Discrete Applied Mathematics 5 (1983) 89–98
North-Holland Publishing Company

ON THE QUADRATIC ASSIGNMENT PROBLEM

A.M. FRIEZE and J. YADEGAR*

University of London, Queen Mary College, London, England

Received 31 March 1982
Revised 11 May 1982

We discuss the relationship between Gilmore–Lawler lower bounds with decomposition for the quadratic assignment problem and a lagrangean relaxation of a particular integer programming formulation

1. Introduction

Let m be a positive integer and $M=\{1,\dots,m\}$. The Quadratic Assignment Problem (QAP) can be formulated as

$$\text{minimise} \quad \sum_{i\in M}\sum_{p\in M}\sum_{j\in M}\sum_{q\in M} a_{ipjq}x_{ip}x_{jq}+\sum_{i\in M}\sum_{p\in M} b_{ip}x_{ip}, \tag{1.1}$$

$$\text{subject to} \quad \sum_{i\in M} x_{ip}=1, \qquad p\in M, \tag{1.2a}$$

$$\sum_{p\in M} x_{ip}=1, \qquad i\in M, \tag{1.2b}$$

$$x_{ip}=0 \text{ or } 1, \qquad i,p\in M. \tag{1.2c}$$

It is known to be NP-hard and indeed even moderately sized problems with say $m=30$ cannot yet be solved in a routine manner.

Surveys of applications and approaches to this problem can be found in Gilmore [10]; Lawler [14]; Nugent, Vollmann and Ruml [16]; Bazaraa and Elshafei [1]; Bazaraa and Sherali [3]; Los [15]; and Burkard and Stratmann [5].

As $x_{ip}x_{jp}=x_{ip}x_{iq}=0$ for $i\neq j$ and $p\neq q$ in a solution to (1.2) and a term $a_{ipip}x_{ip}x_{ip}=a_{ipip}x_{ip}$ can be added to the linear term one can assume that

$$a_{ipjq}=0 \quad \text{for } i=j \text{ or } p=q. \tag{1.3}$$

We will be considering transforming the a_{ipjq} to $\bar{a}_{ipjq}$ and some comments will be made on the desirability of ensuring that the $\bar{a}_{ipjq}$ satisfy (1.3).

A particular special case of this problem is the Koopmans–Beckmann QAP [13] where we have

$$a_{ipjq}=c_{ij}d_{pq} \quad \text{for } i,p,j,q\in M, \tag{1.4}$$

* This author's work was supported by the University of London Scholarship Fund.

and corresponding to (1.3) we can assume

$$c_{ii} = d_{pp} = 0 \quad \text{for } i, p \in M. \tag{1.5}$$

Several authors have proposed branch and bound algorithms for solving this problem. One of the earlier approaches was described independently by Gilmore [10] and Lawler [14]. Recently several researchers including Burkard and Stratmann [5], Edwards [6] and Roucairol [17] have proposed combining a decomposition of the coefficients c_{ij}, d_{pq} into $\bar{c}_{ij} + \lambda_i + \mu_j$, $\bar{d}_{pq} + \nu_p + \varrho_q$ in an attempt to reduce the quadratic coefficients to $\bar{c}_{ij}\bar{d}_{pq}$ and to then apply the Gilmore–Lawler method [10, 14].

The above authors propose different methods for choosing the λ, μ, ν, ϱ none of which are provably the best in the sense of giving the best possible lower bound.

The main purpose of this paper is to link this method to a lagrangean relaxation approach (see for example, Fisher [7] or Geoffrion [8]) which has the possibility of computing a stronger lower bound.

In the next section we discuss the Gilmore–Lawler bound with decomposition and in the final section we describe some integer programming formulations of the QAP together with a particular lagrangean relaxation.

