SPECIAL CASES OF THE QUADRATIC ASSIGNMENT PROBLEM

CARNEGIE-MELLON UNIVERSITY

PREPARED FOR
OFFICE OF NAVAL RESEARCH

APRIL 1976

Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER

W.P.#69-75-76

Management Sciences Research Report No. 391

SPECIAL CASES OF THE QUADRATIC ASSIGNMENT PROBLEM

Nicos Christofides* and M. Gerrard†

April 1976

Management Science Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

*Carnegie-Mellon University, on leave from Imperial College, London
†Imperial College, London

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1 REPORT NUMBER: Technical Report No. 391	2 GOVT ACCESSION NO. / 3 RECIPIENT'S CATALOG NUMBER
4 TITLE (and Subtitle): Special Cases of the Quadratic Assignment Problem	5 TYPE OF REPORT & PERIOD COVERED: Technical Report April 1976 / 6 PERFORMING ORG. REPORT NUMBER
7 AUTHOR(s): Nicos Christofides and M. Gerrard	8 CONTRACT OR GRANT NUMBER(s)
9 PERFORMING ORGANIZATION NAME AND ADDRESS: Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213	10 PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11 CONTROLLING OFFICE NAME AND ADDRESS: Personnel and Training Research Programs, Office of Naval Research (Code 434), Arlington, Virginia 22217	12 REPORT DATE: April 1976 / 13 NUMBER OF PAGES: 15
14 MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15 SECURITY CLASS. (of this report): Unclassified / 15a DECLASSIFICATION/DOWNGRADING SCHEDULE

16 DISTRIBUTION STATEMENT (of this report)

Approved for public release, distribution unlimited

17 DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18 SUPPLEMENTARY NOTES

19 KEY WORDS (Continue on reverse side if necessary and identify by block number)

The quadratic assignment problem, graph theory, dynamic programming

20 ABSTRACT (Continue on reverse side if necessary and identify by block number)

By considering the quadratic assignment problem (QAP) as that of minimizing the product of a distance-graph with a flow-graph several special cases of the QAP are investigated. A polynomial-growth algorithm is described for the QAP when the distance and flow-graphs are isomorphic trees. In the case when the graphs are single stars the algorithm becomes the well known rule for multiplying two sequences of numbers. The case of a complete distance-graph and a tree flow-graph becomes the travelling salesman problem when the tree is a hamiltonian chain and the flows are all unity. A dynamic programming

(over)

ABSTRACT

By considering the quadratic assignment problem (QAP) as that of minimizing the product of a distance-graph with a flow-graph several special cases of the QAP are investigated. A polynomial-growth algorithm is described for the QAP when the distance and flow-graphs are isomorphic trees. In the case when the graphs are single stars the algorithm becomes the well known rule for multiplying two sequences of numbers. The case of a complete distance-graph and a tree flow-graph becomes the travelling salesman problem when the tree is a hamiltonian chain and the flows are all unity. A dynamic programming algorithm is presented for the case of the flow-graph being a general tree with arbitrary flows. The very special case of "narrow" bipartite graphs is also considered.

ib

1. INTRODUCTION

Consider p machines $1,\ldots,\alpha,\ldots,p$ with a known flow of material $f_{\alpha\beta}$ between every pair of machines (α,β). Let there be $q \geq p$ locations $1,\ldots,i,\ldots,q$ with known distances d_{ij} between every pair of locations (i,j). An assignment of machines to locations is a one to one mapping ρ of the set of machines into the set of locations, so that $\rho(\alpha)$ is the location that machine α is assigned to.

The cost of a mapping ρ is defined as

$$z(\rho) = \sum_{\alpha \neq \beta} f_{\alpha\beta} d_{\rho(\alpha)\rho(\beta)} \qquad (1)$$

Given the two matrices $[f_{\alpha\beta}]$ and $[d_{ij}]$, the quadratic assignment problem (QAP) is that of finding a mapping ρ^* which minimizes $z(\rho)$ as given by (1).

The QAP appears in a number of spacial location problems such as the allocation of machines to locations - used above to introduce the QAP - the location of electronic components on circuit boards [12], the ordering of interrelated data on magnetic tape, etc. Other examples not involving special location, but which can be formulated as QAP's include the triangularization of economic input-output matrices [2], the minimization of average job completion time in machine scheduling [8] and extensions of the travelling salesman problem [7].

