NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

INTERDICTION OF A TRANSPORTATION NETWORK

by

Charles Putnam Preston, Jr.

Thesis Advisor: G.T. Howard

March 1972

Approved for public release; distribution unlimited.

Security Classification

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) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Interdiction of a Transportation Network

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; March 1972

5. AUTHOR(S) (First name, middle initial, last name)

Charles P. Preston, Jr.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1972	58	7

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School

13. ABSTRACT

The problem of determining the optimum allocation of aircraft to an airstrike against a transportation network is investigated. The damage function is assumed to be exponential. A solution procedure is developed utilizing dynamic programming and integer solutions are found. The number of aircraft to be assigned to the airstrike is considered a decision variable. A sensitivity analysis is run to determine the optimum value for this variable.

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
interdiction						
networks						

DD FORM 1 NOV 65 1473 (BACK)

S/N 0101-807-6821

Security Classification

A-31409

Interdiction of a Transportation Network

by

Charles Putnam Preston, Jr.
Captain, United States Marine Corps
B.S., Georgia Institute of Technology, 1963

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1972

Author Charles Putnam Preston Jr

Approved by Gilbert T. Howard
Thesis Advisor

Jack R. Borsting
Chairman, Department of Operations Research
and Administrative Sciences

Milton U. Clauser
Academic Dean

ABSTRACT

The problem of determining the optimum allocation of aircraft to an airstrike against a transportation network is investigated. The damage function is assumed to be exponential. A solution procedure is developed utilizing dynamic programming and integer solutions are found. The number of aircraft to be assigned to the airstrike is considered a decision variable. A sensitivity analysis is run to determine the optimum value for this variable.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

A. OBJECTIVE

The purpose of this paper is to present a procedure for determining the optimal allocation of aircraft to a single airstrike against a transportation network. This allocation problem is solved by dynamic programming and a Fortran-coded version of the program is included in the paper.

B. GENERAL

Sustained ground operations require a military force to have some means of resupply. This resupply capability is partially dependent upon a land transportation system. The level of resupply effort required depends upon what type of forces are being supported. Guerrilla forces enjoying local support require less resupply capability in terms of pounds per man per day than would a conventional army, but a greater percentage of this capability depends upon land transportation networks.

Any reduction in the resupply capability of a military force will reduce its combat effectiveness. Tactical air interdiction has been used extensively by the Armed Forces of the United States against its opponents in Southeast Asia to accomplish this reduction.

There are at least three alternative means of using tactical air to reduce the resupply capability of an enemy. Aircraft may be assigned to attack sources of supply to destroy war material before it enters the transportation system and/or to disrupt its production; aircraft can destroy war material as it moves in the transportation system; and finally aircraft can attempt to reduce the resupply capacity of the transportation system itself by destroying bridges, roads, railroads, et cetera. Conventional wisdom argues that the first course of action is the most effective form of interdiction. Unfortunately for military planners, political considerations may rule out this alternative. This paper will focus on the last of these options, the reduction in capacity of the transportation system itself.

C. BACKGROUND

Considerable effort has been devoted to the interdiction problem. In particular two recent papers provided the background for the approach to the problem developed in this paper. McMasters and Mustin [1] developed an algorithm that determines which arcs of a transportation network should be attacked and at what level of effort given a limited availability of resources. In this formulation of the problem the relationship between arc capacity and resource allocation (damage function) was assumed to be linear. The algorithm presented is based upon the max-flow min-cut theorem of Ford and Fulkerson [2] and the relationship between a primal network and its topological dual.

Nugent [3] investigated the same problem under the assumption of an exponential damage function which exhibits diminishing marginal returns. An algorithm was developed that finds a non-integer solution to the problem.

In this paper the transportation system will have the same network formulation as in Refs. 1 and 3. The problem will be formulated differently and dynamic programming will be used to provide integer solutions.

D. INTERDICTION PROBLEM

It will be assumed that, given unlimited aircraft availability, the assignment of aircraft to an airstrike would reach a point beyond which it would become uneconomical to assign further aircraft. In a problem with constraints on aircraft availability this point might or might not occur before all available aircraft were assigned. For this reason, the objective of an operations officer planning an airstrike against a transportation network is not merely to minimize network capacity subject to aircraft availability, but to minimize the capacity subject to aircraft availability and the additional consideration that the cost of any incremental assignment of aircraft to the strike is exceeded by the benefit resulting from that assignment.

To accomplish the objective the strike planner must have information on the availability and cost of assignment of aircraft. Detailed information must be available concerning the transportation network including the upper and lower bounds on the capacity of each arc and its vulnerability to attack. The planner must also know the benefit to attribute to a reduction in resupply capability. With this information and using the procedure that will be outlined the planner can determine: how many aircraft to assigne to the airstrike; which arcs in the network should be attacked; how many aircraft to as.ign to arcs that will be attacked; and the capacity of the network after the airstrike.

II. THE MODEL

A. NETWORK DESCRIPTION

The transportation system under consideration is represented by a planar connected graph of nodes and undirected capacitated arcs. Arcs represent road segments and nodes represent either a road intersection or any other point where it is necessary to distinguish between road characteristics on either side of the node. Three constants are associated with each arc representing the upper and lower bounds on arc capacity and the arc's vulnerability parameter.

It is assumed that the network has one source node from which flow originates and one sink node at which flow terminates. If the transportation system being modeled has more than one originating point or terminating point this may be handled by creating a super-source and/or sink with artificial arcs connecting these super-nodes to sources and sinks as needed. These artificial arcs may not be attacked and their capacities are unbounded. The arc between nodes i and j is represented by (i,j). Nodes are numbered from 1 to n with 1 corresponding to the source and n the sink. With the exception of the source and the sink, flow conservation is assumed to hold. That is, flow out of node i equals flow into node i.

The flow in arc (i,j) is designated as x_{ij} if it is from node i to j and x_{ji} if it is from node j to i. This avoids the necessity of defining negative flows. Flow is assumed to be from the source to the sink although it may be in either direction in the intermediate arcs. The model as formulated considers only flows of a single commodity, tons of resupply per day, and the value of one unit of flow is assumed to be the same for all arcs.

Capacities on arcs represent bounds on flow in either direction. The capacity on arc (i,j) is given by m_{ij} and is assumed to be the same in both directions. The flow in arc (i,j) is restricted by

$$0 \leq x_{ij} \leq m_{ij} .$$

The upper and lower bounds on the capacity of arc (i,j) are represented by u_{ij} and l_{ij} where

$$0 \leq l_{ij} \leq m_{ij} \leq u_{ij} .$$

The vulnerable portion of an arc's capacity is designated w_{ij} with

$$w_{ij} = u_{ij} - l_{ij} .$$

The amount of resource allocated to interdict arc (i,j) is denoted by k_{ij}. The relationship between the capacity of arc (i,j) and the level of resource assigned to its interdiction is defined as the damage function of arc (i,j) and is given by

$$m_{ij}(k_{ij}) = l_{ij} + w_{ij} \exp(-b_{ij}k_{ij}) .$$

In the above damage function the parameter b_{ij} is a measure of the vulnerability of arc (i,j). Larger values of b_{ij} result in greater reductions in capacity for fixed values of l_{ij}, w_{ij} and k_{ij} and hence imply greater vulnerability. If $b_{ij} = 0$ then $m_{ij}(k_{ij}) = u_{ij}$ for all possible values of k_{ij} and no capacity reduction is possible. With this damage function if no aircraft are assigned to (i,j), its capacity will be u_{ij} and in the limit as the number of aircraft assigned to (i,j) becomes infinite the capacity approaches l_{ij}. This lower bound will be referred to as arc capacity after unlimited interdiction.

B. DETERMINATION OF NETWORK CAPACITY

The opposition is assumed to have the means to determine how to maximize the flow in the transportation network. Let the capacity of the network be defined as this maximal flow. The determination of maximum flow is the well-known maximal flow problem and may be found using the max-flow labeling algorithm based upon the max-flow min-cut theorem of Ford and Fulkerson [2]. Ford and Fulkerson's theorem states that the maximum flow possible in a network is equal to the value of the minimal cut set. In this paper the value of a cut set will be referred to as its capacity.

C. ENUMERATION OF CUT SETS

The network capacity has been defined to be equal to the maximum flow possible in the network. As discussed, this maximum flow is equal to the value of the minimum cut set. Therefore, the problem of minimizing this capacity is equivalent to minimizing the capacity of some cut set. It is obvious that aircraft will be allocated to only one cut set since if this were not the case all aircraft could have been assigned to the cut set that was minimal after the first allocation with a resulting decrease in network capacity.

The complicating factor is that there is no easy way to find out which cut set should be selected for attack. To solve the problem it is necessary to have some means of identifying cut sets. In addition, it is desirable to be able to identify these cut sets in order of increasing capacity after unlimited interdiction since once a cut set is found whose capacity after unlimited interdiction is greater than or equal to network capacity before interdiction no more cut sets need be identified. The network capacity before interdiction represents an upper bound on network capacity. Define S_i as the cut set with the i^{th} smallest capacity after unlimited interdiction. The set of S_i whose capacities after unlimited interdiction is less than the upper bound on network capacity will be denoted by S.

The method by which cut sets are identified makes use of the topological dual of a network. Arcs have lengths rather than capacities in the dual network. The cut sets in the primal network have a one-to-one correspondence with the loopless paths in the dual. The problem of finding the shortest path from the dual source to the dual sink corresponds to the primal problem of finding the minimum cut set. The length of the dual shortest path equals the primal capacity.

The topological dual of a given primal network is constructed as follows:

(1) Connect the source and the sink of the primal with an artificial arc. Call the result the modified primal.