2. Gilmore–Lawler bounds with decomposition

Let α, β, γ, δ be real vectors of dimension m^3. Let

$$\bar{a}_{ipjq} = a_{ipjq} - \alpha_{pjq} - \beta_{ijq} - \gamma_{ipq} - \delta_{ipj} \quad \text{for } i, p, q, j \in M. \tag{2.1}$$

Substituting (2.1) into (1.1) transforms the objective function of the QAP into

$$\sum_{i\in M}\sum_{p\in M}\sum_{j\in M}\sum_{q\in M} \bar{a}_{ipjq} x_{ip} x_{jq} + \sum_{i\in M}\sum_{p\in M} \bar{b}_{ip} x_{ip} \tag{2.2}$$

where

$$\bar{b}_{ip} = b_{ip} + \sum_{q\in M} \alpha_{qip} + \sum_{j\in M} \beta_{jip} + \sum_{q\in M} \gamma_{ipq} + \sum_{j\in M} \delta_{ipj}.$$

We have used the fact that (1.2) implies

$$\sum_{i\in M}\sum_{p\in M}\sum_{j\in M}\sum_{q\in M} \delta_{ipj} x_{ip} x_{jq} = \sum_{i\in M}\sum_{p\in M} \left(\sum_{j\in M} \delta_{ipj}\right) x_{ip} \quad \text{etc.}$$

Next for $i, p \in M$ let

$$\bar{f}_{ip} = \text{minimum} \sum_{j\in M}\sum_{q\in M} \bar{a}_{ipjq} z_{jq}, \tag{2.3}$$

$$\text{subject to} \sum_{j\in M} z_{jq} = 1, \qquad q \in M, \tag{2.4a}$$

$$\sum_{q\in M} z_{jq} = 1, \qquad j \in M, \tag{2.4b}$$

$$z_{jq} = 0 \text{ or } 1, \qquad i, q \in M, \tag{2.4c}$$

$$z_{ip} = 1. \tag{2.4d}$$

It is clear that for all $i, p \in M$

$$\bar{f}_{ip} x_{ip} \leq \left(\sum_{j \in M} \sum_{q \in M} \bar{a}_{ipjq} x_{jq} \right) x_{ip},$$

if (1.2) holds and so the expression (2.2) is bounded below by

$$\text{GLB}(\alpha, \beta, \gamma, \delta) = \text{minimum} \sum_{i \in M} \sum_{p \in M} (\bar{f}_{ip} + \bar{b}_{ip}) x_{ip}, \qquad (2.5)$$

$$\text{subject to (1.2).}$$

We show later that computing bounds when (1.4) holds by decomposing the c_{ij}, d_{pq} is a particular case of the above.

We show first however that γ and δ are redundant in (2.5).

Let $\phi : M \to M$ be the permutation corresponding to the optimal solution to (2.5), i.e. $x(i, \phi(i)) = 1$ (for notional clarity we temporarily abandon subscripting and use a more functional notation) in the optimum solution to (2.5). Similarly define $\psi(i, p, j)$ for $i, p, j \in M$ by $z\ (j, \psi(i, p, j)) = 1$ in the optimum solution to (2.3).

Thus

$$\begin{aligned} \text{GLB}(\alpha, \beta, \gamma, \delta) &= \sum_{i \in M} (\bar{f}(i, \varphi(i)) + \bar{b}(i, \varphi(i))) \\ &= \sum_{i \in M} \left(\sum_{j \in M} \bar{a}(i, \varphi(i), j, \psi(i, \varphi(i), j)) + \bar{b}(i, \varphi(i)) \right). \end{aligned}$$

In the above expression the contribution from γ is

$$\begin{aligned} &\sum_{i \in M} \left(-\sum_{j \in M} \gamma(i, \varphi(i), \psi(i, \varphi(i), j)) + \sum_{q \in M} \gamma(i, \varphi(i), q) \right) \qquad (2.6) \\ &\quad = \sum_{i \in M} 0 = 0 \end{aligned}$$

and the contribution from δ is

$$\sum_{i \in M} \left(-\sum_{j \in M} \delta(i, \varphi(i), j) + \sum_{j \in M} \delta(i, \varphi(i), j) \right) = 0. \qquad (2.7)$$

Note that (2.6) and (2.7) are identities independent of φ and ψ. Thus the value of GLB does not depend on γ, δ.