A survey of exact algorithms for the general QAP is given by Pierce and Crowston [11], and an improved algorithm is described in [4]. Exact algorithms, however, are unable to solve general QAP's of even moderate size [4]. Approximate algorithms for the QAP are surveyed by Nugent et al [10] and Moore [9], while Sciabin and Vergin [13] demonstrate that these are, in general, unsatisfactory.

In this paper we consider special cases of the QAP which are easier to solve. By recasting the QAP in graph theoretic terms as the dot-product of a distance-graph with a flow-graph, we consider cases when these graphs have special forms. In particular we describe a polynomial growth algorithm for the QAP when both of these graphs are trees. When only one of the two graphs is a tree and the other is a complete graph, the QAP can be solved by a special dynamic programming algorithm which is a generalization of a similar algorithm for the travelling salesman problem. This last case occurs very often in practical location problems,e.g., in the layout of an assembly line.

2. GRAPHICAL REPRESENTATION

A graph G is defined by the doublet (X,A) where X is a set of vertices and A a set of links. Unless otherwise specified we will use "graph" to mean a "non-directed graph without loops." The terminology used is from [1, 3].

Given a graph $G' = (X',A')$ with a cost matrix $[c'_{ij}]$, an isomorphic graph $G'' = (X'', A'')$ with a cost matrix $[c''_{ij}]$ and a mapping ρ of X' onto X'', the dot-product graph is written - using the product operator $\pi(\rho)$ - as:

$$G'\pi(\rho)G''$$

and is defined as the graph G = (X,A) isomorphic to G' with costs given by $c_{ij} = c'_{ij} \cdot c''_{\rho(i)\rho(j)}$.

The value of a graph G = (X,A) is defined as:

$$V(G) = \sum_{(x_i,x_j)\in A} c_{ij}$$

An image of a graph G' in a graph G'' is any partial subgraph of G'' which is isomorphic to G'. We will denote by $M(G',G'')$ the family of all such image graphs. The cardinality of the set $M(G',G'')$ is called the image number of G' in G'' and is denoted by $m(G',G'')$.

The QAP can now be restated in the following way.

Let $G^f = (X^f,A^f)$ be a flow-graph, whose vertices X^f represent the set of machines and the link costs are the flows between the corresponding machines. Similarly, let $G^d = (X^d,A^d)$ be a distance-graph, whose vertices represent the set of locations and the link costs are the distances between the corresponding locations. We will assume (without loss of generality) that $|X^f| \geq |X^d|$.

The QAP is then the problem of finding a graph G and a mapping ρ of G^f on G which minimizes the expression:

$$\min_{G \in M(G^f, G^d)} [\min_{\rho} V(G, \pi(\rho) G)] \qquad (2)$$

The number of different mappings ρ of G^f onto an isomorphic graph is the isomorphic number $s(G^f)$ of G^f. Thus, the inner minimization of (2) is over a set of cardinality $s(G^f)$ and the outer minimization is over a set of cardinality $m(G^f, G^d)$.

It is reasonable to expect that as $s(G^f)$ and $m(G^f, G^d)$ increase, the difficulty of the QAP will also increase. Cases when both $s(G^f)$ and $m(G^f, G^d)$ are small can be solved trivially by enumeration. Very few special cases in which only one of these two numbers is large can be solved by polynomial growth algorithms. The case where $G^f = G^d = K_n$ (the complete graph on n vertices) is the problem usually considered in the literature as the general QAP and has $m(G^f, G^d) = 1$ and $s(G^f) = n!$

3. CASES WITH G^f ... G^d (Image number = 1)

When the image number $m(G^f, G^d) = 1$ the outer minimization of (2) becomes redundant and only the inner minimization remains.

3.1. Trivial cases (small isomorphic number)

Trivial cases that can be solved by complete enumeration are:

(i) Chains: When G^f and G^d are chains, the isomorphic number $s(G^f) = 2$ and the inner minimization in expression (2) only involves two evaluations.

(ii) Cycles: The isomorphic number $s(G^f) = 2n$.

(iii) Wheels: The isomorphic number $s(G^f) = 2(n-1)$.