(2) Place a node in the area surrounding the modified primal (external face) and one in each face formed by the arcs of the modified primal. Let the dual source be the node in the external face and the dual sink be the node in the face involving the artificial arc.

(3) For each arc in the primal (except the artificial arc) construct a dual arc that intersects it and joins the two nodes in the faces adjacent to it.

(4) Assign each dual arc a length equal to the capacity of the primal arc it intersects.

Once the dual network has been developed, the shortest path through the dual before interdiction is found. This path is determined using the upper bounds on primal capacities as lengths of arcs in the dual. The length of this path represents network capacity before interdiction. Any shortest path algorithm may be used for this determination. Dreyfus [4] evaluated several of these algorithms concluding that the procedure developed by Dijkstra is the most efficient. Next the lengths of the dual arcs are changed to correspond to the lower bounds on primal arc capacities. The lengths of the dual paths now represent the capacities of the corresponding primal cut sets after unlimited interdiction. Paths with loops need not be considered since they correspond to primal cut sets that either include more arcs than necessary to sever the network or contain some arc more than once. The dual paths are identified in order of increasing length by means of an n^{th} shortest path algorithm. Clarke, Krikorian and Rausen [5] developed an algorithm for determining the n best loopless paths, but it is difficult to apply. Pollack [6] in an unpublished paper presented an algorithm which successively develops the best loopless paths using extensions of shortest path algorithms. This procedure is less complex than that of Clarke, Krikorian and Rausen and appears to be more efficient. It should be noted that depending on the number of elements in S and the total number of paths in the dual, the most efficient means of developing S may be to enumerate all paths through the dual and then compare lengths.

III. ANALYSIS OF THE MODEL

A. MATHEMATICAL FORMULATION

The problem, as outlined previously, is to find that allocation of aircraft to an airstrike against a transportation network which will minimize the capacity of that network. This minimization is accomplished subject to a constraint on aircraft availability and the consideration that the incremental benefit of assigning aircraft must exceed the incremental cost of that assignment. If net benefit is defined to be the difference between the total benefit derived from the airstrike and the total cost of aircraft assignment the problem may be restated as follows: maximize net benefit subject to aircraft availability.

Let K represent the total number of aircraft available for assignment to the airstrike and let K* be the number of aircraft that have been assigned to the airstrike. Then for any choice of K* the problem may be stated mathematically as

$$\min_{S_i \varepsilon S} \text{[cut set capacity after optimal interdiction]}$$

or

$$\min_{S_i \varepsilon S} \left[\min \sum_{(i,j)\varepsilon S_i} (l_{ij} + w_{ij} \exp\{-b_{ij}k_{ij}\})\right]$$

$$\text{subject to} \quad \sum_{(i,j)\varepsilon S_i} k_{ij} \leq K^*$$

$$k_{ij} \quad \text{positive integer .}$$

The structure of this problem will allow the development of an efficient solution procedure. Note that with respect to a particular cut set the objective is to minimize its capacity. Since the cut set capacity is the sum of functions that are convex in k_{ij}, this capacity is a convex function and is therefore unimodal with respect to minimization. The overall objective function is the minimum of a set of convex functions and is neither concave nor convex. This together with the problem of not knowing which cut set is going to be attacked requires that each cut set in S be the subject of a minimization problem.

For a particular cut set, S_i, the problem is

$$\min \sum_{(i,j)\varepsilon S_i} (l_{ij} + w_{ij} \exp[-b_{ij}k_{ij}])$$

$$\text{subject to} \quad \sum_{(i,j)\varepsilon S_i} k_{ij} \leq K$$

$$k_{ij} \text{ positive integer .}$$

The term $\sum_{(i,j)} l_{ij}$ is constant and may be deleted during the minimization and then added back to give the solution in terms of capacity. This problem will be solved by means of dynamic programming. Each arc in the cut set under consideration will be represented by a stage in the dynamic program.

Let the number of arcs be n and resubscript each arc (i,j) and its associated parameters in any order with the single subscript i running from one to n. The decision variable for stage i is k_i and the return function for stage i is given by

$$r_i = w_i \exp(-b_i k_i) .$$

The state variable for stage i will be denoted by x_i and represents the remaining resource availability at stage i. The problem may be restated as finding $f_n(x_n)$ where

$$f_n(x_n) \;=\; \min_{0 \le k_n \le x_n} \sum_{i=1}^{n} r_i(x_i)$$

$$\text{subject to} \quad x_{i-1} = x_i - k_i \;.$$

$F_n(x_n)$ is the optimal return from stages n,n-1,...,1 given x_n units of resource. The above problem may be solved by dynamic programming since $f_n(x_n)$ can be decomposed into a series of single variable problems. Nemhauser [7] shows that problems with additive stage returns may always be decomposed. Therefore, the following recursive relationship is valid

$$f_n(x_n) \;=\; \min_{0 \le k_n \le x_n} \;[r_n(k_n) + f_{n-1}(x_{n-1})]\;,\; n>1$$

and

$$f_1(x_1) \;=\; \min_{0 \le k_1 \le x_1} \; r_1(x_1) \;.$$

The value of x_{n-1} is given by

$$x_{n-1} \;=\; t_n(x_n,k_n) \;=\; x_n - k_n$$

where t_n is the transformation which gives the relationship between the amount of resource remaining after stage n given that x_n was available before stage n and k_n was utilized at stage n.

The dynamic program is solved by starting at stage one and working to stage n solving a series of single variable minimizations. These minimizations are facilitated by the convexity of the individual stage returns. Nemhauser [7] provides a proof of the fact that in the minimization of additive stage returns the convexity of each stage return ensures that $f_i(x_i)$ is a convex function of x_i. This means that each single variable optimization performed in the dynamic program is of a unimodal function and permits the use of Fibonacci search to find the optimal values of the decision variables. An application of this technique is found in Ref. 7.

After the optimal allocation of aircraft within each cut set in S is found for a given K*, their capacities are compared. The cut set with the minimum capacity is the one that would be attacked if K* aircraft were to be assigned to the airstrike. The capacity of this minimal cut set is by definition the network capacity and this capacity will be a strictly decreasing function of K*. The remaining problem is to determine how many aircraft to assign to the strike in order to maximize net benefit. To make this determination it is necessary to know the cost of allocating aircraft to the strike. This will be assumed to be a constant C dollars per aircraft. The benefit derived from network capacity reduction must also be known. It will be assumed to be a constant D dollars per unit capacity reduction.

With the above information the problem of determining how many aircraft to allocate may be determined by comparing the incremental cost of assigning aircraft to the benefit resulting from that assignment. To make this comparison it is necessary to define the benefit resulting from the assignment of a single aircraft. This will be defined as the product of the benefit per unit capacity reduction (D) and the amount of capacity reduction that can be achieved by that aircraft.

The amount of capacity reduction that can be achieved by one additional aircraft is a function of the number already assigned and will be denoted as $\delta(K^*)$. A simple decision rule is to assign aircraft $K^* = 1,2,\ldots$ until a point is reached where benefit from the last aircraft assigned does not exceed the cost of assignment. At this point

$$D * \delta(K^*) \leq C$$

or

$$\delta(K^*) \leq C/D$$

and the optimal allocation of aircraft is K^*-1. If $\delta(K^*) > C/D$ for all K^* the optimal allocation is K under this rule. There would be no problems with this decision rule if $\delta(K^*)$ were a non-increasing function of K^*. In this case once a K^* was found such that $\delta(K^*) \leq C/D$ the cost of any further assignment of aircraft would exceed its benefit.

If network capacity after optimal interdiction were determined by only one cut set for all values of K^* then $\delta(K^*)$ would be non-increasing. This is not the case. In general as K^* ranges from 0 to K different cut sets are minimal (see Figure 1). At $K^* = 0$ the cut set that determines network capacity is by definition the one that is minimal before any interdiction takes place. Unless this cut set is also minimal after unlimited interdiction, at some point another cut set must determine network capacity. This crossover may, of course, occur after assignment of all available aircraft. These points where a change in the constraining cut set occurs represent points where $\delta(K^*)$ increases with respect to K^*. Therefore, there is no guarantee that stopping when $\delta(K^*) \leq C/D$ for the first time is optimal. If at some point after further assignment of aircraft is made $\delta(K^*)$ again exceeds C/D, it may be that further assignment of aircraft would have resulted in benefits outweighing costs.

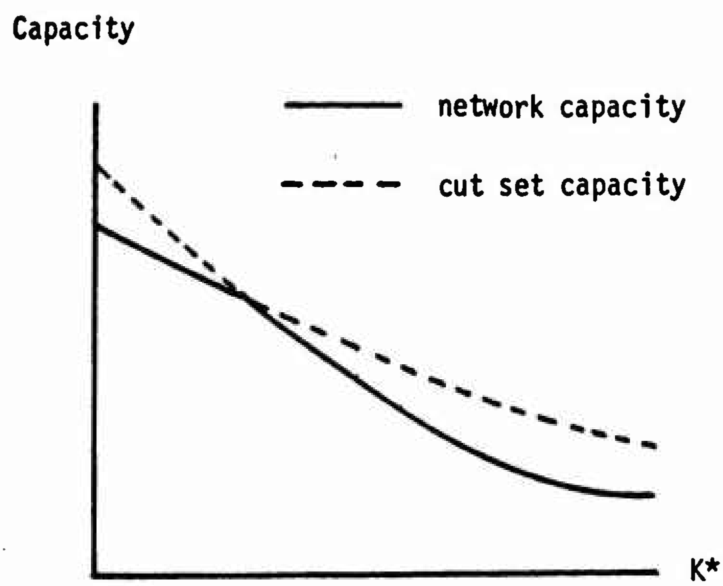

Figure 1. Network Capacity

The problem of determining the optimal K* will be handled as follows:

(1) Find the first value of K* for which $\delta(K^*) \leq C/D$. Subtract one aircraft and let the resulting value of K* be K_1^*. If K* = K before K_1^* is found then the optimal allocation of aircraft is K.