If one wishes to impose (1.3) on $\bar{a}_{ipjq}$ (as one might to save a little storage) one can amend (2.1) to

$$\begin{aligned} \bar{a}_{ipjq} &= a_{ipjq} - \alpha_{pjq} - \beta_{ijq} \quad \text{if } i \neq j \text{ and } p \neq q, \\ &= 0 \qquad\qquad\qquad\quad \text{otherwise,} \end{aligned}$$

and only consider α, β, that satisfy

$$\alpha_{pjp} = \beta_{iiq} = 0 \quad \text{for } i, p, j, q \in M.$$

We clearly have (1.3) satisfied and further

$$a_{ipjq} x_{ip} x_{jp} = (\bar{a}_{ipjq} + \alpha_{pjq} + \beta_{ijq}) x_{ip} x_{jq} \quad \text{for } i, p, j, q \in M \qquad (2.8)$$

for x satisfying (1.2). Equation (2.8) can be substituted into (1.1) and we can proceed as before.

Now let us consider the Koopmans–Beckmann QAP. Let $\lambda, \mu, \nu, \varrho \in \mathbb{R}^m$ and let $\bar{c}_{ij} = c_{ij} - \lambda_i - \mu_j$ and $\bar{d}_{pq} = d_{pq} - \nu_p - \varrho_q$ for $i, p, j, q \in M$. It is then straightforward to see that

$$c_{ij} d_{pq} = \bar{c}_{ij} \bar{d}_{pq} + \alpha_{pjq} + \beta_{ijq} + \gamma_{ipq} + \delta_{ipj} \tag{2.9}$$

where

$$\alpha_{pjq} = \mu_j d_{pq}, \tag{2.10a}$$

$$\beta_{ijq} = c_{ij} \varrho_q - \mu_j \varrho_q, \tag{2.10b}$$

$$\gamma_{ipq} = \lambda_i d_{pq} - \lambda_i \varrho_q - \lambda_i \nu_p, \tag{2.10c}$$

$$\delta_{ipj} = \nu_p c_{ij} - \mu_j \nu_p. \tag{2.10d}$$

Thus we substitute (2.9) into (1.1) and (1.3) will still be satisfied. Note that λ and ν only contribute to the redundant (as far as GLB is concerned) γ and δ. Thus λ and ν are redundant in this decomposition.

We next check that GLB is identical to the Gilmore–Lawler bound applied to $\bar{c}, \bar{d}$ in this case.

Thus consider for some i, p

$$\bar{f}_{ip} = \text{minimum} \sum_{j \in M} \sum_{q \in M} \bar{c}_{ij} \bar{d}_{pq} z_{jq}, \tag{2.11}$$

subject to (2.4).

The assignment problem in (2.11) can be restated as how should we order the vector $\bar{d}_{p1}, \bar{d}_{p2}, \ldots, \bar{d}_{pm}$ as $\bar{d}_{p[1]}, \ldots, \bar{d}_{p[m]}$ so that $\bar{c}_{i1} \bar{d}_{p[1]} + \cdots + \bar{c}_{1m} \bar{d}_{p[m]}$ is minimized subject to $[i] = p$ (from $z_{ip} = 1$).

This can of course be solved by sorting the $\bar{c}_{ij}$, $j \neq i$ into ascending order and the $\bar{d}_{pq}$, $q \neq p$ into descending order and then forming an inner product.

Edwards [6] makes some modifications to the basic idea but these can be handled by a suitable definition of $\alpha, \beta, \gamma, \delta$.

In particular imposing $\bar{c}_{ii} = \bar{d}_{pp} = 0$ regardless of $\lambda, \mu, \nu, \varrho$ is achieved by replacing (2.10) by having

$$\alpha_{pjq} = \mu_j d_{pq} \quad \text{if } p \neq q,$$
$$= 0 \qquad \text{if } p = q$$

etc. Then (2.8) holds with a_{ipjq}, $\bar{a}_{ipjq}$ replaced by $c_{ij} d_{pq}$, $\bar{c}_{ij} \bar{d}_{pq}$.