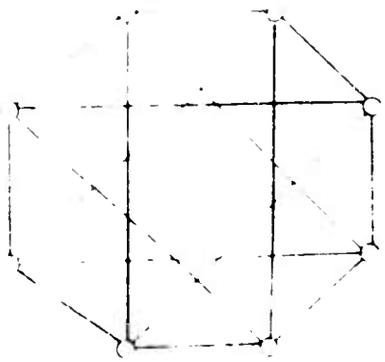
(iv) Regular graphs: Certain regular graphs (e.g., webs of low order) have small isomorphic numbers and can be enumerated. There is, however, great variation in the isomorphic numbers of regular graphs even of the same degree as shown by the example in Fig. 1., and no general statement can be made.

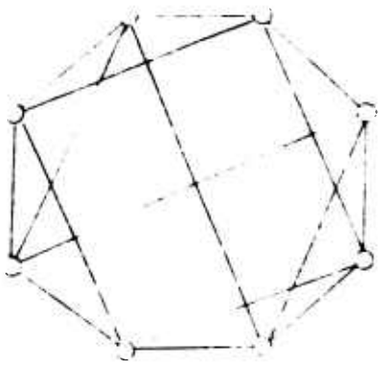
3.2. Solvable cases (large isomorphic number)

A. Simple stars: When G^f and G^d are simple stars with one central vertex (with index 0) and n outer vertices, the isomorphic number is n! However, this special QAP can be solved by a well known rule namely: Order the n flows $f_{o\alpha}$ in ascending order, and the n distances d_{oi} in descending order. The optimum mapping p^* then maps the k^{th} flow in the flow list to the k^{th} distance in the distance list for all $k = 1,\ldots,n$.

B. Multiple stars

The graph in Fig. 2 shows a 28-vertex 3rd order star with vertex 1 as the center. Consider a general k-order multiple





$s(G) = 115$

$c(G) = 4$

FIG 1 THE ISOMORPHIC NUMBERS OF TWO REGULAR GRAPHS OF DEGREE 4 ($N = 8$)

star graph arbitrarily rooted at the center vertex and let the label $\ell(x)$ of vertex x be the cardinality of the path from the center to x. The label of the outermost vertices is then k and the label of the center vertex is zero.

We describe below a dynamic programming algorithm for the solution of QAP's involving arbitrary k-order stars.

Let the flow and distance graphs be $G^f = (X^f, A^f)$ and $G^d = (X^d, A^d)$ respectively. For any vertex $x_\alpha \in X^f$ and $x_i \in X^d$ let $c(\alpha,i)$ be the minimum cost of mapping x_α and all its successors (i.e., vertices reachable from x_α via arcs of the rooted tree) to x_i and all its successors. We will denote by $x_{p(\alpha)}$ the predecessor of vertex x_α.

Description of the algorithm (for k-order stars)

Step 1. For each $x_\alpha \in X^f$ and $x_i \in X^d$ with $\ell(x_\alpha) = \ell(x_i) = k$ set $c(\alpha,i) = f_{p(\alpha)\alpha} \cdot d_{p(i)i}$.

Set LEVEL = k - 1

Step 2. For each $x_\alpha \in X^f$ and $x_i \in X^d$ with $\ell(x_\alpha) = \ell(x_i) =$ LEVEL calculate $c(\alpha,i)$ as follows:

(i) Let $\{\beta_1,\ldots,\beta_r\} = \{\beta \mid p(\beta) = \alpha\}$

and $\{j_1,\ldots,j_r\} = \{j \mid p(j) = i\}$

(ii) Set up the linear assignment problem with cost matrix

$$C = \begin{bmatrix} c(\beta_1,j_1) & \cdots & c(\beta_1,j_r) \\ \vdots & & \vdots \\ c(\beta_r,j_1) & & c(\beta_r,j_r) \end{bmatrix}$$

and let $V_{\alpha i}$ be the value of the solution of this problem.

(iii) Update $c(\alpha,i) = f_{p(\alpha)\alpha} \cdot d_{p(i)i} + V_{\alpha i}$.