(2) Check to see if $\delta(K^*) > C/D$ for any values of $K^* > K_1^*$. If not go to step (4). If so find the next value of K* for which $\delta(K^*) \leq C/D$. Let this number minus one be K_2^*.

(3) Continue in this manner to identify the K* at which $\delta(K^*)$ becomes $\leq C/D$ after there has been an intervening value of K* such that $\delta(K^*) > C/D$. Subtracting one aircraft each time, label the resulting values $K_3^*, K_4^*, \ldots, K_n^*$. If $\delta(K) > C/D$ let $K_n^* = K$. Let K_0^* be defined as 0.

(4) Starting with i = 1 and continuing until i = n, check whether or not the cost to reach K_i from K_{i-1} is exceeded by the benefit. If it is,

let $K^*_{opt} = K_i^*$, increment i by one, and go to the beginning of step (4). If it is not, go to step (5).

(5) Starting with $l = 1$ and continuing until $l = n-i$ check whether the cost to reach K_{i+l} from K_{i-1} is exceeded by the benefit. If it is, let $K^*_{opt} = K^*_{i+l}$, let $i = i+l+1$ and go to step (4). If not, increment l by one and go to the beginning of step (5).

At the end of this procedure K^*_{opt} will be the optimal number of aircraft to assign to the airstrike and the problem will be solved.

B. STEPWISE SOLUTION PROCEDURE

(1) Formulate the topological dual of the transportation network. Find the shortest path through the dual before interdiction. This represents an upper bound on network capacity.

(2) Use Pollack's algorithm [6] to identify the first, second, third, ... shortest paths through the dual using the lower bounds. Continue identifying paths until one is found whose length exceeds the previously found upper bound on network capacity. Let the primal cut sets corresponding to these paths be denoted as set S.

(3) For each cut set that is an element of S, use dynamic programming to find the optimal allocation of aircraft and the resulting capacity for K^* equal to $1,2,\ldots,K$.

(4) For each value of K^* find the network capacity by taking the minimum of the capacities of the elements of S.

(5) Construct the function $\delta(K^*)$ and determine $K_1^*, K_2^*,\ldots,K_n^*$.

(6) Using the procedure previously outlined determine which K_i^* is optimal.

C. SAMPLE PROBLEM

The diagram in Figure 2 represents a hypothetical transportation network. The three numbers associated with each arc are b_{ij}, l_{ij}, and u_{ij}.

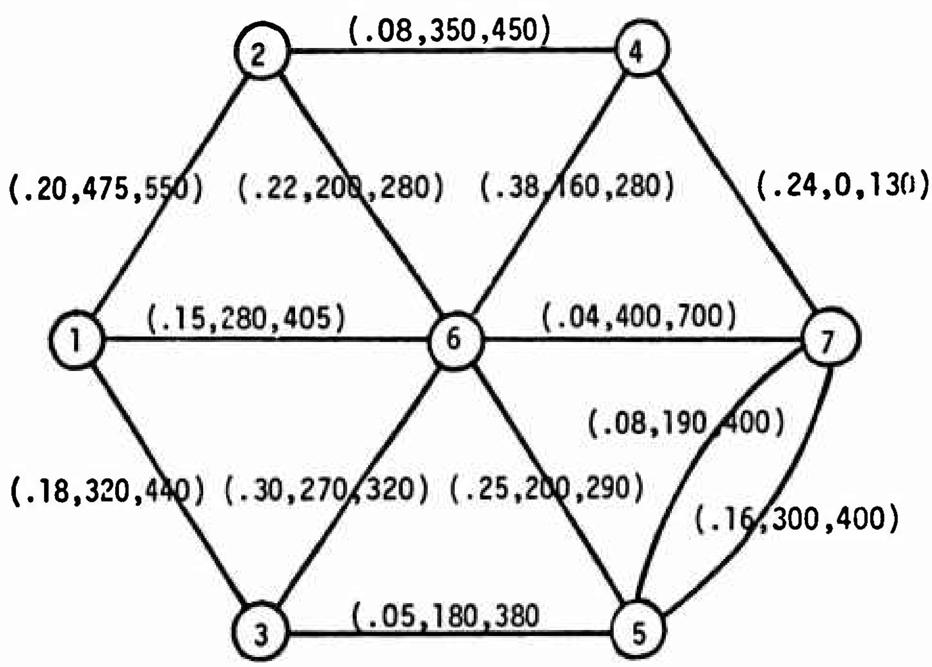

Figure 2. An Example Network

Figure 3 shows how the topological dual of the transportation network is constructed.

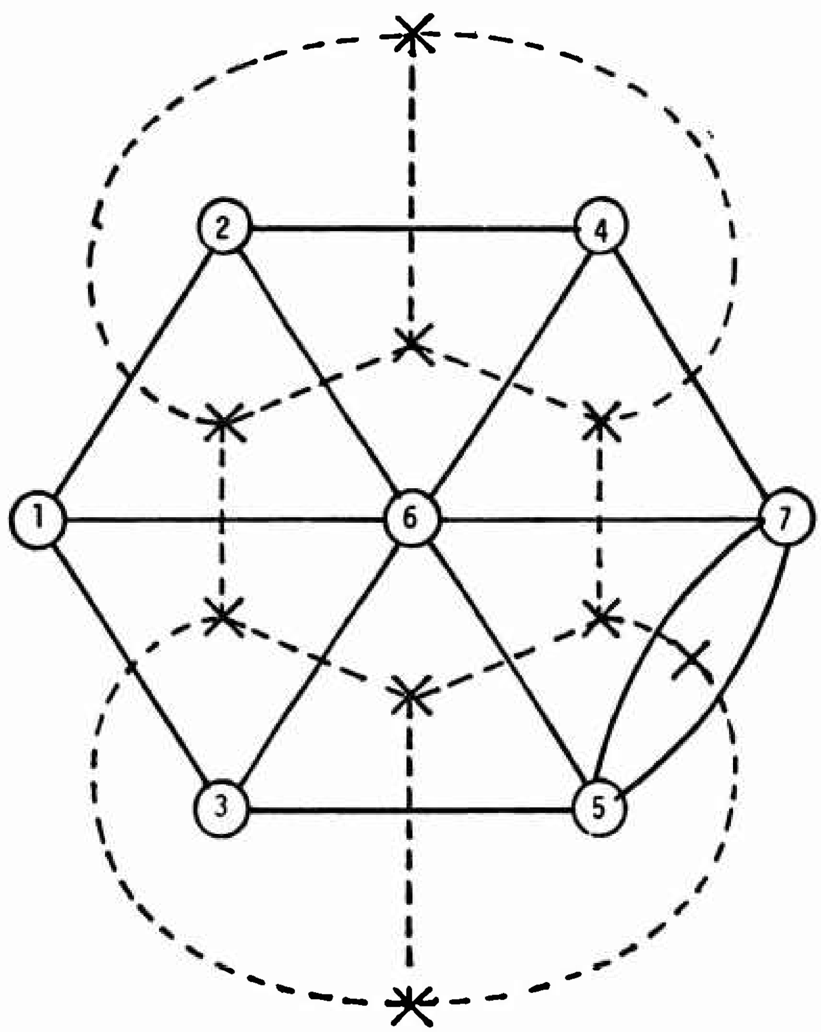

Figure 3. Construction of the Dual

Figure 4 shows the topological dual after the data for each arc has been transferred from the primal. In the dual u_{ij} and l_{ij} represent bounds on arc length.

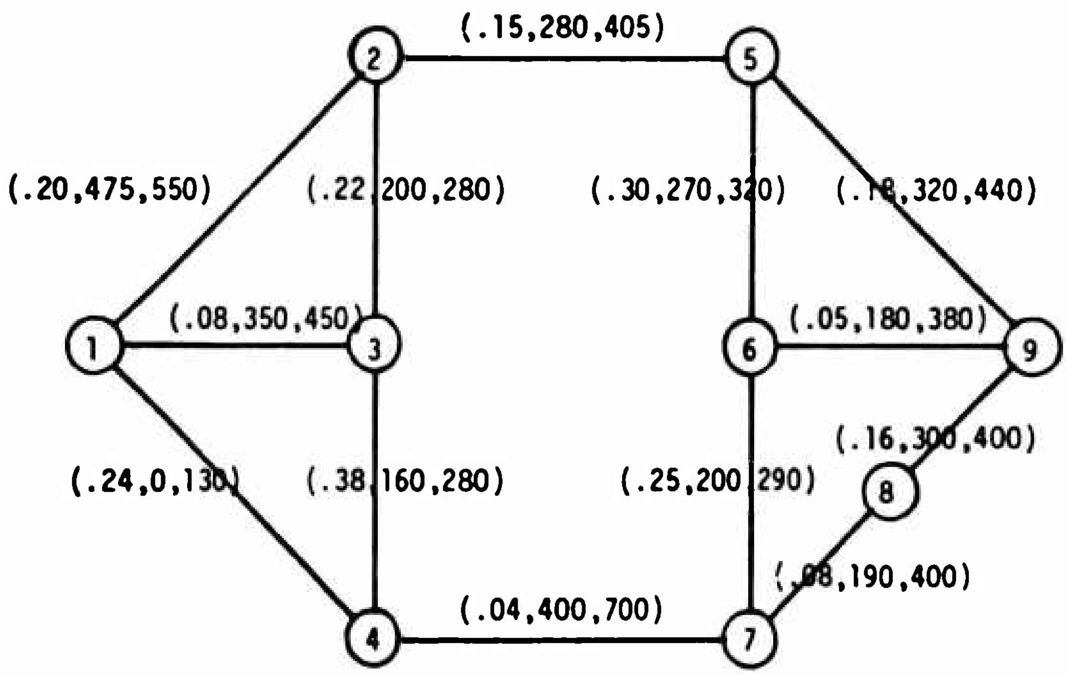

Figure 4. The Topological Dual

To simplify notation, paths through the dual will be designated by the nodes over which they pass. The shortest path through the dual before interdiction is 1,2,5,9 with a length of 1395. This gives an upper bound on network capacity. Table I lists all loopless paths through the dual in order of length after unlimited interdiction. It should be noted that the length of path number 11, the 11^{th} shortest path after unlimited

