Computers & Operations Research 37 (2010) 2007–2016

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Effective formulation reductions for the quadratic assignment problem

Huizhen Zhang [a], Cesar Beltran-Royo [a,*], Miguel Constantino [b]

[a] Statistics and Operations Research, Rey Juan Carlos University, Madrid, Spain
[b] DEIO/Centro de Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, Lisbon, Portugal

A R T I C L E I N F O

Available online 6 February 2010

Keywords:
Quadratic assignment problem
Linear integer programming
Linear programming relaxation
Sparse matrix

A B S T R A C T

In this paper we study two formulation reductions for the quadratic assignment problem (QAP). In particular we apply these reductions to the well known Adams and Johnson [2] integer linear programming formulation of the QAP. We analyze two cases: In the first case, we study the effect of *constraint reduction.* In the second case, we study the effect of *variable reduction* in the case of a sparse cost matrix. Computational experiments with a set of 30 QAPLIB instances, which range from 12 to 32 locations, are presented. The proposed reductions turned out to be very effective.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

The quadratic assignment problem (QAP) was first proposed by Koopmans and Beckmann in 1957 as a mathematical model related to the location of a set of indivisible economical activities [23]. Consider the problem of assigning n facilities to n locations in such a way that each facility is designated to exactly one location and vice-versa. The objective is to minimize the quadratic interaction cost, a function of the distances and flows between the facilities, plus the costs associated with allocating a facility to a certain location. Therefore, given three $n \times n$ matrices with real elements $F=(f_{ik})$, $D=(d_{jl})$ and $C=(c_{ij})$, where f_{ik} is the flow between the facility i and facility k, d_{jl} is the distance between the location j and l, and c_{ij} is the cost of allocating facility i at location j, the QAP can be stated as follows:

$$\min_{x \in X} \sum_{i,j,k,l=1}^{n} q_{ijkl} x_{ij} x_{kl} + \sum_{i,j=1}^{n} c_{ij} x_{ij} \tag{1.1}$$

where $q_{ijkl} = f_{ik} d_{jl}$,

$$x_{ij} = \begin{cases} 1 & \text{if facility } i \text{ is assigned to location } j \\ 0 & \text{otherwise} \end{cases}$$

and X is the set of permutation matrices of dimension n.

This set of permutations can be defined as

$$X = \left\{ x \middle| \sum_{j=1}^{n} x_{ij} = 1, \quad i \in N \right. \tag{1.2}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j \in N \tag{1.3}$$

$$\left. x_{ij} \in \{0,1\}, \quad i,j \in N \right\} \tag{1.4}$$

where $N = \{1, \ldots, n\}$.

Lawler [24] considered a more general QAP, where the q_{ijkl} coefficients in (1.1) are not restricted to flow-distance products, in contrast with the original Koopman–Beckmann formulation.

The QAP has drawn researcher's attention worldwide and extensive research has been done for more than half century. The QAP problem is considered one of the most difficult combinatorial problems: it is *NP*-hard, and even finding an ε-approximate solution is a hard problem [39]. It is surprising the number of fields where the QAP problem can be applied. In addition to its application in facility location, the QAP has been applied in many fields such as printed circuit board assembly process [14], typewriter keyboards and control panels design [36], scheduling [18], numerical analysis [7], and many others. Moreover, many well-known classical combinatorial optimization problems such as the traveling salesman problem, the graph partitioning problem, the maximum clique problem, can also be formulated as special cases of the QAP, see [35] for details.

The advances in theoretical aspects, solution techniques and applications of the QAP have been discussed in more detail, for example, in [9,5,10,27]. Regarding recent QAP advances, it is worth it to mention that during the last years some of the most challenging QAP instances have been solved by combining parallel branch-and-bound algorithms [28,12] with grid computing [4].

Calculation of lower bounds is an essential component of exact QAP methods, which employ implicit enumeration in a branch and bound framework, see [15,21,37]. On the other hand, lower bounds are used to evaluate the quality of solutions produced by heuristic algorithms, like simulated annealing algorithms [11,30], genetic algorithms (GA) [3,13], greedy randomized adaptive search procedure (GRASP) [26], ant colony algorithms [17], and so on.

* Corresponding author.
E-mail addresses: zhzzywz@gmail.com (H. Zhang), cesar.beltran@urjc.es (C. Beltran-Royo), miguel.constantino@fc.ul.pt (M. Constantino).

0305-0548/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2010.02.001

Different QAP bounds have been proposed: Gilmore–Lawler bound, eigenvalue bounds, quadratic programming bounds, LP bounds, polyhedral bounds, semidefinite bounds, among others. Very tight bounds are obtained with polyhedral methods, using cutting plane algorithms [31,40], although a large amount of computer time is needed to achieve those bounds. More details about QAP lower bounds can be found in [41,1,25,6,20,42,32,33].

In this paper we have two main objectives. The first objective is to study the effect of constraint reduction in (linear) integer programming QAP formulations (IPQAP). In the second objective we study the effect of variable and constraint reduction in the case of some null flows (sparse flow matrix).

With the first objective in mind, we present a new LP bound for the QAP, which is more effective than previous LP bounds. It is known that LP and dual-LP bounds are tight for the QAP, but appear to be computationally prohibitive in many cases [5]. To develop the new LP bound, our starting point is the IPQAP formulation of Adams and Johnson [2], that we name *IPQAP-I*. Then, by virtually dividing by two the number of its constraints, we propose an equivalent formulation, that we name *IPQAP-II*. This new formulation is less tight than *IPQAP-I*, but its LP relaxation can be solved much faster. The final result, is that formulation *IPQAP-II* usually requires a B&B tree with more nodes but more efficient in terms of total solving time.