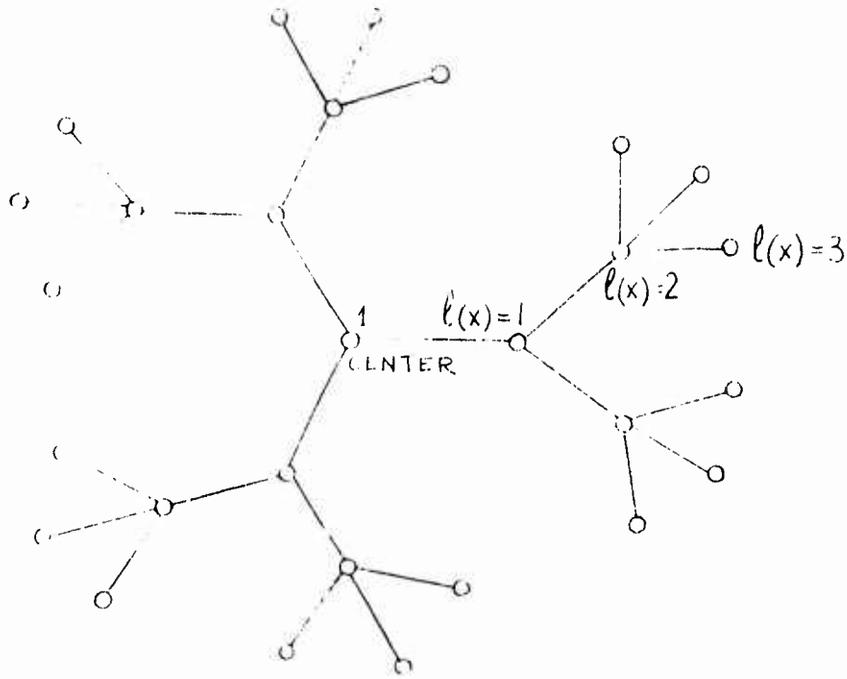


FIG 2. THIRD [illegible] TREE WITH N=28.
ISOMORPHIC NUMBER 2 2 3 9 4 8 8

Step 3. Set LEVEL = LEVEL -1. If LEVEL = 0 go to (4) else go to (2).

Step 4. Stop. If x_{α_o} and x_{i_o} are the center vertices of G^f and G^d respectively, $c(\alpha_o, i_o)$ is the value of the solution to the QAP. (Note: The mapping corresponding to this solution value can be found by backtracking in the usual dynamic programming manner.)

The above algorithm is good, exhibiting polynomial rate of growth with the total number of vertices in the k-order star. Thus, if each vertex of the star has exactly m successors, there are $(m^{k+1}-1)/(m-1)$ vertices in all. The algorithm involves $2m^{k-1}$ sortings, m^{2k-2} evaluations and the solution of $(m^{2k-2}-1)/(m^2-1)$ assignment problems of size m x m. It should be noted here that in the first pass through step 2 of the above algorithm, the solution of the assignment problems defined by C is unnecessary since these assignment problems correspond to simple stars and can be solved by the simple ordering described earlier. Thus, in the case of the simple star the above algorithm disintegrates to the well known ordering rule of 3.7(A).

C. General trees

Any tree can be arbitrarily rooted and considered as a k-order star with ∞-cost (flow/distance) arcs. The algorithm described above can therefore be used to solve QAP's with general trees. This is equivalent to slightly modifying the above algorithm so that $c(\alpha,i)$ is finite only for those pairs of vertices $[x_\alpha, x_i]$ for which the subtree defined by x_α and its successors in G^f is isomorphic to the subtree defined by x_i and its successors in G^d.

D. Narrow Bipartite Graphs

Consider the QAP when G^f and G^d are complete bipartite graphs, say of the form $K_{r,s}$. Using the normal notation we will express G^f in terms of its two independent vertex sets, i.e., $G^f = (X_r^f, X_s^f)$ and similarly $G^d = (X_r^d, X_s^d)$. We call $K_{r,s}$ narrow if min (r,s) << max (r,s). Let us assume that r = min (r,s) and specifically consider the case when r is small. The isomorphic number of $K_{r,s}$ is r!s!, however, if r is small enough, the r! possible mappings of X_r^f on X_r^d could be enumerated. For each such mapping ρ we would then compute $\forall\, x_\alpha \in X_s^f$ and $x_i \in X_s^d$:

$$c(\alpha,i) = \sum_{x_\beta \in X_r^f} f_{\alpha\beta} \cdot d_{i\rho(\beta)}$$

and solve the s by s linear assignment problem with cost matrix $[c(\alpha,i)]$. The least cost assignment solution over all r! mappings ρ is then the solution to the QAP.

