

AD 776 632

TECHNICAL REPORT SECTION
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93940

DEFENDING A LOGISTICS SYSTEM UNDER
MINING ATTACK*

Research Report No. 74-2

by

Peter Bartow McWhite
Center for Naval Analysis

and

H. Donald Ratliff
University of Florida

RESEARCH
REPORT

BEST
AVAILABLE COPY

*Industrial & Systems
Engineering Department
University of Florida
Gainesville, FL. 32601*

University

DEFENDING A LOGISTICS SYSTEM UNDER
MINING ATTACK*

Research Report No. 74-2

by

Peter Bartow McWhite
Center for Naval Analysis

and

H. Donald Ratliff
University of Florida

February 1974

Department of Industrial and Systems Engineering
The University of Florida
Gainesville, Florida 32611

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

*This research was supported in part under Office of Naval Research - Arlington,
Virginia; N00014-68-A-0173-0017.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
University of Florida Gainesville, Florida 32611		Unclassified	
		2b. GROUP	
		N/A	
3. REPORT TITLE			
DEFENDING A LOGISTICS SYSTEM UNDER MINING ATTACK			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
Technical Report			
5. AUTHOR(S) (First name, middle initial, last name)			
Peter Bartow McWhite H. Donald Ratliff			
6. REPORT DATE		7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 1974		21	15
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
NOO 14-69-A-0173-0017		Research Report No. 74-2	
b. PROJECT NO.			
NR 047-107/11-2-71 (431)			
c.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.			
10. DISTRIBUTION STATEMENT			
This document has been approved for public release and sale; its distribution is unlimited			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Operations Research Program Code (439) Office of Naval Research Arlington, Virginia 20360	
13. ABSTRACT			
<p>A fundamental problem in mine warfare defense is to deploy mine countermeasure resources and to route supplies so that shipping losses are minimized. The shipping losses at a port are a function of the mining attack, the quantity and duration of countermeasure efforts, and the amount shipped from the port. Models and solution algorithms are developed in this paper to optimally apportion scarce countermeasure resources when the quantity of supplies shipped out of each port is not subject to control and for the case when one can control both flow routing and countermeasures deployment. When the shipping schedule is fixed, the models are special cases of minimum cost network flow problems. For the more general problem, an enumeration algorithm is developed and computational results presented.</p>			

KEY WORDS

LINK A

LINK B

LINK C

ROLE

WT

ROLE

WT

ROLE

WT

Networks

Mine Countermeasures

Scheduling

ABSTRACT

A fundamental problem in mine warfare defense is to deploy mine countermeasure resources and to route supplies so that shipping losses are minimized. The shipping losses at a port are a function of the mining attack, the quantity and duration of countermeasure efforts, and the amount shipped from the port. Models and solution algorithms are developed in this paper to optimally apportion scarce countermeasure resources when the quantity of supplies shipped out of each port is not subject to control and for the case when one can control both flow routing and countermeasures deployment. When the shipping schedule is fixed, the models are special cases of minimum cost network flow problems. For the more general problem, an enumeration algorithm is developed and computational results presented.

I. INTRODUCTION

The problem considered here is to provide the owner or "defender" of a logistics system under mining attack with a quantitative basis for allocating the helicopters and support equipment that make up his primary mine countermeasure (MCM) resources. The logistics system consists of inland supply points that provide goods for transport over an inland supply system to seaports from which the goods are shipped. The ports in this system are the elements subject to mining attack. A diagram of the system is shown in Figure 1. All of the analysis presented here can be easily extended to the converse system where goods are shipped into seaports and then distributed to inland demand points via an inland transportation system.

In their broadest scope, the defender's responsibilities are to use his MCM and route his logistic traffic so that critical demands are met and losses are not extreme. The question of what actually constitutes optimal MCM deployment is a difficult one to answer both philosophically and computationally. We have chosen to use total expected shipping losses as the measure of effectiveness for MCM deployment. Martin and Ratliff [15] have shown that most of the concepts presented in this paper can be readily extended to similar problems where the maximum probability of lost shipping is the measure of effectiveness of an MCM deployment.

II. MODELING CONSIDERATIONS

The following assumptions are made to ease modeling the problem:

- We consider our planning horizon to be made up of H time periods.
- We assume that a mining attack has been carried out against our ports at some time before the beginning of our planning horizon, and that we can prevent further mining attack for at least the duration of the planning horizon.

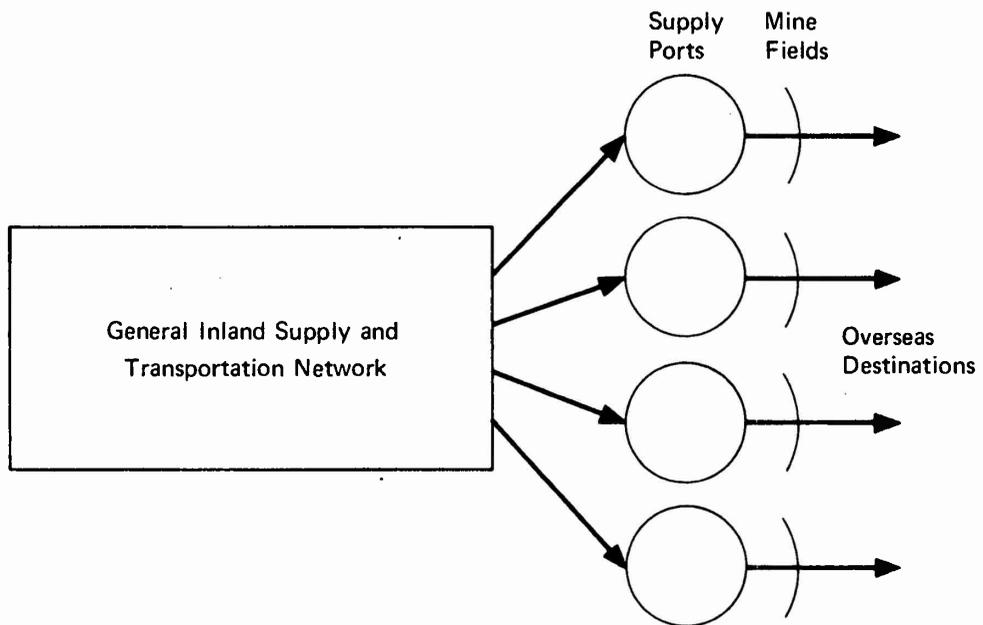


Figure 1: Logistics System Diagram

- We also assume that, given the amounts of MCMs assigned to port i in periods one through t , we have enough information to estimate a "survival factor" for port i during period $t + 1$. This survival factor can be interpreted as that fraction of goods sent out of port i in period $t + 1$ that would survive to the extent that the goods can continue on to their destination.
- It is assumed that each supply point has a fixed amount of goods that must be shipped out of the ports during the planning horizon.
- It is also assumed that the primary restrictions on the inland transportation system are the shipping times along each transportation link and the capacity of each transportation link, where these capacities can be expressed in tons of goods without regard to the type of goods being transported.

