 1 \leq i < j < h \leq n \). The following triangular inequalities are facets of:

\[
D_{ij} \leq D_{ih} + D_{jh}, \\
D_{ih} \leq D_{ij} + D_{jh}, \\
D_{jh} \leq D_{ij} + D_{ih}.
\]

**Theorem 4.** Let \( i, j, h \) satisfying \( 1 \leq i < j < h \leq m \). The following inequalities are facets:

\[
D_{ij} + D_{ih} + D_{jh} \geq 4.
\]

5 Numerical experiments on \((MIP^{++})\)

Our aim is to evaluate the quality of the lower bound corresponding to the linear relaxation of \((MIP^{++})\). We compare our results with the currently published, best known lower bounds obtained with QAPLIB instances [4] for which the distance matrix is given by the shortest path in a grid graph.

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 [20] and the Wilhelm & Ward [22] instances, and only one Thonemann & Bolte [21] instance was solved.

Results are reported in Table 1, where \( \text{Prob} \) 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 \( CPU(\text{sec.}) \). 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 ($SDP_{MittelmanPeng}$) [15], Rendl & Sotirov ($SDP_{RendlSotirov}$) [17], and Zhao et al. ($SDP_{Zhaoetal}$) [23],
- the triangular decomposition method ($TD$) [12],
- level-1, level-2, and level-3 reformulation linearization technique bounds (resp. $RLT_1$ [2], $RLT_2$ [1], and $RLT_3$ [11]),
- a level-3 RLT, performed by parallelization in a distributed environment using up to 100 host machines ($RLT_3Dist$) [9],
- the lift and project approach ($L - P$) by Burer & Vandenbussche [3],
- the interior point method ($IP$) by Resende, Ramakrishnan, and Drezner [18],
- the Gilmore & Lawler bound ($GLB$) [8] [14],
- and the projection method bound ($PB$) [10].
References


<table>
<thead>
<tr>
<th>Rank</th>
<th>m</th>
<th>VOPF (S)</th>
<th>CPU (sec)</th>
<th>SHP4(A)</th>
<th>SHP4(B)</th>
<th>SDF</th>
<th>RUT</th>
<th>RELT</th>
<th>RELT Dist</th>
<th>L (=end - P)</th>
<th>IP</th>
<th>GLB</th>
<th>FB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td>nug1</td>
<td>19</td>
<td>348.3</td>
<td>9.0</td>
<td>509</td>
<td>547</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nug15</td>
<td>15</td>
<td>1083.3</td>
<td>0.0</td>
<td>1044</td>
<td>1122</td>
<td>1075</td>
<td>1083</td>
<td>1041</td>
<td>1150</td>
<td>1149.74</td>
<td>1150</td>
<td>1141</td>
<td>1041</td>
<td>963</td>
</tr>
<tr>
<td>nug16A</td>
<td>16</td>
<td>1133.4</td>
<td>0.0</td>
<td>1102</td>
<td>1188</td>
<td>1132</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nug20</td>
<td>20</td>
<td>2387.6</td>
<td>0.1</td>
<td>2209</td>
<td>2451</td>
<td>2326</td>
<td>2394</td>
<td>2152</td>
<td>2508</td>
<td>2569.05</td>
<td>2570</td>
<td>2506</td>
<td>2182</td>
<td>2057</td>
</tr>
<tr>
<td>nug25</td>
<td>25</td>
<td>3475.0</td>
<td>0.4</td>
<td>3331</td>
<td>3535</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nug30</td>
<td>30</td>
<td>5674.4</td>
<td>1.4</td>
<td>5400</td>
<td>5803</td>
<td></td>
<td>5772</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−43</td>
<td>43</td>
<td>15929.2</td>
<td>10.2</td>
<td>14393</td>
<td></td>
<td></td>
<td>14938</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−49</td>
<td>49</td>
<td>21146.5</td>
<td>19.4</td>
<td>21149</td>
<td></td>
<td></td>
<td>22004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−56</td>
<td>56</td>
<td>30862.7</td>
<td>46.5</td>
<td>31711</td>
<td></td>
<td></td>
<td>32810</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−64</td>
<td>64</td>
<td>42770.6</td>
<td>106.0</td>
<td>45013</td>
<td></td>
<td></td>
<td>45750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−72</td>
<td>72</td>
<td>58504.7</td>
<td>290.6</td>
<td>61764</td>
<td></td>
<td></td>
<td>62001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−12</td>
<td>12</td>
<td>25341.3</td>
<td>0.0</td>
<td>25341</td>
<td>25341</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−20</td>
<td>20</td>
<td>96018.0</td>
<td>0.1</td>
<td>86181</td>
<td>94998</td>
<td></td>
<td>87998</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−30</td>
<td>30</td>
<td>15095.2</td>
<td>1.1</td>
<td>15215</td>
<td>15669</td>
<td></td>
<td>15057</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−40</td>
<td>40</td>
<td>205950.0</td>
<td>4.2</td>
<td>206622</td>
<td></td>
<td></td>
<td>214218</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wil−50</td>
<td>50</td>
<td>11784.1</td>
<td>466.5</td>
<td>18026</td>
<td></td>
<td></td>
<td>24018</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−83</td>
<td>83</td>
<td>79282.3</td>
<td>180.2</td>
<td>85391</td>
<td></td>
<td></td>
<td>80072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−90</td>
<td>90</td>
<td>100068.6</td>
<td>3507.2</td>
<td>108496</td>
<td></td>
<td></td>
<td>108493</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−100</td>
<td>100</td>
<td>139562.4</td>
<td>2488.6</td>
<td>145068</td>
<td></td>
<td></td>
<td>142608</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−100A</td>
<td>100</td>
<td>131767.3</td>
<td>3052.6</td>
<td>144826</td>
<td></td>
<td></td>
<td>143872</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−100B</td>
<td>100</td>
<td>126655.6</td>
<td>3010.7</td>
<td>139059</td>
<td></td>
<td></td>
<td>139402</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−100C</td>
<td>100</td>
<td>127248.7</td>
<td>2354.2</td>
<td>140577</td>
<td></td>
<td></td>
<td>139808</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−100D</td>
<td>100</td>
<td>127774.9</td>
<td>2939.0</td>
<td>140053</td>
<td></td>
<td></td>
<td>140305</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sk−100E</td>
<td>100</td>
<td>127786.1</td>
<td>26.7</td>
<td>140288</td>
<td></td>
<td></td>
<td>139452</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wil−100</td>
<td>100</td>
<td>142973.6</td>
<td>1474.5</td>
<td>256106</td>
<td></td>
<td></td>
<td>263909</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tho−150</td>
<td>150</td>
<td>6166301.4</td>
<td>5049.4</td>
<td>537080</td>
<td></td>
<td></td>
<td>769928</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Global algorithm results


