A Mixed Integer Quadratic Reformulation of the Quadratic Assignment Problem with Rank-1 Matrix

Otto Nissfolk\textsuperscript{a} *, Ray Pörn\textsuperscript{a} Tapio Westerlund\textsuperscript{a} Fredrik Jansson\textsuperscript{b}

\textsuperscript{a} Center of Excellence in Optimization and Systems Engineering, Åbo Akademi University, Biskopsgatan 8, 20500 Åbo, Finland
\textsuperscript{b} Department of Physics and Material Sciences Center, Philipps-Universität Marburg, Renthof 6, D-35032 Marburg, Germany

Abstract

This paper focuses on the formulation and solution of certain quadratic assignment problem (QAP). A new mixed integer quadratic programming (MIQP) formulation of the QAP is presented that is especially well suited for solving instances where the flow or distance matrix is of rank-1. Computational experiments are conducted on some special generated instances as well as on some instances from the QAPLIB (Burkard et al., 1997; QAPLIB, 2012). The QAP is solved using a two-stage procedure. The objective is first simplified as a result of the rank-1 assumption and thereafter the quadratic objective is convexified. The resulting convex MIQP is then solved with a suitable solver.

Keywords: Quadratic assignment problem (QAP), mixed integer quadratic programming (MIQP), semidefinite programming (SDP), quadratic convex reformulation (QCR)

1. Introduction

This paper addresses the important task of solving certain classes of the Quadratic Assignment Problem introduced by Koopmans and Beckmann in 1957. The QAP is a problem where $n$ facilities and $n$ locations are given with specified flows and distances between the facilities and locations, respectively. The cost is a function of the distances and flows between the facilities and an additional cost may be associated with placing a facility at a certain location. The overall objective is to minimize the total cost of placing each facility to a certain location. In addition to facility layout problems, the QAP appears in applications such as backboard wiring (Steinberg, 1961), scheduling (Geoffrion and Graves, 1976), gray pattern generation (Taillard, 1995) and many other.

In its basic form QAP is a non-convex 0-1 quadratic program. A common approach for solving QAPs is based on using different types of linearizations. A linearization procedure overcomes the quadratic structure by introducing a set of new variables and additional linear and binary constraints. Linearization techniques can also be used to obtain bounds for QAP problems. Different heuristics have also proved to be efficient to obtain bounds, but optimality of such solutions cannot be proven (Burkard et al., 1998).

The paper is organized as follows. The different quadratic formulations are introduced in section 2. The set of testproblems and well-known linearizations are presented in section 3. Section 4 contains the solution results and section 5 concludes the paper.

*Otto.Nissfolk@abo.fi
2. Problem formulation

2.1. The Quadratic Assignment Problem
The quadratic assignment problem introduced by Koopmans and Beckmann (1957) has the following form:

\[
\min_{x \in X} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ik}d_{jl}x_{ij}x_{kl} + \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij}
\]  

(1)

where \(X = \{x \mid \sum_{j=1}^{n} x_{ij} = 1 \quad i \in N, \sum_{i=1}^{n} x_{ij} = 1 \quad j \in N, x_{ij} \in \{0, 1\} \quad i, j \in N\} \) and \(f_{ik}\) is the flow between facilities \(i\) and \(k\), \(d_{jl}\) is the distance between locations \(j\) and \(l\), and \(c_{ij}\) is the cost of placing facility \(i\) at location \(j\). The variable \(x_{ij} = 1\) if facility \(i\) is assigned to location \(j\), otherwise, \(x_{ij} = 0\) and \(N = \{1, 2, \ldots, n\}\). With no loss of generality we can assume that \(c_{ij} = 0\) and omit the linear term in (1). In this paper we also assume that the flow and distance matrices are symmetric.

2.2. Trace formulation
Another popular formulation of the QAP is the trace formulation (Edwards, 1980). If \(F\) and \(D\) are given flow and distance matrices and \(X\) the permutation matrix with elements defined by (2) the quadratic objective in (1) (with \(c_{ij} = 0\)) can be expressed using the trace-operator according to

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ik}d_{jl}x_{ij}x_{kl} = \text{tr}(DXFX^T).
\]

2.3. QAP with rank-1 flow matrix
We assume that the flow matrix (or distance matrix) is of rank-1, i.e. \(F = qq^T\) for some vector \(q = (q_1, \ldots, q_n)^T\). The quadratic part of the objective function (1) can, in this case, be restated as

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ik}d_{jl}x_{ij}x_{kl} = \text{tr}(DXFX^T) = \text{tr}(DXqq^TX^T) = \text{tr}(Dyy^T) = y^TDy
\]

where \(y = Xq\), i.e. \(y\) is a permutation of \(q\). By substituting, \(y_i = \sum_{j=1}^{n} q_jx_{ij}\) we then get a quadratic problem of the form

\[
\min_{x \in X, y \in \mathbb{R}^n} y^TDy
\]

subject to \(y_i = \sum_{j=1}^{n} x_{ij}q_j \quad \forall i, \quad \sum_{i=1}^{n} y_i = \sum_{j=1}^{n} q_j\)  

(4)

Problem (3-4) is a mixed integer quadratic optimization problem with \(n\) continuous, \(n^2\) binary variables of SOS1-type and \(n + 1\) linear constraints. The objective function is not necessarily convex.
2.4. Convex QAP with rank-1 flow matrix