These assumptions allow us to retain most of the realism of the actual system while still being able to model the inland transportation system as a single commodity flow problem on a time-expanded network (see Ford and Fulkerson [8] p. 142).

Given all of these assumptions, the problem is then how to route the goods and deploy the MCMs so that the specified amount of goods from each supply point will be sent out during the planning horizon with the minimum possible losses. The problem can be stated equivalently as maximizing the amount of goods successfully transiting the ports during the planning horizon, given that a specified amount is sent from each supply point. Three versions of this problem are considered:

1. How to optimally route goods when the MCM deployment is already specified (fixed deployment problem).
2. How to optimally deploy MCMs when the routing of goods is already specified (fixed routing problem).

3. How to optimally route goods and deploy MCM when neither is prespecified (general problem).

In addition to providing insight into the general problem, the fixed deployment and fixed routing problems are useful in their own right. In many areas, there is now no attempt to coordinate the routing and deployment either because of other considerations that force certain routing or deployments, or simply because the two problems are the responsibility of two different organizations.

III. FIXED DEPLOYMENT PROBLEM

The fixed deployment problem can be modeled as a network flow problem by creating a node p_{ik} for each port i and time period k for $i = 1, 2, \dots, P$ and $k = 1, 2, \dots, H$, and a node s_{jt} for each supply point j and time period t for $j = 1, 2, \dots, S$ and $t = 1, 2, \dots, H$. For each $i = 1, 2, \dots, P$, an arc $(p_{ik}, p_{i,k+1})$ is constructed from p_{ik} to $p_{i,k+1}$ for $k = 1, 2, \dots, H-1$ with capacity equal to the amount that can be carried over at port i from period k to period $k + 1$. Similarly, for each $j = 1, 2, \dots, S$, an arc $(s_{jt}, s_{j,t+1})$ is constructed from s_{jt} to $s_{j,t+1}$ for $t = 1, 2, \dots, H-1$ with capacity equal to the amount that can be carried over at supply point i from period t to $t + 1$. For any two nodes s_{jt} and p_{ik} , an arc (s_{jt}, p_{ik}) is constructed from s_{jt} to p_{ik} if goods shipped from supply point j at time t were to arrive at port i at time k . Transportation links between two supply points or two demand points are constructed the same way. Transshipment points, when they exist, can be modeled the same as the ports and supply points.

Finally, a super source node V and a super sink node \bar{V} are constructed. Arcs (V, s_{j1}) are constructed from V to s_{j1} for $j = 1, 2, \dots, S$ having capacity equal to the total amount that is to be shipped from supply point j . Arcs (p_{ik}, \bar{V}) are constructed from p_{ik} to \bar{V} for $i = 1, 2, \dots, P$ and $k = 1, 2, \dots, H$

having capacity equal to the capacity of port i in period k .

An example network for a problem with two supply points, two ports, and a planning horizon of three periods is shown in Figure 2. In this example, it takes one period to get from supply point 1 to port 1, two periods to get from supply point 1 to port 2, and one period to get from supply point 2 to either port 1 or port 2. It takes one period to get from port 1 to port 2 or from port 2 to port 1.

If a_{it} is the survival factor for port i in period t , the fixed deployment problem is to find a network flow that maximizes $\sum_{i=1}^P \sum_{t=1}^H a_{it} f(p_{it}, \bar{V})$, where $f(p_{it}, \bar{V})$ is the flow on the arc (p_{it}, \bar{V}) . It is easily shown that the following procedures solve this problem:

- Step (0): Remove all arcs of the form (p_{it}, \bar{V}) from the network.
- Step (1): Among all arcs of the form (p_{it}, \bar{V}) $i = 1, 2, \dots, P$ and $t = 1, 2, \dots, H$ not currently in the network, find the one with the largest survival factor and reinsert it into the network. When none exists, the procedure is terminated and the network flow found in step (2) is optimal.
- Step (2): Maximize the flow from V to \bar{V} in the current network.
- Step (3): Set a lower bound of $f(p_{it}, \bar{V})$ on arcs of the form (p_{it}, \bar{V}) , where $f(p_{it}, \bar{V})$ is the maximum flow found in step (2), and return to step (1).

Each time the maximum flow problem is solved in step (2), the previous maximum flow can be used as a starting flow. The maximum flow problem can be solved very efficiently using the algorithm of Ford and Fulkerson [8].