Regarding the second objective, we have observed that quite a lot of QAP instances have a sparse flow matrix. This fact had already been observed by Padberg and Rijal [34] and also by Kaibel [40], who refers some computational experiments by Elf [29] showing that running times of a cutting plane algorithm for the QAP are reduced substantially by exploiting sparsity of the objective function. Here we study how to exploit zero flows in a QAP instance which is to be solved by a commercial IP software.

The key point is that in presence of one single zero flow, say $f_{i_0k_0}$, many coefficient costs $q_{i_0jk_0l}$ become also zero. We will show how the associated variables $y_{i_0jk_0l}$ can be eliminated in the QAP formulation. We name *IPQAP-III* and *IPQAP-IV* this reduced variable version of formulations *IPQAP-II* and *IPQAP-I*, respectively.

In our numerical experiments, we have used a set of 30 QAPLIB instances [8], that range from 12 to 32 locations, all of them with a sparse flow matrix. The results obtained by the new formulations have been surprisingly remarkable, especially, if we take into account that we have conducted our tests with a standard laptop, CPLEX 9.0 with default parameters and 4 h of CPU time limit. Within these conditions and by using the IPQAP formulations I, II, III and IV, we have solved up to optimality 2, 8, 15 and 17 QAPLIB instances, respectively.

This paper is organized as follows. In Section 2, we review the Adams and Johnson QAP linearization. In Section 3 we study the constraint reduction. In Sections 4 and 5, we study some variable reductions, especially in the case of some null flows (sparse flow matrix). The numerical experiments are presented in Section 6. Concluding remarks are made in the last section.

2. Adams and Johnson linearization

Adams and Johnson [2] linearization is the well-known linear integer programming QAP formulation:

$$\min_{x,y} \quad \sum_{i,j,k,l=1}^{n} q_{ijkl}\, y_{ijkl} + \sum_{i,j=1}^{n} c_{ij}x_{ij} \tag{2.1}$$

$$\text{s.t.} \quad \sum_{l=1}^{n} y_{ijkl} = x_{ij}, \quad i,j,k \in N \tag{2.2}$$

$$\sum_{k=1}^{n} y_{ijkl} = x_{ij}, \quad i,j,l \in N \tag{2.3}$$

$$y_{ijkl} = y_{klij}, \quad i,j,k,l \in N \tag{2.4}$$

$$y_{ijkl} \in \{0,1\}, \quad i,j,k,l \in N \tag{2.5}$$

$$x \in X \tag{2.6}$$

This formulation, that we name *IPQAP-I*, contains $o(n^4)$ variables and $o(n^4)$ constraints. Although, it produces tight LP bounds, usually it poses an obstacle for efficiently solving QAP instances from medium to large scale. Even to solve the associated LP relaxation can be difficult [38]. Other QAP linearizations can be found in literature: Lawler's linearization [19] as the first one, Kaufmann and Broeckx's linearization [22] has the smallest number of variables and constraints and Frieze and Yadegar's linearization [16], among others.

3. Formulation reduction by constraint elimination

In this section we introduce the new formulation *IPQAP-II*, which corresponds to formulation *IPQAP-I* without constraints (2.3), and with half of the constraints (2.2) relaxed into the $\leq$ form, that is

$$\min_{x,y} \quad \sum_{i,j,k,l=1}^{n} q_{ijkl}\, y_{ijkl} + \sum_{i,j=1}^{n} c_{ij}x_{ij} \tag{3.1}$$

$$\text{s.t.} \quad \sum_{l=1}^{n} y_{ijkl} = x_{ij}, \quad i,j,k \in N, \quad i \leq k \tag{3.2}$$

$$\sum_{l=1}^{n} y_{ijkl} \leq x_{ij}, \quad i,j,k \in N, \quad i > k \tag{3.3}$$

$$y_{ijkl} = y_{klij}, \quad i,j,k,l \in N \tag{3.4}$$

$$y_{ijkl} \in \{0,1\}, \quad i,j,k,l \in N \tag{3.5}$$

$$x \in X \tag{3.6}$$

Proposition 3.1. *IPQAP-II is a (valid) formulation for the QAP.*

Proof. We consider the following equivalent formulation of the QAP in the (x,y) space (we name it QAP′):

$$\min_{x,y} \quad \sum_{i,j,k,l=1}^{n} q_{ijkl}\, y_{ijkl} + \sum_{i,j=1}^{n} c_{ij}x_{ij}$$
$$\text{s.t.} \quad x_{ij}x_{kl} = y_{ijkl}, \quad i,j,k, \quad l \in N \quad x \in X$$

Let us name $F_{QAP'}, F_I$ and F_{II} the feasible sets of formulations QAP′, *IPQAP-I* and *IPQAP-II*, respectively. To prove this proposition, it is enough to prove that $F_{QAP'} = F_{II}$, since QAP′ and *IPQAP-II* have the same objective function.

First, let us see that $F_{QAP'} \subset F_{II}$. We know that QAP′ and *IPQAP-I* are equivalent formulations and that *IPQAP-II* is a relaxation of *IPQAP-I*. Therefore,

$$F_{QAP} = F_I \subset F_{II}$$

Second, let us see that $F_{II} \subset F_{QAP'}$. We consider $(x,y) \in F_{II}$ and will prove that $(x,y) \in F_{QAP'}$. Since $x \in X$, it is enough to see that $y_{ijkl} = x_{ij} x_{kl}$ for $1 \leq i,j,k,l \leq n$. Without loss of generality, we assume that $i \leq k$.

Download English Version:

https://daneshyari.com/en/article/475814

Download Persian Version:

https://daneshyari.com/article/475814

Daneshyari.com