3. An integer programming formulation

In the following integer program y_{ipjq} is implicitly $x_{ip} x_{jq}$:

IP1 minimise $\sum_{i\in M}\sum_{p\in M}\sum_{j\in M}\sum_{q\in M} a_{ipjq}\, y_{ipjq} + \sum_{i\in M}\sum_{p\in M} b_{ip}x_{ip}$, (3.1)

$$\text{subject to } \sum_{i\in M} x_{ip} = 1, \qquad p\in M, \tag{3.2a}$$

$$\sum_{p\in M} x_{ip} = 1, \qquad i\in M, \tag{3.2b}$$

$$\sum_{i\in M} y_{ipjq} = x_{jq}, \qquad p, j, q\in M, \tag{3.2c}$$

$$\sum_{p\in M} y_{ipjq} = x_{jq}, \qquad i, j, q\in M, \tag{3.2d}$$

$$\sum_{j\in M} y_{ipjq} = x_{ip}, \qquad i, p, q\in M, \tag{3.2e}$$

$$\sum_{q\in M} y_{ipjq} = x_{ip}, \qquad i, p, j\in M, \tag{3.2f}$$

$$y_{ipip} = x_{ip}, \qquad i, p\in M, \tag{3.2g}$$

$$x_{ip} = 0 \text{ or } 1, \qquad i, p\in M, \tag{3.2h}$$

$$1 \ge y_{ipjq} \ge 0, \qquad i, p, j, q\in M. \tag{3.2i}$$

We next prove the equivalence of IP1 and QAP.

It is convenient for later reference to prove the equivalence of QAP and

IP2 minimise (3.1),

subject to (3.2a), (3.2b), (3.2g), (3.2h), (3.2i),

$$\sum_{i\in M}\sum_{p\in M} y_{ipjq} = m x_{jq}, \qquad j, q\in M, \tag{3.3a}$$

$$\sum_{j\in M}\sum_{q\in M} y_{ipjq} = m x_{ip}, \qquad i, p\in M. \tag{3.3b}$$

Given an $\boldsymbol{x}$ satisfying (1.2) by taking $y_{ipjq} = x_{ip}x_{jq}$ it is straightforward to show that $(\boldsymbol{x}, \boldsymbol{y})$ is a feasible solution to IP2 and further that the objective values are the same.

Conversely let $\boldsymbol{x}, \boldsymbol{y}$ be a feasible solution to IP2. We will have shown equivalence if we can show that $y_{ipjq} = x_{ip}x_{jq}$ is satisfied.

(i) $x_{ip} = 0 \Rightarrow y_{ipjq} = 0$ from (3.3b),

(ii) $x_{jq} = 0 \Rightarrow y_{ipjq} = 0$ from (3.3a).

Let φ be the permutation of M such that $x_{i\varphi(i)} = 1$ for $i\in M$. We need only show that $y_{i\varphi(i)j\varphi(j)} = 1$ for $i, j\in M$.

Now by (1) above $\sum_{p\in M} y_{ipjq} = y_{i\varphi(i)jq}$ for $i, j, q\in M$ and so by (3.3a) with $q = \varphi(j)$ we have

$$\sum_{i\in M} y_{i\varphi(i)j\varphi(j)} = m \quad \text{for } j\in M.$$

The result now follows from (3.2i).

The equivalence of IP1 and QAP is now easy. If $\boldsymbol{x}$ is a solution to (1.2), then putting $y_{ipjq} = x_{ip}x_{jq}$ gives a feasible solution to IP1. Conversely if $(\boldsymbol{x}, \boldsymbol{y})$ is a feasible solution to IP1 it is clearly a feasible solution to IP2 and hence we have $y_{ipjq} = x_{ip}x_{jq}$.