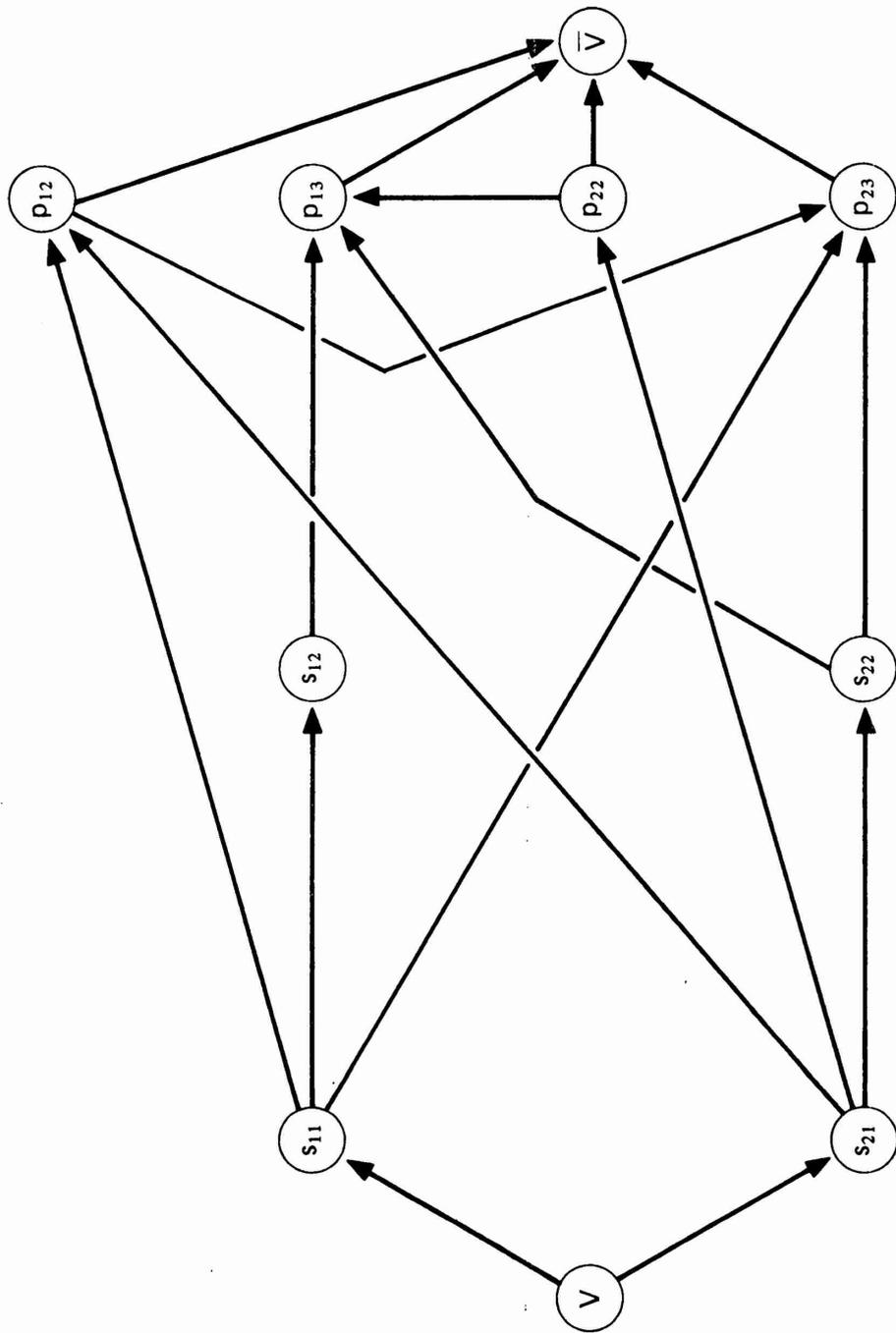


Figure 2: Example Network for Fixed Deployment Problem

IV. FIXED ROUTING PROBLEM

The fixed routing problem is considered under two different assumptions concerning MCM deployment. In the first case, we assume that the MCMs are assigned to ports at the beginning of the planning horizon and remain at their assigned ports for the duration of the planning horizon.

For $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, H$, let $a_{it}(q_i)$ represent the survival factor for port i during time period t when q_i units of MCM are assigned to port i , and let c_{it} represent the tonnage scheduled to leave port i during period t . The fixed routing problem, to maximize the amount of goods successfully transiting the ports, can then be formulated as

$$\begin{aligned} \text{Maximize} \quad & \sum_{i=1}^m \sum_{t=1}^H a_{it}(q_i) c_{it} \\ \text{subject to} \quad & \sum_{i=1}^m q_i \leq Q \\ & 0 \leq q_i \leq \bar{q}_i \text{ and integer for } i = 1, 2, \dots, m, \end{aligned} \tag{1}$$

where Q is the total number of MCMs available and \bar{q}_i is the maximum number of MCMs that can work together effectively at port i . Since a port's survival factor is, by definition, the portion of flow that successfully transits the port, it will always be the case that $0 \leq a_{it}(q_i) \leq 1$ for $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, H$.

Under the assumption that all $c_{it}(\cdot)$ for $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, H$ are concave functions, problem (1) can be solved very efficiently. This assumption is realistic in many situations since it still allows the survival factors for a port to increase or decrease from period to period, an important consideration in modeling the effect of delayed activation mines.

By defining $\bar{a}_i(q_i) = \sum_{t=1}^H a_{it}(q_i) c_{it}$,
 problem (1) becomes

$$\begin{aligned} & \text{Maximize } \sum_{i=1}^m \bar{a}_i(q_i) \\ & \text{subject to } \sum_{i=1}^m q_i \leq Q \\ & \quad 0 \leq q_i \leq \bar{q}_i \text{ and integer for } i = 1, 2, \dots, m \end{aligned}$$

Since $a_{it}(\cdot)$ for $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, H$ are concave functions, all $\bar{a}_i(\cdot)$ for $i = 1, 2, \dots, m$ are also concave functions.

It is easily shown that an optimal solution to (2) is obtained by sequentially assigning each unit of MCM where it will yield the greatest increase in the objective function. Hence, this version of the fixed routing problem can be solved using a very simple rule.

Now consider the fixed routing problem where we allow MCMs to be moved from port to port but only between periods. For example, when the ports are reasonably close together, we can use the MCMs during the day and move them from port to port at night without losing any of their effectiveness. For $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, H$, let $a_{it}(\sum_{k=1}^{t-1} q_{ik})$ represent the survival factor for port i , where q_{ik} is the number of MCMs assigned to port i in period k . Implicit in this definition of the survival factors is the assumption that the effect of using MCMs is independent of the period. This appears to be a reasonable assumption in many mine-hunting operations. The fixed routing problem under these assumptions can be formulated as

$$\begin{aligned} & \text{Maximize } \sum_{i=1}^m \sum_{t=1}^H a_{it}(\sum_{k=0}^{t-1} q_{ik}) c_{it} \\ & \text{subject to } \sum_{i=1}^m q_{it} \leq Q_t \text{ for } t = 1, 2, \dots, H \\ & \quad 0 \leq q_{it} \leq \bar{q}_{it} \text{ and integer for} \\ & \quad \quad \quad i = 1, 2, \dots, m \text{ and } t = 1, 2, \dots, H, \end{aligned} \tag{3}$$

where c_{it} is again the tonnage shipped out of port i during period t , Q_t is the number of MCMs available during period t , \bar{q}_{it} is the maximum number of MCMs which can work together efficiently at port i during period t , and q_{i0} is defined to be zero for $i = 1, 2, \dots, m$.