TABLE I. DUAL PATHS

PATH	NODES	
1	1,4,7,6,9	780
2	1,4,7,8,9	890
3	1,4,3,2,5,9	960
4	1,2,[illegible],9	1075
5	1,4,3,2,5,6,9	1090
6	1,3,2,5,9	1150
7	1,4,7,6,5,9	1190
8	1,2,5,6,9	1205
9	1,3,2,5,6,9	1280
10	1,3,4,7,6,9	1290
11	1,3,4,7,8,9	1400
12	1,4,3,2,5,6,7,8,9	1600
13	1,2,3,4,7,6,9	1615
14	1,3,4,7,6,5,9	1700
15	1,2,5,6,7,8,9	1715
16	1,2,3,4,7,8,9	1725
17	1,3,2,5,6,7,8,9	1790
18	1,2,3,4,7,6,5,9	2025

interdiction, exceeds the upper bound on network capacity. Therefore the cut sets comprising set S correspond to paths 1 through 10.

It is assumed for purposes of this example that there are 100 aircraft available for assignment at a cost of 30,000 dollars for each aircraft assigned. It is further assumed that the benefit derived from a reduction of one ton per day in network capacity is 7,500 dollars.

The dynamic program for each S_i that is constraining including a sensitivity analysis on K^* is contained in the computer output. A graph of the resulting network capacity is given by Figure 5. For K^* in the range 1 through 25 cut set 4 determines network capacity, for K^* in the range 26 through 57 cut set 3 is constraining, and for K^* from 58 to 100 cut set 1 is minimal.

From the given values of C and D, 30,000 and 7,500 respectively, the points of interest are those at which $\delta(K^*)$ becomes $\leq C/D = 4$. This occurs for the first time when $K^* = 42$. Therefore, $K_1^* = 41$. At $K^* = 58$ $\delta(K^*)$ again exceeds 4 so it is necessary to search for another point where $\delta(K^*) \leq 4$. This next occurs at $K^* = 62$ and K_2^* is 61. Since $\delta(K^*)$ does not exceed 4 for any $K^* > 62$, $K_n^* = K_2^*$.

It is obvious that the benefit to get to K_1^* exceeded the cost since K_1^* was the first point at which the allocation of another aircraft did not produce benefits exceeding costs. However, it is not quite as obvious when the decision is made whether or not to allocate K_2^* aircraft. The benefit to get from K_1^* to K_2^* is equal to the incremental capacity reduction multiplied by D.

$$\text{Benefit} = 62.23 \times 7,500$$

$$= 466,725.$$

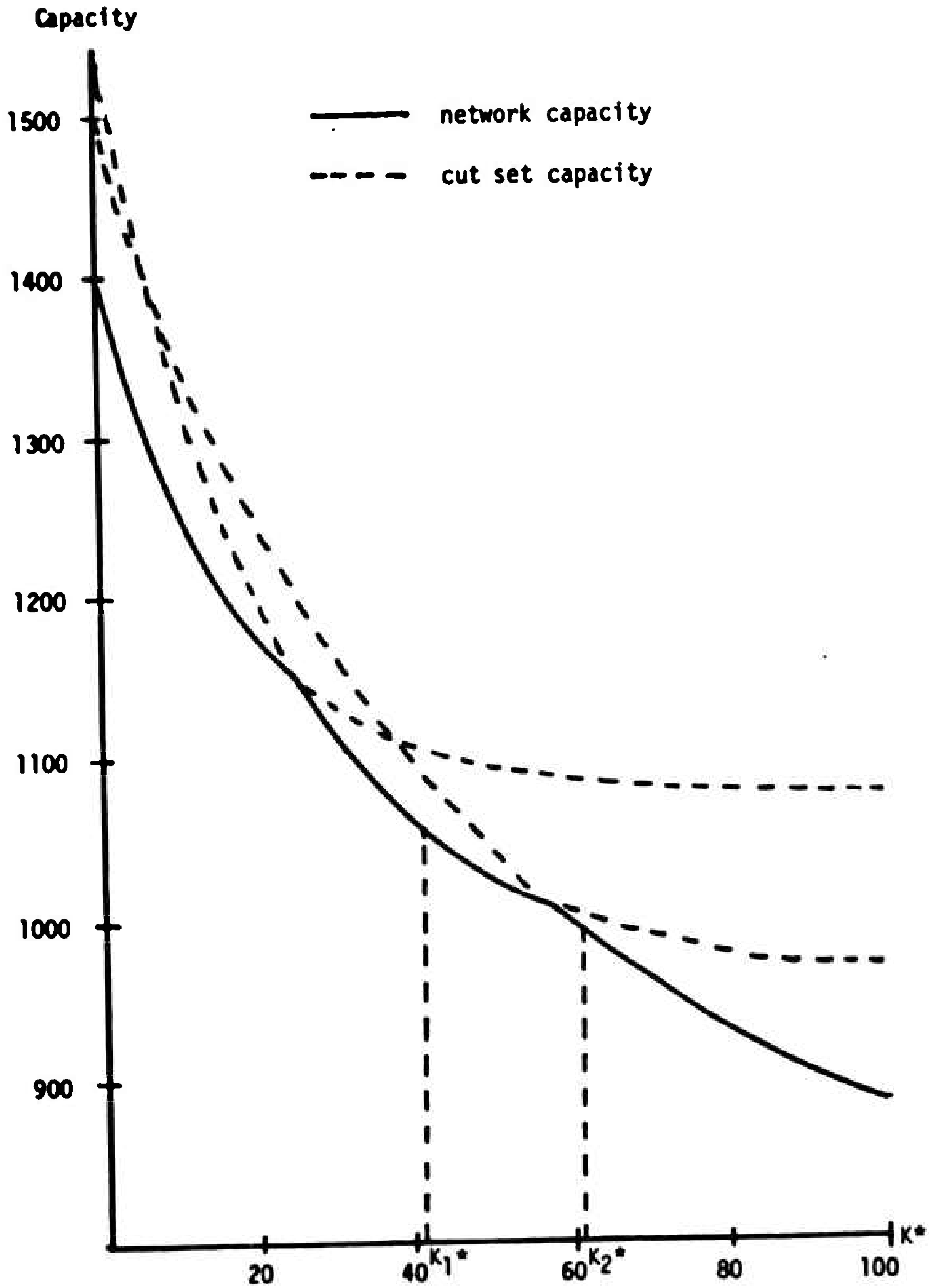

Figure 5. Example Network Capacity

This benefit is compared to the cost of allocating $K_2^* - K_1^*$ aircraft.

$$\text{Cost} = 20 \times 30{,}000$$

$$= 600{,}000.$$

Thus the benefit is outweighed by the cost. Since there is no allocation of aircraft greater than K_2^* that will result in benefits exceeding costs, it may be concluded that $K^* = 41$ represents the optimal number of aircraft to assign to the airstrike. At this level of interdiction the network capacity will be 1056.06 tons per day. This is a reduction of 338.94 tons per day with a resulting benefit of 2,542,050 dollars. The cost of this reduction is 1,230,000 dollars. The cut set that will be attacked is the cut set corresponding to path number three which contains the following primal arcs : (4,7); (4,6); (2,6); (1,6); and (1,3). These arcs correspond to dynamic programming stages 1,2,3,4, and 5 respectively. Looking at the dynamic programming stages the optimal allocation of aircraft is: $k_{4,7} = 9$; $k_{4,6} = 6$; $k_{2,6} = 7$; $k_{1,6} = 10$; and $k_{1,3} = 9$. This completes the solution.

IV. DISCUSSION

A. PROPERTIES OF THE SOLUTION TECHNIQUE

The dynamic programming approach taken to the problem guarantees that the solution found will be a global minimum over the feasible region. The integer constraints pose no problem. In fact, the integer restriction limits the number of values the decision variables may assume and allows an exact solution to be found. Dynamic programming also provides a built in capability for sensitivity analysis.

The convexity of the damage function allowed the use of Fibonacci search within the dynamic program resulting in a tremendous savings in the number of separate calculations made in each dynamic program. With K equal to 100 the reduction was on the order of 10^{-1} times the number of calculations needed for exhaustive search. Larger values of K will produce savings of an even greater magnitude. The execution time required for the sample problem was 15.06 seconds on an IBM 360/67. Utilizing Fibonacci search it was found that the increase in execution time for larger values of K was approximately linear. Execution time was also roughly linear with respect to the total number of dynamic programming stages required (45 in the sample problem). From the above observations the amount of computer time required for larger problems may be predicted. For example, a problem with 15 cut sets in S averaging 6 arcs per cut set would require 90 dynamic programming stages. If 200 aircraft were available a reasonable estimate would be that this problem would take approximately 4 times as long to solve as the sample problem.

A further reduction in the number of calculations required may be achieved with a coarse grid. Aircraft can be allocated in packages of five and the constraining cut sets determined. These cut sets can then have aircraft reallocated one at a time and the optimal solution found as before. This approach can not guarantee that the correct constraining cut sets will be selected, but if they are the solution will be optimal.