Obviously, such a procedure is only practical when r is very small (say $\leq$ 5) but with a given r the complexity as a function of s is $O(s^{2.5})$ since it only involves the linear assignment problem.

4. CASES WHEN G^d IS COMPLETE

We will now take G^d to be a complete graph on n vertices and consider cases when G^f is of different forms.

A. G^f is a simple star with n vertices

The image number $m(G^f,G^d)$ is n. Each image corresponds to a star partial graph of G^d with a specific center vertex. Once an image graph G^d is chosen the optimum mapping ρ of G^f onto G^d can be found as in section 3.2(A) earlier. The total complexity of the procedure is therefore $O(n^2 \log n)$.

B. General trees with n vertices

Although the procedure for simple stars given in section 3 for the case in which $G^f = G^d$ is generalized above to the case in which G^d is a complete graph, the corresponding algorithm of section 3.2(B) for k-order stars (or arbitrary trees) does not generalize. The fact that such a generalization is not possible can be demonstrated by considering G^f to be a simple chain of n vertices numbered consecutively from an end vertex and take $f_{\alpha,\alpha+1} = 1$ for all $\alpha = 1,\ldots,n-1$. We now have $s(G^f) = 2$ and $m(G^f,G^d) = 1/2n!$ In fact, the image graphs of G^f in G^d are all the hamiltonian paths of G^d, and since we have taken all flows to be unity, the value of the product graph of G^f with an image graph is simply the length of the hamiltonian path forming the image graph. (Note that the 2 possible mappings of G^f onto the image graph give the same value.) Thus, the QAP with G^f a simple chain and G^d the complete graph becomes equivalent to the open-ended travelling salesman problem.

Although the algorithm of section 3.2(B) does not generalize to the present case, this case (of G^f being an arbitrary tree and G^d the complete graph) is possibly the most important of all cases of the QAP - as far as

practical applications are concerned - since in many situations (e.g., in assembly line layout, pipeline design, etc.) the flow graph is of this form. In view of its importance we present here a specialized algorithm which can solve QAP's of considerably larger size than any algorithm for the general QAP.

The algorithm is a generalization of the dynamic programming algorithm in [6] for the travelling salesman problem. The generalization is in two directions: (i) it considers different flows between machines, and (ii) it can accommodate arbitrary trees instead of simply chains.

Consider a general tree graph G^f and suppose it is arbitrarily rooted at some vertex x_o. Let $T(x_\alpha)$ be the directed subtree reachable from x_α, including x_α itself as the root of the subtree. We will also use $T(x_\alpha)$ to mean the set of vertices of this subtree. Let $x_{p(\alpha)}$ be the immediate predecessor of vertex x_α in the rooted tree G^f.

For a given vertex $x_\alpha \in X^f$ let $S_\alpha \subseteq X^d$ and $x_i \in S_\alpha$ where $|S_\alpha| = |T(x_\alpha)|$. Let $\langle S_\alpha \rangle$ be the induced subgraph on the subset S_α of vertices of the distance graph G^d. Define the function $g(S_\alpha, x_i)$ as:

$$g(S_\alpha, x_i) = \min_{G \in M(T(x_\alpha),\, \langle S_\alpha \rangle)} \left[\min_{\substack{\rho \\ \rho(x_\alpha)=x_i}} V\,(T(x_\alpha)\pi(\rho)G) \right] \tag{3}$$

i.e., $g(S_\alpha, x_i)$ is the solution to the QAP defined on the subgraphs $T(x_\alpha)$ and $\langle S_\alpha \rangle$ with the restriction that the optimum mapping should have $\rho(x_\alpha) = x_i$.

The function $g(S_\alpha, x_i)$ can be computed recursively as follows:

(i) If x_α is the predecessor of only one vertex x_β:

$$g(S_\alpha, x_i) = \min_{x_j \in S_\beta} [g(S_\beta, x_j) + f_{\alpha\beta} d_{ij}] \tag{4}$$

where $S_\alpha = S_\beta \cup \{x_i\}$.