When we again make the assumption that all $a_{it}(\cdot)$ are concave functions for $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, H$ problem (3) can be solved very efficiently as a minimum cost network flow problem. To construct the network first create nodes n_t for $t = 1, 2, \dots, H$ and p_{it} for $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, H$. For $t = 1, 2, \dots, H$, construct an arc from n_t to p_{it} with capacity \bar{q}_{it} and cost zero for each $i = 1, 2, \dots, m$. For $t = 1, 2, \dots, H-1$ and $i = 1, 2, \dots, m$, construct an arc from p_{it} to $p_{i,t+1}$ with capacity $\sum_{k=1}^t \bar{q}_{ik}$ and cost $-a_{it}(f(p_{it}, p_{i,t+1}))c_{it}$, where $f(p_{it}, p_{i,t+1})$ is the flow in $(p_{it}, p_{i,t+1})$. Finally, construct a source node S with arcs from S to n_t having capacities Q_t and cost 0 for $t = 1, 2, \dots, H$, and a sink node T with arcs from $p_{i,H}$ to T with capacities $\sum_{t=1}^m \bar{q}_{it}$ and cost $-a_{iH}(f(p_{iH}, T))c_{iH}$ for $i = 1, 2, \dots, m$. Problem (3) is then equivalent to finding a minimum cost flow of value $\sum_{t=1}^H Q_t$ in this network. Since the costs on arcs from p_{it} to $p_{i,t+1}$ are convex functions, they can be handled as discussed by Ford and Fulkerson ([8], p. 155), and the problem can be solved using any of the very efficient minimum cost network flow algorithms. An example network for a 2-port, 3-period problem is shown in Figure 3. This network construction can be viewed as a special case of a similar construction corresponding to a production scheduling problem developed by Dorsey, Hodgson, and Ratliff [5].

V. GENERAL PROBLEM

We now consider a version of the problem where neither the supply, routing, nor MCM deployment has been predetermined. We

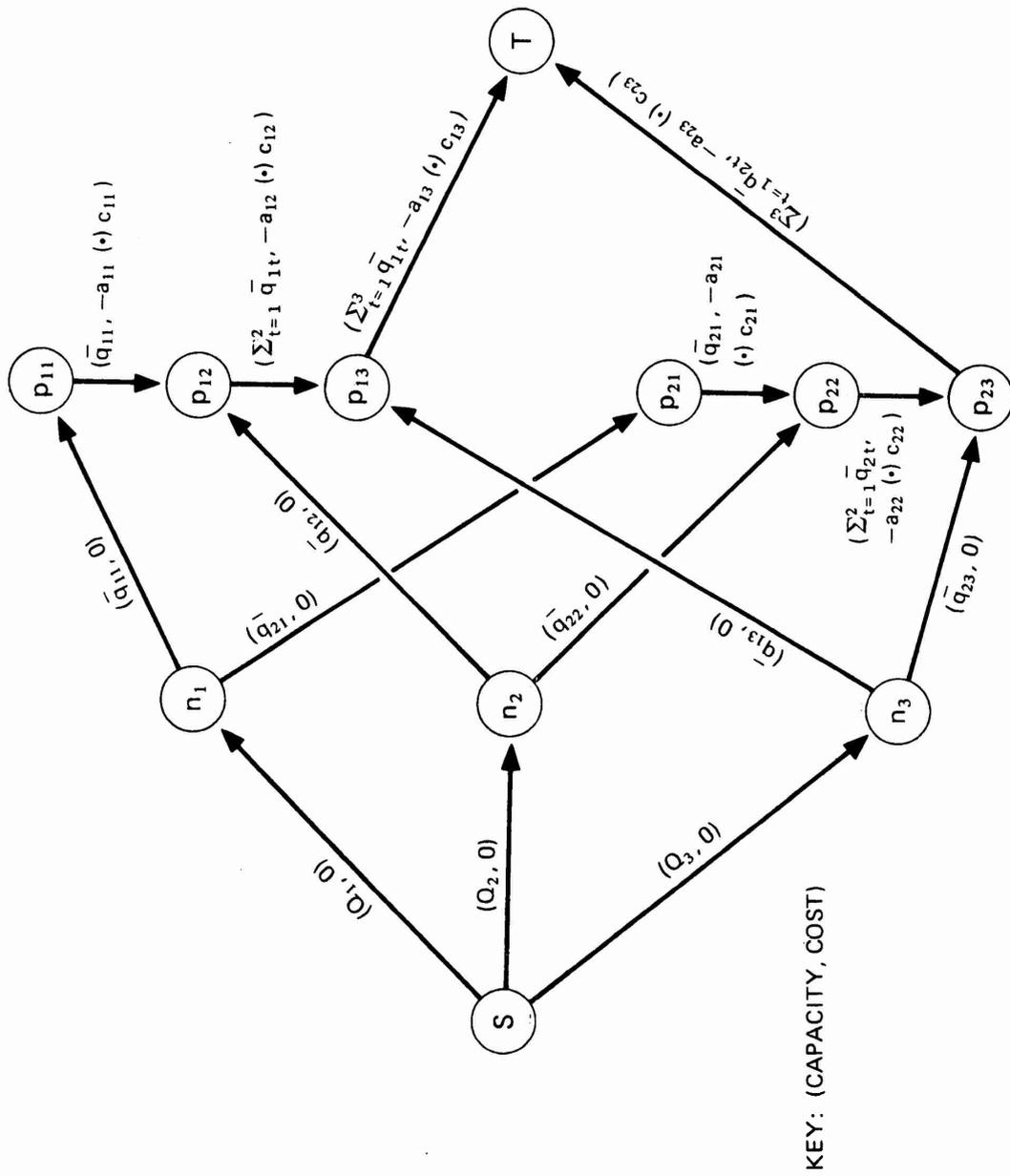


Figure 3: Network Representation for Fixed Routing Problem

again assume that MCMS are assigned to ports and remain at the same ports for the duration of the planning horizon. (The procedure discussed here readily extends to the case where MCMS can be moved from port to port between periods).