(Note that we have not used (3.2g) which is redundant as $y_{ipip}=x_{ip}^2=x_{ip}$. It is however needed for the Langrangean relaxation described below. It does of course remove the variables y_{ipip} from the problem. We can also remove y_{ipjp} for $i\neq j$ and y_{ipiq} for $p\neq q$ as these are automatically zero - see also (3.7).)

Now consider a lagrangean relaxation of IP1 with multipliers α_{pjq} for constraints (3.2c) and multipliers β_{ijq} for constraints (3.2d).

The lagrangean function $L(\alpha,\beta)$ is thus defined by $L(\alpha,\beta)=$

$$\text{minimum} \sum_{i\in M}\sum_{p\in M}\sum_{j\in M}\sum_{q\in M}\bar{a}_{ipjq}y_{ipjq}+\sum_{i\in M}\sum_{p\in M}\bar{b}_{ip}x_{ip}, \tag{3.4}$$

$$\text{subject to (3.2a), (3.2b), (3.2e), (3.2f), (3.2g), (3.2h), (3.2i)} \tag{3.5}$$

where

$$\bar{a}_{ipjq}=a_{ipjq}-\alpha_{pjq}-\beta_{ijq}, \qquad i,p,j,q\in M,$$
$$\bar{b}_{ip}=b_{ip}+\sum_{q\in M}\alpha_{qip}+\sum_{j\in M}\beta_{jip}, \qquad i,p\in M.$$

We wish to show that $L(\alpha,\beta)=\text{GLB}(\alpha,\beta)$ of (2.5). Note that we have already demonstrated that GLB is a function of α,β only. This is straightforward.

Thus suppose $\boldsymbol{x}^*$ solves (2.5). If $x_{ip}^*=0$ let $y_{ipjq}^*=0$. If $x_{ip}^*=1$ let y_{ipjq}^* be the value of z_{jq} in the solution to (2.3) with this particular i, p and so

$$\bar{f}_{ip}=\sum_{j\in M}\sum_{q\in M}\bar{a}_{ipjq}y_{ipjq}^*.$$

This $(\boldsymbol{x}^*,\boldsymbol{y}^*)$ satisfies (3.5) and the value of (3.4) will be that of (2.5) and so $L(\alpha,\beta)\leq\text{GLB}(\alpha,\beta)$. Conversely if $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})$ solves (3.4), then

$$\sum_{j\in M}\sum_{q\in M}\bar{a}_{ipjq}\hat{y}_{ipjq}\geq\bar{f}_{ip}\hat{x}_{ip}$$

and so $L(\alpha,\beta)\geq\text{GLB}(\alpha,\beta)$.

It follows then that lower bounds obtained by decomposition in conjunction with the Gilmore-Lawler method can be no larger than $L^*=\max_{\alpha,\beta}L(\alpha,\beta)$ which from Geoffrion [8] is equal to the minimum objective value in the linear relaxation of IP1, i.e. when (3.2g) is replaced by $x_{ip}\geq 0$.

4. Computational considerations

Computing L^* by solving the linear relaxation of IP1 by the simplex algorithm does not look very promising as we have $4m^3+2m$ equality constraints to deal with.

A natural approach is to use the sub-gradient algorithm - see Fisher [7] - to try and find a near optimal set of multipliers α^*,β^*. This however requires $O(m^3)$ storage space for the multipliers and requires the solution of m^2+1 assignment problems at each step.

If we have a general (non Koopmans-Beckmann) problem, then since this requires $O(m^4)$ storage for the coefficients the storage problem for the multipliers is marginal.

We still have the problem of solving m^2+1 (small) assignment problems. This may just be viable using a powerful parallel processing computer like the I.C.L. Distributed Array Processor.

The most important practical problems are of the Koopmans–Beckmann type and here the data requires $O(m^2)$ storage space. In this case it is worth considering a restricted set of multipliers i.e. identify a subset $S \subseteq \mathbb{R}^{2m^3}$ and restrict attention to $(\alpha, \beta) \in S$.