Dynamic programming allows some generalizations to be made in the problem. To begin with, since additive stage returns are always decomposable, the technique places no restrictions on the damage functions. The negative exponential damage function used in this paper has intuitive appeal since it does exhibit diminishing marginal returns. This function also contributes to computational efficiency since its convexity allowed the use of Fibonacci search. However, if actual interdiction data suggests damage functions of another form, the problem can still be solved with somewhat greater expenditures of computer time.

Another generalization suggested by dynamic programming is to consider the allocation of two types of aircraft. In this case a damage function of the form

$$m_{ij}(k_{ij},h_{ij}) = l_{ij} + w_{ij} \exp(-b_{ij}k_{ij} - a_{ij}h_{ij})$$

might be assumed with k_{ij}, l_{ij}, and w_{ij} defined as before, h_{ij} representing the number of aircraft of the second type assigned to arc (i,j), and a_{ij} denoting the vulnerability parameter corresponding to the second type of aircraft. Dynamic programming may again be used to solve the problem, but two state and two decision variables are required.

Although the new damage function preserves convexity, in this case the series of minimizations is of functions of two variables and Fibonacci search is not applicable. A minimization problem was run for a hypothetical cut set containing five arcs. The execution time required for solution was 5.63 seconds when 10 aircraft of two types were available; with 19 aircraft of each type, the time required was 32.79 seconds; and when 25 of each type aircraft were available, over a minute of computer time was used. To deal with even relatively small networks the computer time requirements would become prohibitive if it were necessary to consider larger aircraft availabilities. To assign three types of aircraft, dynamic programming would require three state and three decision variables and the technique would be impractical even for small problems.

Another application of the dynamic programming approach is in a modification of Nugent's algorithm [3]. This modification will provide integer solutions. Nugent presented a method of finding non-integer allocations of resources that would minimize network capacity subject to $\sum_{(i,j)\varepsilon S_i} k_{ij} \leq K$ and $k_{ij} \geq 0$. As previously discussed, the objective function in this problem is convex with respect to k_{ij}. In Nugent's formulation the feasible region defined by the constraints is also convex. Therefore, for any particular cut set the problem is a convex non-linear program and Kuhn-Tucker theory provides conditions that are both necessary and sufficient for a global minimum. Nugent solves these Kuhn-Tucker conditions and using an upper bounding technique arrives at the cut set that will be minimal after optimum interdiction.

In the modification the set S and the upper bound on network capacity are found as before. The Kuhn-Tucker conditions are then solved to find non-integer constrained solutions that minimize cut set capacity for each element of S. The minimum of these solutions represents the optimal solution without integer constraints. When integer constraints are added this minimum represents a lower bound on network capacity. The cut set with the minimal non-integer solution is deleted from S and becomes the subject of a dynamic program to find an integer solution. If this integer solution is less than the non-integer solutions corresponding to the remaining elements of S it is optimal. If it is greater than some or all of the elements of S it represents a new, smaller upper bound on network capacity. Any elements of S with non-integer solutions greater than this new upper bound are deleted from S. From the remaining elements of S the cut set with the smallest non-integer solution is selected from S. Again dynamic programming used to find a new integer solution. The new integer solution is compared with the old integer solution and the minimum is called the current integer solution. The current integer solution is then compared to the remaining non-integer solutions and the process is repeated. This iterative procedure is continued until either S is the null set or until the current integer solution is less than or equal the non-integer solutions corresponding to all of the remaining elements of S. In either case the current integer solution represents the optimal solution to the integer constrained problem.

In general, if the number of aircraft to be allocated to the air-strike is known, this modification is more efficient that using dynamic programming on every element of S to solve the minimization problem. In solving the example problem from Nugent's paper it was necessary to run only one dynamic program and with exponential damage functions the Kuhn-Tucker conditions are easy to solve relative to solving a dynamic program. However, this modification does not lend itself to the sensitivity analysis on K that is necessary when the number of aircraft to be assigned to the strike is taken to be a decision variable.

As already mentioned, there are limitations on the technique presented. One difficulty that has not yet been discussed is in the measurement of the costs and benefits of aircraft assignment. In this paper the problem was ignored and constant dollar values of C and D were selected arbitrarily. This problem is important since the selection of C and D determines how many aircraft will be assigned to the strike. If D had been taken to be [illegible]000 dollars per ton of flow reduced vice 7,500 and the rest of the problem remained unchanged, the decision would have been made to allocate 77 aircraft in a strike against cut set one resulting in a network capacity of 938.19 tons per day. On the other hand, if D was less than 1,519 dollars per ton of flow reduced the solution would be to make no attack against the network.

B. RECOMMENDATIONS FOR FURTHER STUDY

The possibility of deriving damage functions from actual interdiction data was mentioned earlier. If the method of this paper were to be put to use in solving a real-world interdiction problem some verification of the damage function would be essential. However, due to the sensitivity of the solution to both costs and benefits, the measurement problem associated with costs and benefits should receive at least as much attention as the damage function.

Another possibility for further study would be the utilization of the model described in this paper to represent real-world problems other than aircraft interdiction. One obvious example might be the problem of allocating resources to the improvement of a highway system. In this example it would probably be relatively easy to get data from which to derive improvement functions, but the measurement of costs and benefits would be as difficult as before.

The model presented could be refined by assigning different values to capacity reduction in the various arcs of the network. The objective then would be to minimize the maximum value of flow possible in the network rather than to minimize network capacity. The solution technique presented could still be used. A further refinement might be to consider not only arc vulnerability, but also the repair capability of the opponent. This would require capacity reduction to be taken as a function of time as well as aircraft allocation and would make the analysis of the model more difficult. Many other refinements could be made in order to make the model more representative of the real world, but in general the increased realism gained would be at the expense of increased computational effort.

V. SUMMARY

A solution procedure has been developed for the problem of determining the optimal allocation of aircraft in planning an airstrike against a transportation network. The damage function for arcs in the network is assumed to have a negative exponential form. To make use of the procedure it is necessary to have available the following information: the upper and lower bounds on the capacity of each arc, the vulnerability parameter for each arc, the number of aircraft available for assignment to the airstrike, the cost of assigning an aircraft to the strike, and the benefit resulting from network capacity reduction.

In the solution procedure every cut set that is designated a candidate for attack is the subject of a dynamic program. A sensitivity analysis is performed on the number of aircraft to be assigned and this gives the network capacity after optimal interdiction as a function of the number of aircraft assigned to the strike. A cost benefit analysis is then made to determine the largest number of aircraft that can be assigned before costs of further allocation begin to outweigh the benefits resulting from that allocation.

At the end of the procedure the solution consists of the following: the number of aircraft to assign to the airstrike, the cut set that will be attacked, the number of aircraft to allocate to each arc of the cut set chosen, and the capacity of the network after this assignment of aircraft.

COMPUTER OUTPUT

DP SOLNS CUT SET 1

ACFT AVAIL	CAPACITY	CHANGE IN CAPACITY
0	1500.0000	0.0
1	1472.2615	-27.7385
2	1450.4417	-21.8198
3	1430.5337	-19.9080
4	1413.3696	-17.1641
5	1397.8652	-15.5044
6	1384.3635	-13.5017
7	1372.2888	-12.0747
8	1360.5256	-11.7632
9	1349.2236	-11.3020
10	1338.3647	-10.8589
11	1327.7441	-10.6206
12	1317.3113	-10.4329
13	1307.2874	-10.0239
14	1297.5332	-9.7542
15	1287.9023	-9.6309
16	1278.4985	-9.4038
17	1269.2200	-9.2786
18	1259.9668	-9.2532
19	1251.0762	-8.8906
20	1242.2502	-8.8259
21	1233.7085	-8.5417
22	1225.3132	-8.3953
23	1216.9587	-8.3545
24	1208.7517	-8.2070
25	1200.7656	-7.9861
26	1192.8806	-7.8850
27	1185.2842	-7.5964
28	1177.7083	-7.5759
29	1170.3845	-7.3237
30	1163.1057	-7.2788
31	1155.8796	-7.2261
32	1148.8862	-6.9934
33	1142.0127	-6.8735
34	1135.2935	-6.7192
35	1128.7214	-6.5720
36	1122.1831	-6.5383
37	1115.7273	-6.4558
38	1109.5078	-6.2195
39	1103.3052	-6.2026
40	1097.3457	-5.9595
41	1091.4297	-5.9160
42	1085.7039	-5.7258
43	1080.0000	-5.7039
44	1074.3726	-5.6274
45	1068.8713	-5.5012
46	1063.5181	-5.3533
47	1058.2324	-5.2856
48	1053.0627	-5.1697
49	1047.9707	-5.0920

ACFT AVAIL	CAPACITY	CHANGE IN CAPACITY
50	1042.8923	-5.0784
51	1038.0132	-4.8792
52	1033.1694	-4.8435
53	1028.4817	-4.6879
54	1023.8743	-4.6075
55	1019.3701	-4.5040
56	1014.9280	-4.4421
57	1010.5452	-4.3828
58	1006.2178	-4.3274
59	1002.0488	-4.1691
60	997.8911	-4.1577
61	993.8245	-4.0666
62	989.8296	-3.9947
63	985.8640	-3.9657
64	982.0259	-3.8381
65	978.2537	-3.7723
66	974.5659	-3.6876
67	970.9778	-3.5883
68	967.4346	-3.5430
69	963.9751	-3.4595
70	960.5618	-3.4133
71	957.1577	-3.4041
72	953.8872	-3.2706
73	950.6404	-3.2469
74	947.4414	-3.1989
75	944.2991	-3.1423
76	941.2104	-3.0885
77	938.1914	-3.0191
78	935.2534	-2.9379
79	932.3528	-2.9008
80	929.5581	-2.7946
81	926.7710	-2.7870
82	924.0769	-2.6942
83	921.3992	-2.6777
84	918.7407	-2.6583
85	916.1680	-2.5727
86	913.6394	-2.5287
87	911.1230	-2.5164
88	908.6511	-2.4719
89	906.2458	-2.4053
90	903.8708	-2.3749
91	901.5830	-2.2880
92	899.3010	-2.2818
93	897.1086	-2.1924
94	894.9324	-2.1765
95	892.8259	-2.1064
96	890.7275	-2.0983
97	888.6572	-2.0703
98	886.6335	-2.0238
99	884.6541	-1.9794
100	882.6848	-1.9693