The general problem can then be formulated as

$$\begin{aligned}
 \text{Maximize} \quad & v = \sum_{i=1}^m \sum_{t=1}^H a_{it}(q_i) c_{it} \\
 \text{subject to} \quad & \sum_{i=1}^m q_i \leq Q \\
 & 0 \leq q_i \leq \bar{q}_i \text{ and integer for } i = 1, 2, \dots, m \\
 & c_{it} \in F,
 \end{aligned} \tag{4}$$

where $a_{it}(\cdot)$, \bar{q}_i , and Q are as defined for problem (1) and F represents the flow constraints for the fixed deployment problem discussed in section III. When all q_i are fixed, problem (4) is the fixed deployment problem; when all c_{it} are fixed, problem (4) is the fixed routing problem (1).

Although the constraint set is linear and therefore convex, the objective function has neither the desirable property of concavity nor the more general property of quasiconcavity. In general, to solve a nonconvex problem such as (4), one either enumerates all local optima or else develops a technique capable of rejecting and never evaluating some local optima. Falk and Soland [7] and Falk [6] have developed techniques to solve nonconvex problems where the objective function has the form $\phi(x) = \sum_{i=1}^m \phi_i(x_i)$ -- that is, a sum of functions of a single variable. However, the objective function of (4) is the sum of functions of two variables. Recently, Burdet [1] studied using a generalization of polar sets as a possible basis for a cutting plane algorithm for nonconvex problems. However, a computational algorithm has yet to be developed.

For the special case where all the $a_{it}(\cdot)$ are piecewise linear functions, problem (4) is a special case of a bilinear programming problem (that is, fixing all of the q_i or all of the c_{it} results in a linear programming problem in the other variables). Konno [11] proposed a bilinear algorithm based on Ritter's cutting plane algorithm [4] for nonconvex quadratic problems. Earlier, Mangasarian and Stone [14] and Mangasarian [13] proposed a total enumeration technique for bilinear problems. In this paper, we propose an implicit enumeration scheme to solve (4). The concept of implicit enumeration (or branch and bound) has been applied to nonconvex quadratic problems by a number of authors, including Gilmore [10], Lawler [12], and Cabot and Francis [2]. The authors cannot find any reference to implicit enumeration algorithms developed specifically for a bilinear problem.

The special structure of (4) permits us to use either of two enumeration techniques. One possibility is to enumerate over possible values of the c_{it} and solve the resulting fixed routing problems. To find all potential c_{it} , one could permute the order in which flow is sent to the seaports. As an example, for $m = 10$ and $H = 1$, one set of c_{it} values would be found by first maximizing the flow to p_1 , then to p_2 , continuing in numerical order until the remaining flow is sent to p_{10} . Another ordering would be p_2 first, then p_1 , then p_3 , etc. This type of enumeration does not appear computationally tractable since there are $(mH)!$ possible permutations.

A second possibility is to enumerate over possible values for the q_i and solve the resulting fixed deployment problem (1). Now there are at most $\prod_{i=1}^m \bar{q}_i$ values that would have to be considered. Since \bar{q}_i is usually a small number (5 or 6), this seems to be the better possibility.

An enumeration tree for the \bar{q}_i is shown in Figure 4. The fundamental notion of implicit enumeration is to sequentially commit variables along a branch until the best solution along the branch has been determined or until it can be established that there is no solution on the branch better than the best solution found so far. When one of these conditions is established, the branch is terminated and a new branch examined. An excellent discussion of implicit enumeration methods is given in Garfinkel and Nemhauser [9].

With respect to problem (4), there are some very straightforward ways to determine that a branch can be terminated. Suppose that solutions are enumerated as shown in Figure 4 (first fix a value for q_i then for q_2 , etc.). Assume that q_1, q_2, \dots, q_r have been fixed at $\hat{q}_1, \hat{q}_2, \dots, \hat{q}_r$, and that $q_{r+1}, q_{r+2}, \dots, q_{r+m}$ have not yet been fixed. Let V^* be the value of the objective function for problem (4) corresponding to the best MCM deployment found so far.

Obviously, when $\sum_{i=1}^r q_i > Q$, the branch can be terminated. When U_b is an upper bound on the value of V for this branch, the branch can also be terminated when $U_b \leq V^*$. We consider two different ways of obtaining upper bounds.

The first upper bounding technique is called the fixed deployment bound. We determine a value for total shipping output flow that is not less than the best flow that could be attained by adding any combination of free variables (variables not yet fixed) to the branch. To accomplish this end, we compute an upper bound for the shipping flow that could safely leave the active ports (ports $1, 2, \dots, r$). Then, in a separate computation, we apportion the remaining $Q - \sum_{i=1}^r \hat{q}_i$ MCMs among the free ports and obtain an upper bound on the amount of shipping that can safely leave the free ports. The sum of the active port flow and the free port flow form the fixed flow bound.

To compute the active port output flow, we set the survival functions of the active ports at the values determined by $\hat{q}_1, \hat{q}_2, \dots, \hat{q}_r$ and the survival functions of the free ports at the values determined