(a) The simplest approach is to restrict α, β so that they can be expressed as

$$\alpha_{pjq} = u_{pj}^{(1)} + u_{pq}^{(2)} + u_{jq}^{(3)},$$
$$\beta_{ijq} = u_{ij}^{(4)} + u_{iq}^{(5)} + u_{jq}^{(6)},$$

so that we need only store the 6 m^2-vectors $\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(6)}$. The redundancy of γ, δ implies that only $\boldsymbol{u}^{(3)}, \boldsymbol{u}^{(6)}$ are non-redundant in computing GLB. Since we would actually work with the expression $\bar{a}_{ipjq} + \boldsymbol{u}_{jq}^{(3)} + \boldsymbol{u}_{jq}^{(6)}$ we would combine $\boldsymbol{u}^{(3)} + \boldsymbol{u}^{(6)}$ into a single vector $\boldsymbol{u}$.

If we examine $L(\boldsymbol{u})$ we see that this is the same as replacing (3.2c), (3.2d) by (3.2a) and applying lagrangean relaxation (with multipliers $\boldsymbol{u}$ for constraints (3.3a)) to an integer program which is also equivalent to QAP as IP2 is.

(b) Another approach is to restrict α, β so that they can be expressed in terms of μ, ϱ as in (2.10). The advantage here is that $L(\alpha, \beta)$ is the Gilmore–Lawler bound with decomposition and the assignment problems corresponding to the $\bar{f}_{ip}$ can be solved easily by sorting. The major drawback however is that the set S is non-convex and so one cannot use the sub-gradient algorithm to optimise α, β over S.

One heuristic approach to finding a good μ, ϱ that springs to mind is the following:

Given α, β as defined by (2.10) compute $L(\alpha, \beta)$ and a sub-gradient $(\Delta\alpha, \Delta\beta)$ and choose a step length $t > 0$. Thus $(\alpha', \beta') = (\alpha + t\Delta\alpha, \beta + t\Delta\beta)$ is likely to be an improvement on α, β but probably would not be of the form (2.10).

One could then choose μ in order to minimise

$$\sum_{p\in M} \sum_{j\in M} \sum_{q\in M} (\alpha'_{pjq} - \mu_j d_{pq})^2 \tag{4.1a}$$

and having chosen μ one could choose ϱ to minimise

$$\sum_{i\in M} \sum_{j\in M} \sum_{q\in M} (\beta'_{ijq} - \varrho_q (c_{ij} - \mu_j))^2. \tag{4.1b}$$

The values of μ and ϱ that minimise (4.1) might then produce (via (2.10a), (2.10b)) a solution 'close' to α', β'.

The formulae for μ, ϱ that minimise (4.1) are

$$\mu_j = \left(\sum_{p\in M} \sum_{q\in M} \alpha'_{pjq} d_{pq} \right) \Big/ \left(\sum_{p\in M} \sum_{q\in M} d_{pq}^2 \right), \qquad j \in M,$$

$$\varrho_q = \left(\sum_{i\in M} \sum_{j\in M} \beta'_{ijq} (c_{ij} - \mu_j) \right) \Big/ \left(\sum_{i\in M} \sum_{j\in M} (c_{ij} - \mu_j)^2 \right), \quad q \in M.$$

Table 1

p	m	lb0	lb1	it1	lb2	it2	bv
1	4	806[a]	806	2	806	1	806*
2	4	179[b]	184	6	184	4	184*
3	5	50[c]	50	1	50	1	50*
4	6	82[d]	86	166	82	1	86*
5	7	137[d]	148	376	138	30	148*
6	8	186[d]	194	411	187	20	214*
7	12	493[d]	-	-	494	350	578*
8	15	963[d]	-	-	963	1	1150*
9	20	2057[d]	-	-	2057	1	2570
10	36	3196.81[d]	-	-	3196.81	1	4119.55
11	7	505[b]	559	23	511	50	559*
12	4	130[b]	132	6	132	2	132*
13	8	727[b]	811	211	733	130	891*
14	8	10043116[b]	11174262	821	10135364	220	11902372*
15	9	11298[e]	17293	347	12569	100	25388*

[a] method 4 [b] method 5 [c] methods 1, 3 [d] methods 1, 2, 3 [e] method 2

We have carried out some computational experiments to try to evaluate the strength of the proposed bounds. The results of these experiments are given in Table 1 above.