(ii) If x_α is the predecessor of r (≥ 1) vertices $x_{\beta_1},\ldots,x_{\beta_r}$

$$g(S_\alpha,x_i) \quad \underset{S_{\beta_1},\ldots,S_{\beta_r}}{\text{Min}} \left\{ \sum_{k=1}^{r} \quad \underset{x_{j_k} \in S_{\beta_k}}{\text{Min}} \quad [g(S_{\beta_k},x_{j_k}) + f_{\alpha\beta_k} d_{ij_k}] \right\} \qquad (5)$$

where the outer minimization is over all possible sets $S_{\beta_1},\ldots,S_{\beta_r}$ with $|S_{\beta_k}| = |T(x_{\beta_k})|$, and

$$S_\alpha = S_{\beta_1} \cup \ldots \cup S_{\beta_r} \cup \{x_i\},$$

$$S_{\beta_k} \cap S_{\beta_\ell} = \emptyset \quad \forall\, k,\ell \in \{1,\ldots,r\}.$$

The initial values of $g(S_\alpha,x_i)$ are taken to be 0 for $S_\alpha = \{x_i\}$, $\forall\, x_i \in X^d$ and for all terminal vertices x_α of the directed tree G^f.

It may be worthwhile to indicate the order in which the computation of the functions $g(S_\alpha,x_i)$ would take place for the example of Fig. 3.

x_7 is a terminal vertex of G_f. We can start from $g(S_7,x_i) = 0$ with $S_7 = \{x_i\}$ $\forall\, x_i \in X^d$, and then calculate $g(S_3,x_i)$ for each $S_3 \subset X^d$ with $|S_3| = |T(3)| = 5$ and for each $x_i \in S_3$ by using expression (4) iteratively. Similarly we can compute: $g(S_8,x_i)$ $\forall\, S_8$ with $|S_8| = 3$; $g(S_{11},x_i)$ $\forall\, S_{11}$ with $|S_{11}| = 2$; $g(S_{13},x_i)$ $\forall\, S_{13}$ with $|S_{13}| = 2$; $g(S_{16},x_i)$ $\forall\, S_{16}$ with $|S_{16}| = 2$; and $g(S_{18},x_i)$ $\forall\, S_{18}$ with $|S_{18}| = 3$.

The next computation would be $g(S_{15},x_i)$ from equation (5) $\forall\, S_{15}$ with $|S_{15}| = |S_{16}| + |S_{18}| + 1 = 6$. The next computation would be $g(S_2,x_i)$ also from equation (5) $\forall\, S_2$ with $|S_2| = |S_3| + |S_8| + |S_{11}| + 1 = 11$. Finally $g(S_1,x_i)$ is computed from equation (5) for $S_1 = X^f$ (i.e., $|S_1| = |S_2| + |S_{13}| + |S_{15}| + 1$) and $\forall\, x_i \in S_1$. The value $g(S_1,x_i)$ then is the solution to the

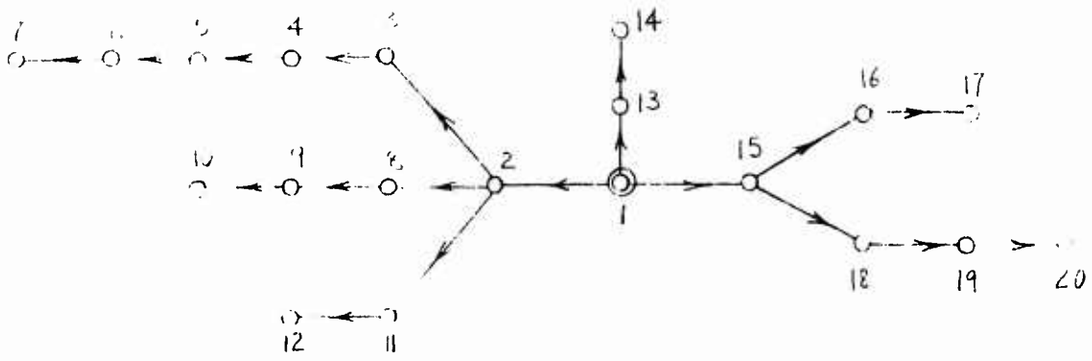


FIG 3 FLOW GRAPH FOR THE EXAMPLE

QAP with vertex 1 of the flow graph G^f mapped onto vertex x_i of the distance graph G^d. The value of the solution to the QAP would then be:

$$z = \min_{x_i \in X^d} [g(S_1, x_i)]$$

It is interesting to note that the case of G^f being a single star-graph involves only one application of iteration (5) to calculate $g(S_1, x_i)$ for each x_i and even for that single application, the linear assignment problem can be used to solve the outside minimization as mentioned earlier. This is the simplest case mentioned in section 4.A. above. The computationally most difficult case is the case of G^f being a hamiltonian chain which, as mentioned earlier, leads to the travelling salesman problem. Cases of trees G^f with values of graph diameter between these two extremes are of intermediate complexity.

C. Narrow bipartite graphs

When $G^f = (X_r, X_s)$ is the complete bipartite graph $K_{r,s}$ with $r \leq s$ the QAP can be solved when r is very small by the method of section 3.D., i.e., simply enumerating all $r!\binom{n}{r}$ mappings of the set X^r into a set of r vertices of G^d and solving an s by s assignment problem for each such mapping. Obviously this would only be practical for r = 2 or at most 3 even for graphs G^d with only 20 or 30 vertices.

D. G^f is a collection of links

For instructional purposes it may be worthwhile to note that if G^f is a disconnected graph composed of q components each of which is a single link, then if all flows are unity the QAP becomes a matching problem and can be solved as such. It is not at all clear if the problem with non-unity

flows could be solved as a matching problem. Although the matching algorithm can also be used to solve general degree-constrained partial graph problems defined on G^d, the form of the solution cannot be guaranteed to correspond to any a priori defined flow graph G^f.

5. CONCLUSION

We have expressed the QAP in terms of graph multiplication and classified and investigated cases depending on the form that the distance graph G^d and flow graph G^f take. The case when G^f and G^d are both simple star graphs was known as a solvable case of the QAP. A polynomial-growth algorithm has now been given for the solution of QAP's when both G^f and G^d are arbitrary trees. Although the algorithm for simple star graphs generalizes to the case of one graph G^f being a star and the other G^d being a complete graph, the new algorithm for arbitrary trees does not, since any such generalization implies the travelling salesman problem. However, a specialized algorithm for solving QAP's where G^f is any tree and G^d the complete graph, is described which can solve considerably larger problems than any general QAP algorithm.

REFERENCES

[1] C. Berge, The Theory of Graphs, Methuen, London (1962).

[2] V. J. Bowman, D. A. Pierce and F. Ramsey, "A Linear Programming Formulation of a Special Quadratic Assignment Problem," Carnegie-Mellon University, Management Sciences Research Report No. 277 (1971).

[3] N. Christofides, Graph Theory-An Algorithmic Approach, Academic Press, London (1975).

[4] N. Christofides and M. Gerrard, "An Improved Algorithm for the QAP," Imperial College, Report MS 76-1 (1976).

[5] P. C. Gilmore, "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem," J. SIAM, 10, 305-313 (1962).

[6] M. Held and R. Karp, "A Dynamic Programming Treatment of the Travelling Salesman Problem," J. SIAM, 10, 196-210 (1962).

[7] E. L. Lawler, "The Quadratic Assignment Problem," Man. Sci., 9, 586-594 (1963).

[8] W.L. Maxwell, "The Scheduling of Single Machine Systems: A Review," I. J. Prod. Res., 3, 177-199 (1964).

[9] J. M. Moore, "Computer Aided Facilities Design: An International Survey," I. J. Prod. Res., 12, 21-44 (1974).

[10] C. E. Nugent, T. E. Vollman and J. Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," O.R., 16, 150-173 (1968).

[11] J. F. Pierce and W. B. Crowston, "Tree-search Algorithms for the Quadratic Assignment Problem," N.R.L.Q., 18, 1-36 (1971).

[12] T. Pomentate, "On Minimization of Backboard Wiring Functions," SIAM Review, 9, 564-568 (1967).

[13] M. Scriabin and R. C. Vergin, "Relative Effectiveness of Computer and Manual Methods for Plant Layout," Man. Sci., 21, 564-568 (1975).