STAGE NUMBER 1

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	1	2	2
3	3	4	4	5	5
6	6	7	7	8	8
9	9	10	10	11	11
12	12	13	13	14	14
15	15	16	16	17	17
18	18	19	19	20	20
21	21	22	22	23	23
24	24	25	25	26	26
27	27	28	28	29	29
30	30	31	31	32	32
33	33	34	34	35	35
36	36	37	37	38	38
39	39	40	40	41	41
42	42	43	43	44	44
45	45	46	46	47	47
48	48	49	49	50	50
51	51	52	52	53	53
54	54	55	55	56	56
57	57	58	58	59	59
60	60	61	61	62	62
63	63	64	64	65	65
66	66	67	67	68	68
69	69	70	70	71	71
72	72	73	73	74	74
75	75	76	76	77	77
78	78	79	79	80	80
81	81	82	82	83	83
84	84	85	85	86	86
87	87	88	88	89	89
90	90	91	91	92	92
93	93	94	94	95	95
96	96	97	97	98	98
99	99	100	100		

STAGE NUMBER 2

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	0	2	0
3	0	4	0	5	1
6	2	7	3	8	3
9	4	10	5	11	6
12	7	13	8	14	9
15	9	16	10	17	11
18	12	19	13	20	14
21	15	22	15	23	16
24	17	25	18	26	19
27	20	28	21	29	21
30	22	31	23	32	24
33	25	34	26	35	27
36	27	37	28	38	29
39	30	40	31	41	32
42	33	43	33	44	34
45	35	46	36	47	37
48	38	49	39	50	39
51	40	52	41	53	42
54	43	55	44	56	45
57	45	58	46	59	47
60	48	61	49	62	50
63	51	64	51	65	52
66	53	67	54	68	55
69	56	70	57	71	57
72	58	73	59	74	60
75	61	76	62	77	63
78	63	79	64	80	65
81	66	82	67	83	68
84	69	85	69	86	70
87	71	88	72	89	73
90	74	91	75	92	75
93	76	94	77	95	78
96	79	97	80	98	81
99	81	100	82		

STAGE NUMBER 3

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	0	2	0
3	1	4	1	5	2
6	2	7	3	8	3
9	3	10	3	11	3
12	3	13	3	14	3
15	4	16	4	17	4
18	4	19	4	20	4
21	4	22	4	23	5
24	5	25	5	26	5
27	5	28	5	29	5
30	5	31	5	32	6
33	6	34	6	35	6
36	6	37	6	38	6
39	6	40	7	41	7
42	7	43	7	44	7
45	7	46	7	47	7
48	8	49	8	50	8
51	8	52	8	53	8
54	8	55	8	56	9
57	9	58	9	59	9
60	9	61	9	62	9
63	9	64	9	65	10
66	10	67	10	68	10
69	10	70	10	71	10
72	10	73	11	74	11
75	11	76	11	77	11
78	11	79	11	80	11
81	12	82	12	83	12
84	12	85	12	86	12
87	12	88	12	89	13
90	13	91	13	92	13
93	13	94	13	95	13
96	13	97	13	98	14
99	14	100	14		

STAGE NUMBER 4

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	0	2	0
3	0	4	0	5	0
6	0	7	0	8	0
9	0	10	0	11	0
12	0	13	0	14	1
15	1	16	1	17	2
18	2	19	2	20	3
21	3	22	4	23	4
24	4	25	5	26	5
27	6	28	6	29	6
30	6	31	7	32	7
33	8	34	8	35	8
36	9	37	9	38	10
39	10	40	10	41	11
42	11	43	11	44	12
45	12	46	13	47	13
48	13	49	14	50	14
51	14	52	15	53	15
54	16	55	16	56	16
57	17	58	17	59	18
60	18	61	18	62	18
63	19	64	19	65	20
66	20	67	21	68	21
69	21	70	22	71	22
72	22	73	23	74	23
75	23	76	24	77	24
78	25	79	25	80	26
81	26	82	26	83	26
84	27	85	27	86	28
87	28	88	28	89	29
90	29	91	30	92	30
93	30	94	31	95	31
96	31	97	32	98	32
99	32	100	33		

DP SOLNS CUT SET 3

ACFT AVAIL	CAPACITY	CHANGE IN CAPACITY
0	1535.0000	0.0
1	1497.0632	-37.9368
2	1469.3250	-27.7383
3	1443.3813	-25.9436
4	1421.5618	-21.8196
5	1401.7942	-19.7676
6	1384.0525	-17.7417
7	1366.6409	-17.4116
8	1349.4768	-17.1641
9	1332.9653	-16.5115
10	1317.1670	-15.7983
11	1302.1807	-14.9863
12	1288.3894	-13.7913
13	1274.8879	-13.5015
14	1261.9890	-12.8989
15	1249.3105	-12.6785
16	1237.1775	-12.1331
17	1225.6582	-11.5193
18	1214.5559	-11.1021
19	1203.9353	-10.6208
20	1193.7605	-10.1748
21	1184.1384	-9.6219
22	1174.5828	-9.5556
23	1166.2283	-8.3547
24	1157.9309	-8.2972
25	1149.7063	-8.2246
26	1141.5410	-8.1655
27	1133.5042	-8.0369
28	1126.4250	-7.0790
29	1119.7122	-6.7130
30	1113.1401	-6.5720
31	1106.5872	-6.5529
32	1100.4941	-6.0930
33	1094.8201	-5.6741
34	1089.2129	-5.6072
35	1083.9541	-5.2589
36	1078.7097	-5.2442
37	1073.5400	-5.1697
38	1068.8567	-4.6835
39	1064.3428	-4.5138
40	1060.1226	-4.2203
41	1056.0559	-4.0666
42	1052.1440	-3.9120
43	1048.2588	-3.8850
44	1044.3787	-3.8803
45	1040.9917	-3.3869
46	1037.6477	-3.3439
47	1034.3801	-3.2675
48	1031.1814	-3.1989
49	1028.3032	-2.8781

ACFT AVAIL	CAPACITY	CHANGE IN CAPACITY
50	1025.5740	-2.7293
51	1022.8560	-2.7181
52	1020.2021	-2.6536
53	1017.6858	-2.5164
54	1015.2087	-2.4772
55	1012.9290	-2.2797
56	1010.7476	-2.1813
57	1008.6155	-2.1321
58	1006.6360	-1.9794
59	1004.7319	-1.9042
60	1002.8967	-1.8352
61	1001.0820	-1.8147
62	999.3315	-1.7505
63	997.7410	-1.5905
64	996.1616	-1.5795
65	994.6045	-1.5571
66	993.1997	-1.4048
67	991.8401	-1.3595
68	990.5115	-1.3285
69	989.2705	-1.2410
70	988.0457	-1.2249
71	986.8755	-1.1702
72	985.7483	-1.1274
73	984.6384	-1.1096
74	983.6313	-1.0072
75	982.6680	-0.9635
76	981.7410	-0.9269
77	980.8362	-0.9048
78	979.9695	-0.8669
79	979.1208	-0.8487
80	978.3464	-0.7742
81	977.5886	-0.7579
82	976.8425	-0.7461
83	976.1165	-0.7261
84	975.4697	-0.6466
85	974.8276	-0.6422
86	974.2314	-0.5962
87	973.6487	-0.5827
88	973.0684	-0.5804
89	972.5156	-0.5527
90	971.9753	-0.5401
91	971.4998	-0.4757
92	971.0308	-0.4690
93	970.5630	-0.4676
94	970.1121	-0.4511
95	969.7024	-0.4095
96	969.3057	-0.3969
97	968.9287	-0.3768
98	968.5535	-0.3753
99	968.1846	-0.3689
100	967.8320	-0.3524

STAGE NUMBER 1

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	1	2	2
3	3	4	4	5	5
6	6	7	7	8	8
9	9	10	10	11	11
12	12	13	13	14	14
15	15	16	16	17	17
18	18	19	19	20	20
21	21	22	22	23	23
24	24	25	25	26	26
27	27	28	28	29	29
30	30	31	31	32	32
33	33	34	34	35	35
36	36	37	37	38	38
39	39	40	40	41	41
42	42	43	43	44	44
45	45	46	46	47	47
48	48	49	49	50	50
51	51	52	52	53	53
54	54	55	55	56	56
57	57	58	58	59	59
60	60	61	61	62	62
63	63	64	64	65	65
66	66	67	67	68	68
69	69	70	70	71	71
72	72	73	73	74	74
75	75	76	76	77	77
78	78	79	79	80	80
81	81	82	82	83	83
84	84	85	85	86	86
87	87	88	88	89	89
90	90	91	91	92	92
93	93	94	94	95	95
96	96	97	97	98	98
99	99	100	100		

STAGE NUMBER 2

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	1	2	1
3	2	4	2	5	3
6	3	7	3	8	4
9	4	10	4	11	5
12	5	13	6	14	6
15	6	16	7	17	7
18	8	19	8	20	8
21	9	22	9	23	10
24	10	25	10	26	11
27	11	28	11	29	12
30	12	31	13	32	13
33	13	34	14	35	14
36	15	37	15	38	15
39	16	40	16	41	16
42	17	43	17	44	18
45	18	46	18	47	19
48	19	49	20	50	20
51	20	52	21	53	21
54	22	55	22	56	22
57	23	58	23	59	23
60	24	61	24	62	25
63	25	64	25	65	26
66	26	67	27	68	27
69	27	70	28	71	28
72	28	73	29	74	29
75	30	76	30	77	30
78	31	79	31	80	32
81	32	82	32	83	33
84	33	85	34	86	34
87	34	88	35	89	35
90	35	91	36	92	36
93	37	94	37	95	37
96	38	97	38	98	39
99	39	100	39		