In order to efficiently solve the quadratic formulation defined in section 2.3 we have to convexify the objective (3). The convexification can be done, for example, by adding the smallest eigenvalue to the diagonal so that $\text{Diag}(u) = -\lambda_{\min}(D)I$ or by using the QCR-method (Billionnet et al., 2009). By solving an SDP problem we will get an optimal $u$-vector so that the lower bounding is as tight as possible. If we add $u$ to the diagonal of $D$ then we have to subtract new variables $z_v (z_v = y^2_v)$ to obtain the same objective value as in (3).

$$\min_{x \in X, y, z \in \mathbb{R}^n} y^T (D + \text{Diag}(u))y - u^T z$$ \tag{5}

subject to $y_i = \sum_{j=1}^{n} x_{ij}q_j$, $z_i = \sum_{j=1}^{n} x_{ij}q_j^2$ $\forall i$, $\sum_{i=1}^{n} y_i = \sum_{j=1}^{n} q_j$ \tag{6}

Problem (5-6) is a convex MIQP with $2n$ continuous, $n^2$ binary variables and $4n + 1$ constraints (counting SOS1-constraints). This formulation is referred to as QAP-r1. If the vector $q$ contains many identical elements, an alternative formulation can be derived. Let $\{v_1, \ldots, v_m\} (m \leq n)$ be the distinct values in $q$ and $\{f_1, \ldots, f_m\}$ the corresponding frequencies. This observation leads to a slightly different formulation.

$$\min_{y, z \in \mathbb{R}^n} y^T (D + \text{Diag}(u))y - u^T z$$ \tag{7}

s.t. $y_i = \sum_{j=1}^{m} x_{ij}v_j$, $z_i = \sum_{j=1}^{m} x_{ij}v_j^2$ $\forall i$, $\sum_{i=1}^{n} y_i = \sum_{j=1}^{m} f_j v_j$, $f_j = \sum_{i=1}^{n} x_{ij} \forall j$, $\sum_{j=1}^{m} x_{ij} = 1$ $\forall i$ \tag{8}

Problem (7-8), referred to as QAP-r1-freq, is also a convex MIQP but with $2n$ continuous, $nm$ binary variables and $3n + m + 1$ constraints. Formulation (7-8) does, however, not give the optimal permutation of $q$ only the optimal objective value. If $m < n$ the formulation QAP-r1-freq is considerably smaller than formulation QAP-r1.

3. Testproblems

The problems solved in this paper are gray-scale pattern instances (Taillard, 1995) and rank-1 approximations of all symmetric problems from the QAPLIB (2012). The problems are solved using CPLEX 12.2.0.0 on a Intel i7-930 processor with 6GB RAM running Windows 7.

3.1. Linearizations

The linearizations XYL and GLL (Zhang et al., 2012) are used as comparison to the quadratic formulations of rank-1 QAP. Both XYL and GLL have $n^2$ continuous variables, $n^2$ binary variables and $2n^2$ constraints.

3.2. Taixxc-problems

The taixxc problems found in the QAPLIB are of size $8 \times 8$ and $16 \times 16$. These problems are too large for testing so we created some smaller instances using the formula found
The tai36c problem that is used in 4.2 is a grayscale problem of size $6 \times 6$. In these instances, the flow and distance matrices are defined as follows:

$$T_{rstu} = \max_{v,w \in \{-1,0,1\}} \frac{1}{(r-t+nv)^2 + (s-u+mw)^2}$$

$$f_{ij} = \begin{cases} 1 & \text{if } i \leq m \text{ and } j \leq m \\ 0 & \text{otherwise} \end{cases}, \quad d_{ij} = d_{n(r-1)+s}, n(t-1)+u = T_{rstu}$$

4. Results

4.1. QAPLIB-results

First all symmetric problems, 111 out of 135 in total, from QAPLIB (2012) are approximated as rank-1 problems and then solved for 1800 seconds. The solutions times for the four different formulations are the compared using performance profiles showing the solution time versus number of problems solved. As one can see from figure 1 the QAP-r1-freq formulation seems to be the best, solving about 90% of all problems followed by QAP-r1 formulation solving a bit over 80% of all problems. As one could expect, the linearizations are quite similar and not as good as the quadratic ones, both solving around 20% of all problems. The linearizations work on small problems but the quadratic formulations are the best approach for these problems.

4.2. Taixxc-results

In figure 2 the results for the tai36c problem are presented. The $b$-values on the x-axis correspond to different densities of gray where $b = 0$ is all white and $b = 36$ is all black. The timelimit for the solver is set at 1800 seconds. The solution time of instances with gray
density close to 50% is very low, since the gray pattern will resemble a checkerboard and therefore there are very few good solutions. Instances with gray density close to 0% and 100% are also easy since they correspond to almost all white and all black, respectively. The intermediate instances are difficult since there exists many near-optimal patterns. The QAP-r1-freq formulation is the better one for solving these problems.

5. Conclusions
In this paper two MIQP formulations of rank-1 QAPs were derived. The results show that the formulations are efficient for solving QAPs with a rank-1 flow matrix. The QAP-r1-freq formulation seems to be the most efficient approach. This is due to a model with fewer binary variables than QAP-r1. The drawback with QAP-r1-freq is that the solution only contains the objective value but no information about the permutation vector.

References