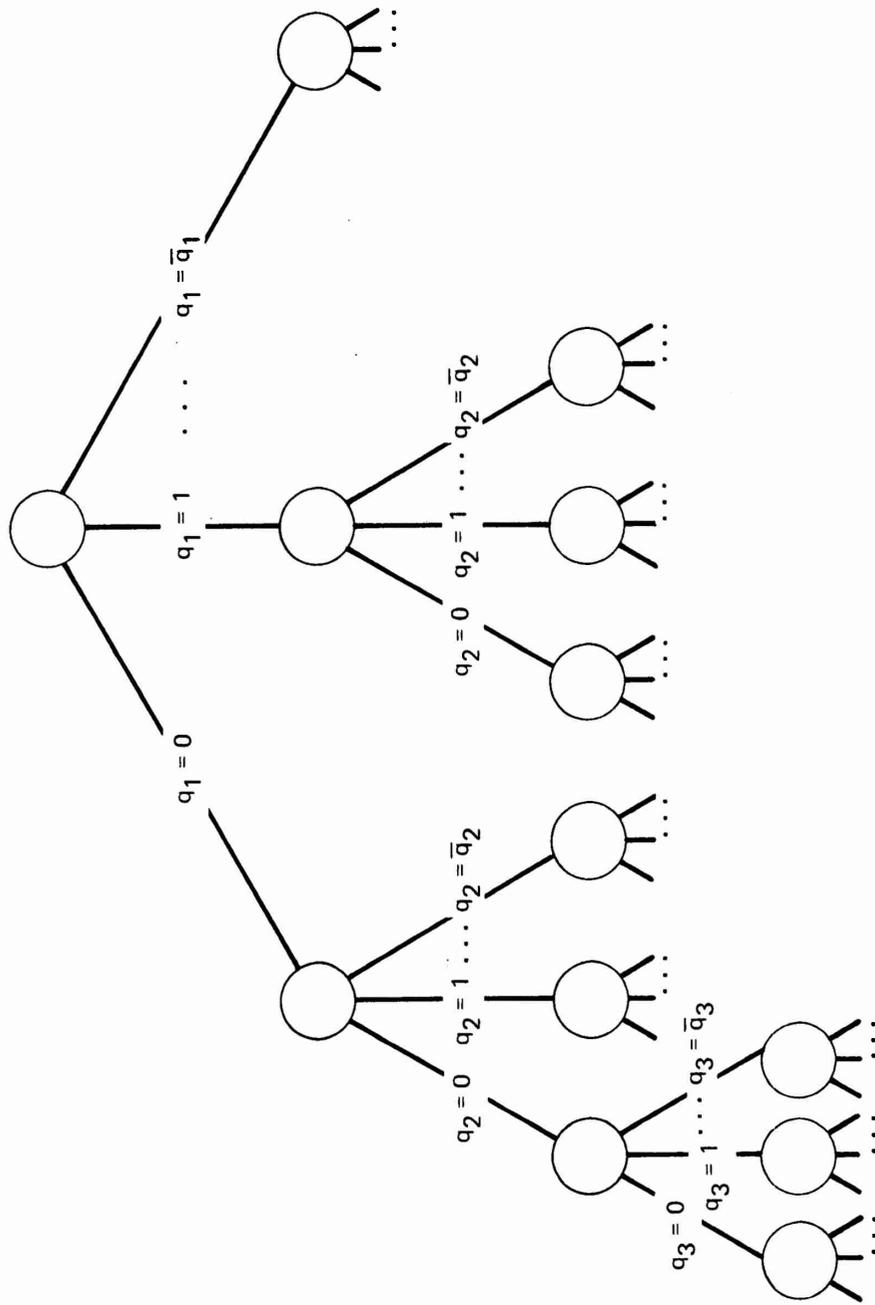


Figure 4: Enumeration Tree for the q_i

by $q_i = 0$ for $i = r+1, r+2, \dots, m$. We then solve the fixed deployment problem of section III, and the total flow that safely leaves the active ports is noted. To verify that this is an upper bound for active port output, we observe that since the survival functions of the free ports were set at their lower bound, no subsequent assignment of MCMs to the free ports could increase the amount of shipping flow to the active ports. The survival factors at the present active ports will not change when additional ports are added to the branch. Therefore, we have a valid upper bound for the shipping that safely leaves the active ports.

For the network in section III, let f_{it} be the maximum flow from node V to node p_{it} . The f_{it} values can be calculated before starting the enumeration. An upper bound on the output flow from the free ports can be obtained by solving problem (1) with $c_{it} = f_{it}$ for $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, H$ and $q_i = \hat{q}_i$ for $i = 1, 2, \dots, r$. This value plus the previously determined upper bound on active port output flow is a valid upper bound on V .

A second upper bound on V can be obtained by setting $q_i = \hat{q}_i$ for $i = 1, 2, \dots, r$ and $q_i = \bar{q}_i$ for $i = r+1, r+2, \dots, m$, then solving the fixed deployment problem of section III. Both of these bounds were used in the enumeration scheme tested.

An initial MCM deployment was generated by first setting $q_i = 0$ for $i = 1, 2, \dots, m$ and solving the fixed deployment problem of section III. Based on the port flows generated by this problem the MCM assignment (q_i) was increased by one at the port yielding the largest increase in the objective function value for one additional unit of MCM. The fixed deployment problem was then solved again, and the procedure repeated until all MCMs were assigned. While this procedure does not always generate an optimal solution, for the problems tested it produced solutions within 10 percent of the optimal value in all cases.

Finally, given any specification of the q_i for $i = 1, 2, \dots, m$, solving the fixed deployment problem gives an optional set of c_{it} for $i = 1, 2, \dots, m$ and $t = 1, 2, \dots, H$. Using this c_{it} set, solving

problem (1) produces a q_i set that is as good and possibly better than the original set. This process can be repeated until no further improvement is obtained, to produce a solution to problem (4), which is in a sense locally optimal. This procedure is obviously finite, since repeating a set of q_i 's would cause the procedure to stop. The procedure was applied every time a complete specification of the q_i 's occurred on a branch. This was done to generate a "good" solution as quickly as possible in case computation had to be terminated before an optimal solution had been established.

VI. COMPUTATIONAL RESULTS

The enumeration algorithm was coded in PL/1 and tested on an IBM 370/165 computer. The $a_{it}(\cdot)$ used in the tests were of the form shown in Figure 5. When the survival functions are of this form, the functions $\bar{a}_i(\cdot)$ of problem (2) are piecewise linear of the form shown in Figure 6. When using the procedure suggested to solve problem (2) for a given c_{it} set, all except possibly one port would have $q_i = \bar{q}_{it}$ for some $t = 1, 2, \dots, H$. This limits, to some extent, the values of q_i that must be considered in the enumeration. This fact was taken advantage of in the computational tests.

All computational times are execution times (CPU times) on an IBM 370/165 computer. The results presented below and in Figures 7 and 8 represent the average optimal solution times required to solve six problems for each data point. Randomly generated supply networks consisting of a single source supplying the indicated number of seaports via five intermediate terminals were used to obtain these results:

Number of seaports	5	6	7	4	4	4
Length of planning horizon	3	3	3	8	10	12
Number of problems	6	6	6	6	6	6
Minimum CPU time (sec.)	1.2	2.7	8.2	12.1	10.5	17.2
Maximum CPU time (sec.)	8.1	29.0	82.0	24.2	38.5	62.6
Average CPU time (sec.)	5.1	14.4	48.2	17.8	24.8	31.0

Figure 7 indicates that excessive computational times may result when problems with more than seven ports are solved. However, Figure 8 indicates that increasing the length of the planning horizon does not affect computational time as severely as does increasing the number of ports. This is about what one would expect since increasing the length of the planning horizon does not increase the size of the set being enumerated.