STAGE NUMBER 3

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	0	2	0
3	0	4	0	5	0
6	0	7	1	8	1
9	2	10	2	11	2
12	3	13	3	14	3
15	4	16	4	17	5
18	5	19	6	20	6
21	7	22	7	23	7
24	8	25	8	26	9
27	9	28	9	29	10
30	10	31	10	32	11
33	11	34	12	35	12
36	12	37	13	38	13
39	14	40	14	41	14
42	15	43	15	44	16
45	16	46	16	47	17
48	17	49	18	50	18
51	19	52	19	53	19
54	20	55	20	56	21
57	21	58	21	59	22
60	22	61	22	62	23
63	23	64	24	65	24
66	24	67	25	68	25
69	26	70	26	71	26
72	27	73	27	74	28
75	28	76	28	77	29
78	29	79	30	80	30
81	31	82	31	83	31
84	32	85	32	86	33
87	33	88	33	89	34
90	34	91	35	92	35
93	35	94	36	95	36
96	36	97	37	98	37
99	38	100	38		

STAGE NUMBER 4

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	0	2	0
3	0	4	0	5	0
6	1	7	1	8	1
9	2	10	2	11	3
12	3	13	3	14	4
15	4	16	4	17	5
18	5	19	5	20	6
21	6	22	7	23	7
24	7	25	8	26	8
27	8	28	9	29	9
30	10	31	10	32	10
33	11	34	11	35	11
36	12	37	12	38	13
39	13	40	13	41	13
42	14	43	14	44	15
45	15	46	16	47	16
48	16	49	17	50	17
51	17	52	18	53	18
54	18	55	19	56	19
57	20	58	20	59	20
60	21	61	21	62	21
63	22	64	22	65	23
66	23	67	23	68	23
69	24	70	25	71	25
72	25	73	26	74	26
75	26	76	26	77	27
78	28	79	28	80	28
81	28	82	29	83	29
84	29	85	30	86	30
87	31	88	31	89	31
90	32	91	32	92	33
93	33	94	33	95	33
96	34	97	34	98	35
99	35	100	36		

STAGE NUMBER 5

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	0	2	0
3	0	4	0	5	1
6	1	7	1	8	1
9	2	10	2	11	2
12	3	13	3	14	3
15	3	16	3	17	4
18	4	19	4	20	4
21	5	22	5	23	5
24	5	25	5	26	5
27	6	28	6	29	7
30	7	31	7	32	7
33	7	34	8	35	8
36	8	37	8	38	9
39	9	40	9	41	9
42	10	43	10	44	10
45	10	46	10	47	11
48	11	49	11	50	12
51	12	52	12	53	12
54	12	55	13	56	13
57	13	58	13	59	14
60	14	61	14	62	14
63	15	64	15	65	15
66	15	67	15	68	16
69	16	70	16	71	16
72	16	73	17	74	17
75	17	76	18	77	18
78	18	79	18	80	19
81	19	82	19	83	19
84	20	85	20	86	20
87	20	88	20	89	20
90	21	91	21	92	21
93	21	94	22	95	22
96	22	97	23	98	23
99	23	100	23		

DP SOLNS CUT SET 4

ACFT AVAIL	CAPACITY	CHANGE IN CAPACITY
0	1395.0000	0.0
1	1375.2324	-19.7676
2	1357.8208	-17.4116
3	1341.3096	-16.5112
4	1326.3232	-14.9861
5	1312.5320	-13.7914
6	1298.9368	-13.5952
7	1286.0381	-12.8988
8	1274.5186	-11.5195
9	1263.3877	-11.1308
10	1252.2856	-11.1021
11	1242.6638	-9.6219
12	1233.1082	-9.5556
13	1223.9949	-9.1131
14	1215.7703	-8.2246
15	1207.7334	-8.0369
16	1200.2722	-7.4612
17	1193.1934	-7.0790
18	1186.4802	-6.7130
19	1180.3716	-6.1087
20	1174.2786	-6.0930
21	1168.6714	-5.6072
22	1163.4272	5.2442
23	1158.4258	-5.0014
24	1153.7424	-4.6835
25	1149.2285	-4.5138
26	1145.1338	-4.0948
27	1141.2219	-3.9120
28	1137.3367	-3.8850
29	1133.9841	-3.3525
30	1130.6404	-3.3439
31	1127.3728	-3.2675
32	1124.4946	-2.8781
33	1121.7500	-2.7448
34	1119.0205	-2.7293
35	1116.5435	-2.4772
36	1114.2637	-2.2797
37	1112.0164	-2.2473
38	1109.8843	-2.1322
39	1107.9800	-1.9042
40	1106.1401	-1.8399
41	1104.3049	-1.8352
42	1102.7146	-1.5905
43	1101.1350	-1.5795
44	1099.6287	-1.5064
45	1098.2690	-1.3595
46	1096.9407	-1.3285
47	1095.7073	-1.2333
48	1094.5371	-1.1702
49	1093.4275	-1.1096

ACFT AVAIL	CAPACITY	CHANGE IN CAPACITY
50	1092.4177	-1.0098
51	1091.4106	-1.0072
52	1090.4836	-0.9269
53	1089.6169	-0.8669
54	1088.7900	-0.8267
55	1088.0159	-0.7742
56	1087.2698	-0.7461
57	1086.5930	-0.6769
58	1085.9463	-0.6466
59	1085.3042	-0.6422
60	1084.7500	-0.5542
61	1084.1973	-0.5527
62	1083.6570	-0.5401
63	1083.1814	-0.4757
64	1082.7275	-0.4537
65	1082.2764	-0.4511
66	1081.8669	-0.4095
67	1081.4902	-0.3768
68	1081.1187	-0.3715
69	1080.7661	-0.3524
70	1080.4514	-0.3148
71	1080.1472	-0.3041
72	1079.8440	-0.3033
73	1079.5811	-0.2629
74	1079.3201	-0.2611
75	1079.0710	-0.2490
76	1078.8462	-0.2247
77	1078.6267	-0.2196
78	1078.4229	-0.2039
79	1078.2292	-0.1934
80	1078.0459	-0.1834
81	1077.8789	-0.1669
82	1077.7126	-0.1665
83	1077.5593	-0.1532
84	1077.4160	-0.1433
85	1077.2793	-0.1367
86	1077.1514	-0.1280
87	1077.0281	-0.1233
88	1076.9163	-0.1119
89	1076.8093	-0.1069
90	1076.7031	-0.1062
91	1076.6116	-0.0916
92	1076.5203	-0.0914
93	1076.4309	-0.0893
94	1076.3523	-0.0786
95	1076.2773	-0.0750
96	1076.2026	-0.0746
97	1076.1350	-0.0677
98	1076.0728	-0.0623
99	1076.0112	-0.0614
100	1075.9531	-0.0583

STAGE NUMBER 1

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	1	2	2
3	3	4	4	5	5
6	6	7	7	8	8
9	9	10	10	11	11
12	12	13	13	14	14
15	15	16	16	17	17
18	18	19	19	20	20
21	21	22	22	23	23
24	24	25	25	26	26
27	27	28	28	29	29
30	30	31	31	32	32
33	33	34	34	35	35
36	36	37	37	38	38
39	39	40	40	41	41
42	42	43	43	44	44
45	45	46	46	47	47
48	48	49	49	50	50
51	51	52	52	53	53
54	54	55	55	56	56
57	57	58	58	59	59
60	60	61	61	62	62
63	63	64	64	65	65
66	66	67	67	68	68
69	69	70	70	71	71
72	72	73	73	74	74
75	75	76	76	77	77
78	78	79	79	80	80
81	81	82	82	83	83
84	84	85	85	86	86
87	87	88	88	89	89
90	90	91	91	92	92
93	93	94	94	95	95
96	96	97	97	98	98
99	99	100	100		

STAGE NUMBER 2

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	1	2	2
3	2	4	3	5	3
6	4	7	5	8	5
9	6	10	6	11	7
12	7	13	8	14	9
15	9	16	10	17	10
18	11	19	11	20	12
21	13	22	13	23	14
24	14	25	15	26	15
27	16	28	17	29	17
30	18	31	18	32	19
33	19	34	20	35	21
36	21	37	22	38	22
39	23	40	23	41	24
42	25	43	25	44	26
45	26	46	27	47	27
48	28	49	29	50	29
51	30	52	30	53	31
54	31	55	32	56	33
57	33	58	34	59	34
60	35	61	35	62	36
63	37	64	37	65	38
66	38	67	39	68	39
69	40	70	41	71	41
72	42	73	42	74	43
75	43	76	44	77	45
78	45	79	46	80	46
81	47	82	47	83	48
84	49	85	49	86	50
87	50	88	51	89	51
90	52	91	53	92	53
93	54	94	54	95	55
96	55	97	56	98	57
99	57	100	58		