So far we have tested 2 ideas:

(i) The use of the subgradient algorithm to try to get an approximate value for $\max L(\alpha, \beta)$. The largest problem size we have tried this on is with $m=9$. We hope to tackle larger m later on using the I.C.L. Distributed Array Processor.

(ii) The idea of (b) above to use the subgradient algorithm in conjunction with (4.1).

Explanation of Table 1

P: Source of problems. Problem 1 is from Gavett and Plyter [9].

Problem 2 is from Roucairol [17].

Problems 3–9 are from Nugent, Vollmann and Ruml [16].

Problem 10 is from Steinberg [18] (Euclidean distance).

Problem 11 is from Lawler [14].

Problems 12–15 are from Burkard and Gerstl [4], being respectively B12, B39, B40, B32.

lb0. This is the maximum of 5 lower bounds using choices for μ, ϱ as suggested by:

1. Gilmore [10] ($\mu=\varrho=0$).
2. Roucairol [17] ($\mu_j=\min_{i\neq j} c_{ij}$, $\varrho_q=\min_{p\neq q} d_{pq}$).
3. Burkard and Stratmann [5] ($\mu_j=\min_{i\neq j} c_{ij}$, $\varrho_q=0$).
4. Roucairol [17] (no simple formula, method 2 of her paper).
5. Edwards [6] ($\mu_j=(\sum_i c_{ij})/(n-1)$, $\varrho_q=(\sum_p d_{pq})/(n-1)$).

These values minimise $\sum\sum_{i \neq j} (c_{ij} - \mu_j)^2$ etc. and are similar but not identical to those in [6]).

lb1. This is the largest value obtained for $L(\alpha, \beta)$ using the subgradient algorithm.

it1. The number of iterations of the subgradient algorithm needed to reach lb1. (The approach of Bazaraa and Goode [2] was most successful except on examples 3–6 where the Held, Wolfe and Crowder [11] approach seemed better, though we think we can get the former method to work as well on these.)

lb2. This is the largest lower bound obtained using the idea of (b).

it2. The number of iterations needed to reach lb2.

bv. The value of the best known solution to these problems. An * indicates that this is known to be minimal.

Table 1 shows that lb1 is a stronger bound than lb0, indeed for $m \leq 7$ above $\text{lb1} = \text{bv}$.

However computing lb1 is very time consuming and we are banking on parallel computation to make it practical for the larger problems.

lb2 is not much better than lb1 and evidently approach (b) needs to be refined.

We finish by describing an integer program equivalent to QAP which has $\frac{1}{2}(m^4 - 2m^3 + 3m^2)$ variables and $m^2 + 2m$ constraints as opposed to that given in Bazaraa and Sherali [3] which has the same number of variables but has $2m^2$ constraints.

We should point out that Kaufman and Broeckx [12] have produced an integer programming formulation with only $2m^2$ variables and $m^2 + 2m$ constraints. The following formulation however does seem to have a simpler structure than that in the above paper [12].