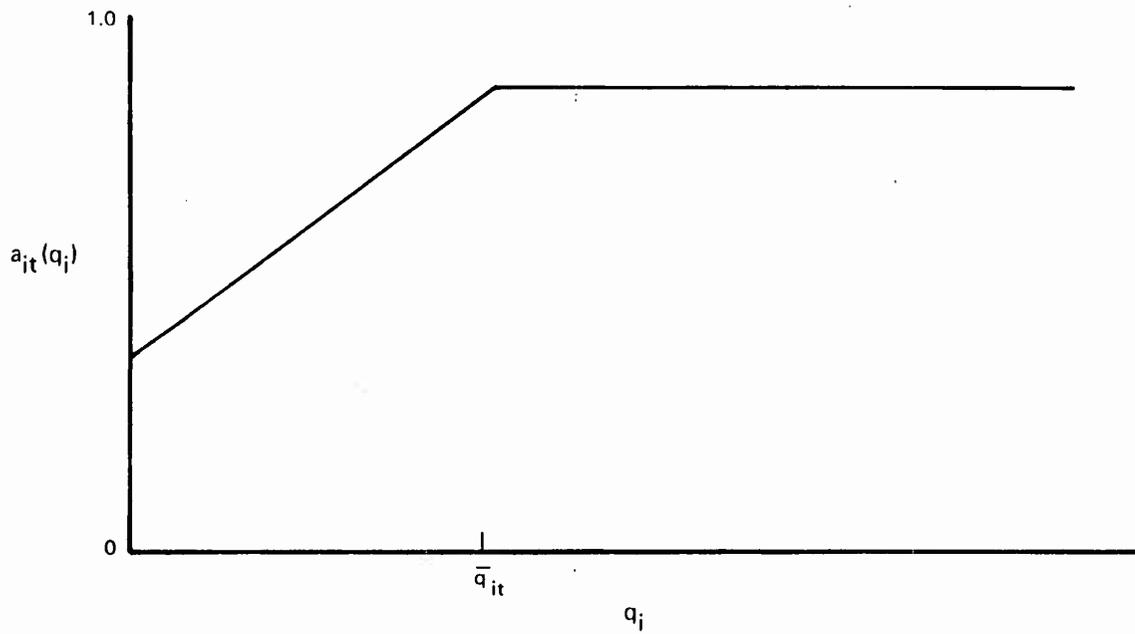


Figure 5: Form of $a_{it}(\cdot)$ Used for Testing Enumeration

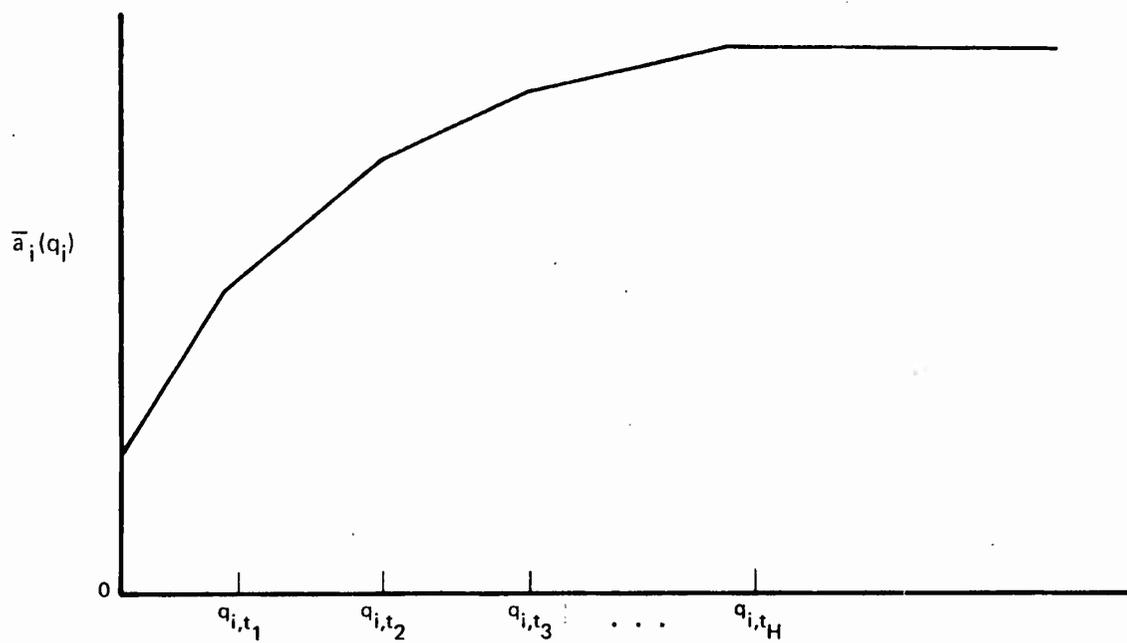


Figure 6: Form of $\bar{a}_i(\cdot)$ Used for Testing Enumeration

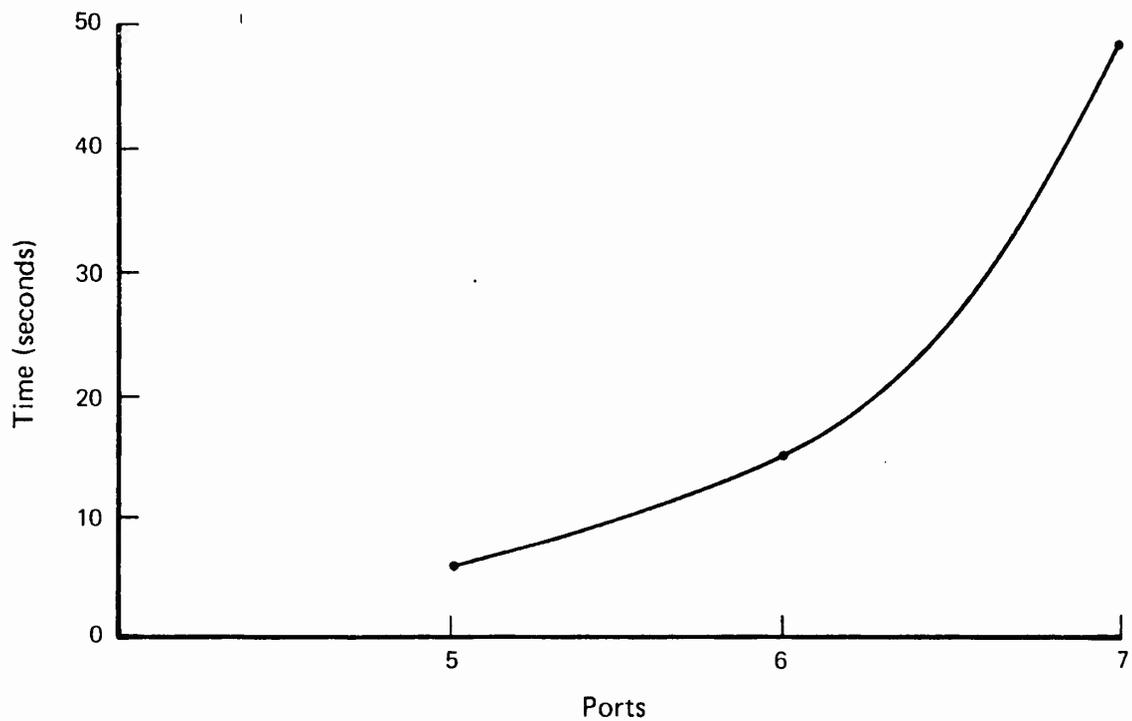


Figure 7: Average Solution Times for Problems with a Three Period Planning Horizon

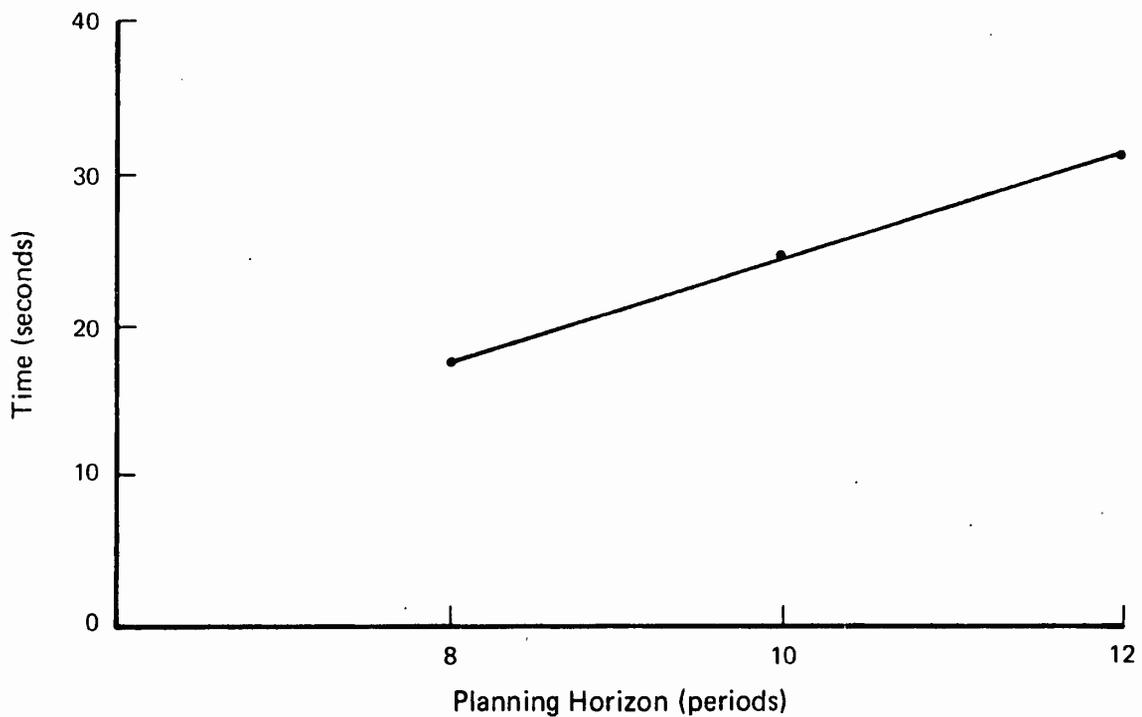


Figure 8: Average Solution Times for Problems with Four Ports

REFERENCES

1. Burdet, C.A., "Polaroids: A New Tool in Non-Convex and in Integer Programming," *Naval Research Logistics Quarterly* 20 13-24 (1973).
2. Cabot, A.V. and Francis, R.L., "Solving Certain Nonconvex Quadratic Minimization Problems by Ranking the Extreme Points," *Operations Research* 18, 32-86 (1970).
3. Cooper, L., "The Transportation-Location Problem," *Operations Research* 20, 94-108 (1972).
4. Cottle, R.W., and Mylander, W.C., "Ritter's Cutting Plan Method for Quadratic Programming," Technical Report No. 69-11, Operations Research House, Stanford University, California (1969).
5. Dorsey, R.C. Hodgson, T.J., and Ratliff, H.D., "A Network Approach to a Multi-Facility, Multi-Product Scheduling Problem with Back Ordering," Research Report 73-6, University of Florida 1973.