STAGE NUMBER 3

ISTATE	IDEC	ISTATE	IDEC	ISTATE	IDEC
0	0	1	1	2	1
3	2	4	2	5	3
6	3	7	3	8	4
9	4	10	4	11	5
12	5	13	5	14	5
15	6	16	6	17	6
18	7	19	7	20	7
21	8	22	8	23	8
24	9	25	9	26	9
27	10	28	10	29	10
30	10	31	11	32	11
33	11	34	12	35	12
36	13	37	13	38	13
39	14	40	14	41	14
42	15	43	15	44	15
45	15	46	16	47	16
48	16	49	17	50	17
51	17	52	18	53	18
54	18	55	19	56	19
57	19	58	20	59	20
60	20	61	20	62	21
63	21	64	21	65	22
66	22	67	23	68	23
69	23	70	24	71	24
72	24	73	25	74	25
75	25	76	25	77	26
78	26	79	26	80	27
81	27	82	27	83	28
84	28	85	28	86	29
87	29	88	29	89	30
90	30	91	30	92	30
93	31	94	31	95	31
96	32	97	32	98	33
99	33	100	33		

```
C                        COMPUTER PROGRAM
C                        -------- -------
C
C          THIS PROGRAM IS DESIGNED TO FIND THE MINIMUM OF A
C          SEQUENCE OF FUNCTIONS.  EACH FUNCTION IN THE SEQUENCE
C          IS A SUM OF NEGATIVE EXPONENTIAL FUNCTIONS OF THE
C          FORM BLO+W*EXP(-B*IDEC).  IDEC REPRESENTS THE AMOUNT
C          OF RESOURCE ALLOCATED TO REDUCE THE VALUE OF A
C          PARTICULAR NEGATIVE EXPONENTIAL AND IS THE DECISION
C          VARIABLE.  THE MINIMIZATION IS SUBJECT TO CONSTRAINT
C          THAT THE SUM OF THE IDEC'S FOR EACH FUNCTION IN THE
C          SEQUENCE IS LESS THAN OR EQUAL TO K WHERE K
C          REPRESENTS THE TOTAL RESOURCE AVAILABILTY.  DYNAMIC
C          PROGRAMMING IS THE SOLUTION TECHNIQUE USED.
C
C        THE INPUT PARAMETERS FOR THIS PROGRAM ARE AS FOLLOWS -
C
C             NCUT-NUMBER OF FUNCTIONS TO BE MINIMIZED.
C             K - MAXIMUM AMOUNT OF RESOURCE AVAILABLE.
C             UPBND - PREDETERMINED UPPER BOUND.
C             N - NUMBER OF EXPONENTIALS IN PARTICULAR FUNCTION.
C             B,W,BLO - CONSTANTS ASSOC WITH EACH EXPONENTIAL.
C
C        THE OUTPUT FROM THIS PROGRAM IS AS FOLLOWS -
C             (FOR EACH FUNCTION N DYNAMIC PROGRAMMING STAGES
C             ARE REQUIRED.  FOR EACH STAGE THE
C             OPTIMAL VALUE OF THE DECISION VARIABLE IS PRINTED
C             FOR EACH POSSIBLE VALUE OF THE STATE VARIABLE.)
C
C             ISTATE - VALUE OF STATE VARIABLE.
C             IDEC  - OPTIMAL VALUE FOR DECISION VARIABLE.
C
C             (AT THE END OF THE STAGES THE SOLUTION IS
C             PRINTED FOR EACH FUNCTION)
C
C             AIRCRAFT AVAIL - AMOUNT OF RESOURCE AVAILABLE.
C             CAPACITY - SOLUTION TO MINIMIZATION PROBLEM.
C             DELCAP - INCREMENTAL IMPROVEMENT IN SOLUTION.
C
      DIMENSION FCN(200),FCNNEW(200),CAP(200)
      DIMENSION IPLANE(200),KPLANE(200),DELCAP(200)
      DIMENSION IFIBNO(20)
      DATA IFIBNO/1,2,4,7,12,20,33,54,88,143,232,376,609,
     1986,1596,2583,4180,6764,10945,17710/
C
C             READ # PATHS IN NETWORK, # PLANES AVAIL
C             & LUB ON CAPACITY
C
      READ(5,40) NCUT,K,UPBND
   40 FORMAT(2(I3),F9.4)
      DO 1200 IPATH=1,NCUT
      DATA FCN/200*0.0/,FCNNEW/200*0.0/,CAP/200*0.0/
      DATA IPLANE/200*0/,KPLANE/200*0/
      DATA DELCAP/200*0.0/
C
C             READ # ARCS IN CUT SET
C
      READ(5,50) N
   50 FORMAT(I2)
      M=K+1
C
C             READ IN ARC PARAMETERS (1ST ARC)
C
      READ(5,60) B,W,BLO
   60 FORMAT(3(F10.6))
      BLOSUM=BLO
C
C             DP STAGE 1
C
      DO 100 I=1,M
      ARG=-B*(I-1)
```

```
      FCN(I)=W*EXP(ARG)
      IPLANE(I)=I-1
      KPLANE(I)=IPLANE(I)
  100 CONTINUE
      ISTAGE=1
      WRITE(6,1000) ISTAGE,(KPLANE(I),IPLANE(I),I=1,M)
C
C        DP STAGES 2 THRU N
C
      DO 900 ISTAGE=2,N
C
C        READ IN ARC PARAMETERS (REMAINING ARCS)
C
      READ(5,60) B,W,BLO
      BLOSUM=BLOSUM+BLO
      IF(BLOSUM.GE.UPBND) GO TO 2550
C
C        SET STATE VARIABLE & RUN FIB SEARCH
C
      DO 800 J=2,M
      IX=J-1
      DO 200 NO=1,20
      IF(IFIBNO(NO).GE.J) GO TO 300
  200 CONTINUE
  300 NOM=NO-1
      IA=0
      IB=IFIBNO(NO)-1
      DO 500 ITER=1,NOM
      NO1=NO-ITER
      NO2=NO-ITER-1
      IF(IA+IFIBNO(NO1).GE.J) GO TO 400
      ARG1=-B*(IA+IFIBNO(NO1))
      IT1=IX-IA-IFIBNO(NO1)+1
      IF(NO2.EQ.0) GO TO 350
      ARG2=-B*(IA+IFIBNO(NO2))
      IT2=IX-IA-IFIBNO(NO2)+1
      GO TO 360
  350 ARG2=-B*IA
      IT2=IX-IA+1
  360 F1=W*EXP(ARG1)+FCN(IT1)
      F2=W*EXP(ARG2)+FCN(IT2)
      IF(F2.LT.F1) GO TO 400
      IA=IFIBNO(NO2)+1+IA
      GO TO 500
  400 IB=IFIBNO(NO1)-1+IA
  500 CONTINUE
      FCNNEW(J)=AMIN1(F2,F1)
      IPLANE(J)=IA
      IF(ISTAGE.EQ.N) CAP(J)=FCNNEW(J)+BLOSUM
      IF(ISTAGE.EQ.N.AND.J.GT.2)
     1DELCAP(J)=FCNNEW(J)-FCNNEW(IX)
  800 CONTINUE
      FCN(1)=FCN(1)+W
      DO 850 I=2,M
      FCN(I)=FCNNEW(I)
  850 CONTINUE
      WRITE(6,1000) ISTAGE,(KPLANE(I),IPLANE(I),I=1,M)
  900 CONTINUE
      CAP(1)=FCN(1)+BLOSUM
      DELCAP(2)=FCN(2)-FCN(1)
C
C        END DP STAGES
C
 1000 FORMAT('1'////////' ',29X,'STAGE NUMBER ',I2/////
     1' ',14X,3('ISTATE',4X,'IDEC',2X)///
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
```

```
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X)/
     1' ',14X,3(I6,4X,I4,2X)/' ',14X,3(I6,4X,I4,2X))
      WRITE(6,1050) IPATH
 1050 FORMAT('1'/////////' ',27X,'DP SCLNS CUT SET',I2/////
     1' ',12X,'ACFT AVAIL',5X,'CAPACITY',5X,
     1'CHANGE IN CAPACITY'//)
      M12=M/2
      M121=M12+1
      DO 1100 I=1,M12
      WRITE(6,1075) KPLANE(I),CAP(I),DELCAP(I)
 1075 FORMAT(' ',10X,I10,3X,F10.4,13X,F10.4)
 1100 CONTINUE
      WRITE(6,1125)
 1125 FORMAT('1'////////
     1' ',12X,'ACFT AVAIL',5X,'CAPACITY',5X,
     1'CHANGE IN CAPACITY'//)
      DO 1150 I=M121,M
      WRITE(6,1075) KPLANE(I),CAP(I),DELCAP(I)
 1150 CONTINUE
 1200 CONTINUE
      GO TO 2600
 2550 WRITE(6,2575) IPATH
 2575 FORMAT(///' ',25X,'LUB EXCEEDED BY PATH # ',I2)
 2600 STOP
      END
```

BIBLIOGRAPHY

1. McMasters, A. W., and Mustin, T. M., "Optimal Interdiction of a Supply Network," Naval Research Logistics Quarterly, V. 17, n. 3, p. 261-268, September 1970.

2. Ford, L. R., Jr., and Fulkerson, D. R., Flows in Networks, The RAND Corporation, R 375-PR, August 1962.

3. Nugent, R. O., The Optimum Allocation of Airstrikes Against a Transportation Network for an Exponential Damage Function, M.S. Thesis, Naval Postgraduate School, 1969.

4. Dreyfus, S. E., "An Appraisal of Some Shortest-Path Algorithms," Operations Research, V. 17, n. 3, p. 395-412, May-June 1969.

5. Clarke, S., Krikorian, A., and Rausen, J., "Computing the N Best Loopless Paths in a Network," J. Soc. Indust. Appl. Math., V. 11, n. 4, p. 1096-1102, December 1963.

6. Pollack, M., Shortest Route Solutions of the Kth Best Route Problem, paper presented at the 36th National Meeting of the Operations Research Society of America, Miami Beach, Florida, 10-12 November 1969.

7. Nemhauser, G. L., Introduction to Dynamic Programming, Wiley, 1966.