$$\text{minimise} \quad \sum_{i \in M} \sum_{p \in M} \sum_{j \in M} \sum_{q \in M} a_{ipjq} y_{ipjq} + \sum_{i \in M} \sum_{p \in M} b_{ip} x_{ip}, \tag{4.2a}$$

$$\text{subject to} \quad \sum_{i \in M} x_{ip} = 1, \qquad p \in M, \tag{4.2b}$$

$$\sum_{p \in M} x_{ip} = 1, \qquad i \in M, \tag{4.2c}$$

$$\sum_{i \in M} \sum_{p \in M} y_{ipjq} = m x_{jq}, \qquad j, q \in M, \tag{4.2d}$$

$$y_{ipjq} = y_{jqip}, \qquad i, p, j, q \in M, \tag{4.2e}$$

$$y_{ipip} = x_{ip}, \qquad i, p \in M, \tag{4.2f}$$

$$y_{ipjp} = y_{ipiq} = 0, \qquad i \neq j,\ p \neq q \in M, \tag{4.2g}$$

$$x_{ip} = 0 \text{ or } 1, \qquad i, p \in M, \tag{4.2h}$$

$$0 \leq y_{ipjq} \leq 1, \qquad i, p, j, q \in M. \tag{4.2i}$$

(4.1e)–(4.1g) are only implicit constraints. Their effect is simply to reduce the number of variables. Writing the program in this way simplifies the notation.

The proof of equivalence is almost identical to that of IP2 and is omitted.

References

[1] M.S. Bazaraa and A.N. Elshafei, An exact Branch-and-Bound procedure for the Quadratic Assignment Problem, Naval Res. Logist. Quart. 26(1) (1979) 109–121.

[2] M.S. Bazaraa and J.J. Goode, The Travelling Salesman Problem: a duality approach, Math. Programming 13 (1977) 221–237.

[3] M.S. Bazaraa and H.D. Sherali, Benders' partitioning scheme applied to a new formulation of the Quadratic Assignment Problem, Naval Res. Logist. Quart. 27(1) (1980) 29–41.

[4] R.E. Burkard and A. Gerstl, Quadratische Zourdnungs Probleme III: Test Beispiele und Rechenzeiten, Rechenzentrum Graz Report No. 85, August 1973.

[5] R.E. Burkard and K.H. Stratmann, Numerical investigation on the Quadratic Assignment Problem, Naval Res. Logist. Quart. 25(1) (1978) 129–148.

[6] C.S. Edwards, A Branch and Bound algorithm for the Koopmans-Beckmann Quadratic Assignment Problem, Math. Programming Study 13 (1980) 35–52.

[7] M.L. Fisher, Lagrangean relaxation methods for combinatorial optimisation, Proc. Summer School in Combinatorial Optimization, Urbino, Italy, July 1978, to appear.

[8] A.M. Geoffrion, Lagrangean relaxation and its uses in integer programming, Math. Programming Study 2 (1974) 82–114.

[9] J.W. Gavett and N.V. Plyter, The optimal assignment of facilities to locations by Branch and Bound, Operations Res. 14 (1966) 210–232.

[10] P.C. Gilmore, Optimal and suboptimal algorithms for the Quadratic Assignment Problem, SIAM J. Appl. Math. 10 (1962) 305–313.

[11] M. Held, P. Wolfe and H.P. Crowder, Validation of subgradient optimisation, Math. Programming 6 (1974) 62–88.

[12] L. Kaufman and F. Broeckx, An algorithm for the Quadratic Assignment Problem using Benders' decomposition, European J. Oper. Res. 2 (1978) 204–211.

[13] T.C. Koopmans and M. Beckmann, Assignment problems and the location of economic activities, Econometrica 25 (1957) 53–76.

[14] E.L. Lawler, The Quadratic Assignment Problem. Management Sci. 19 (1963) 586–590.

[15] M. Los, Comparison of several heuristic algorithms to solve Quadratic Assignment Problems of the Koopmans-Beckmann type, presented at the International Symposium on Location Decisions, Banff, Alberta, April 1978.

[16] C.E. Nugent, T.E. Vollmann and J. Ruml, An experimental comparison of techniques for the assignment of facilities to locations, Operations Res. 16 (1968) 150–173.

[17] C. Roucairol, A Branch and Bound method for Quadratic Assignment Problems: the reduction method, presented at the 3rd European Congress on Operations Research, Amsterdam, April 1979.

[18] L. Steinberg, The backboard wiring problem: a placement algorithm, SIAM Review 3 (1961) 37–50.