6. Falk, J.E., "Lagrange Multipliers and Nonconvex Programs," *SIAM Journal on Control* 7, 534-545 (1969).
7. Falk, J.E., and Soland, R.M., "An Algorithm for Separable Non-convex Programming Problems," *Management Science* 15, 550-569 (1969).
8. Ford, L.R., Jr., and Fulkerson, D.R., Flows in Networks (Princeton University Press, Princeton, 1962).
9. Garfinkel, R.S., and Nemhauser, G.L., Integer Programming (John Wiley & Sons, New York, 1972).
10. Gilmore, P.C., "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem," *Journal Society of Industrial and Applied Math* 10, 305-313 (1962).
11. Konno, H., "Bilinear Programming: Part I. Algorithm for Solving Bilinear Programs," Technical Report No. 71-9 Department of Operations Research, Stanford University (1971).
12. Lawler, E.L., "The Quadratic Assignment Problem," *Management Science* 9, 586-599 (1963).
13. Mangasarian, O.L., "Equilibrium Points of Bimatrix Games," *Journal Society of Industrial and Applied Math* 12, 778-780 (1964).

14. Mangasarian, O.L., and Stone, H., "Two-Person Nonzero-Sum Games and Quadratic Programming," *Journal Math Analysis and Applications* 9, 348-355 (1964).
15. Martin, L.A., and Ratliff, H.D., Unpublished Report, University of Florida.

U159171