A RAND NOTE

THE TAXIWAY REPAIR SCHEDULE PROBLEM:
A HEURISTIC RULE AND A BRANCH-AND-BOUND SOLUTION

Louis H. Wegner

October 1982

N-1883-AF

Prepared for

The United States Air Force

Rand
SANTA MONICA, CA. 90406

The research reported here was sponsored by the Directorate of Operational Requirements, Deputy Chief of Staff/Research, Development, and Acquisition, Hq USAF, under Contract F49620-82-C-0018. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

The Rand Publications Series: The Report is the principal publication documenting and transmitting Rand's major research findings and final research results. The Rand Note reports other outputs of sponsored research for general distribution. Publications of The Rand Corporation do not necessarily reflect the opinions or policies of the sponsors of Rand research.

Published by The Rand Corporation

A RAND NOTE

THE TAXIWAY REPAIR SCHEDULE PROBLEM:
A HEURISTIC RULE AND A BRANCH-AND-BOUND SOLUTION

Louis H. Wegner

October 1982

N-1883-AF

Prepared for

The United States Air Force

Rand
SANTA MONICA, CA. 90406

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

PREFACE

Two computer simulation models have been developed at The Rand Corporation for studying means of sustaining and improving wartime sortie-generation capabilities at airbases. TSAR (Theater Simulation of Airbase Resources) is a detailed simulation of the airbase activities required to generate aircraft sorties. TSARINA creates sample patterns of airbase damage from airbase attacks; the output of TSARINA can be used as input to TSAR to permit evaluation of the effects of attacks on sortie generation.

One of the outputs of TSARINA is a list of damaged aircraft taxiway sections and the extent of the damage (the number and sizes of the bomb craters in each section). In TSAR, taxiway repair resources are allocated to repair the damaged taxiway sections and clear paths of taxiways from the runway to the aircraft locations. The heuristic rule presented in this Note was developed to provide TSAR with a straightforward means for determining near-optimal schedules for repairing the damaged taxiway sections.

The work on TSAR and TSARINA was carried out under the Project AIR FORCE Resource Management Program project entitled "Strategies To Improve Sortie Production in a Dynamic Wartime Environment."

SUMMARY

After an attack on an airbase, the system of taxiways that connect aircraft shelters or parking areas to the airbase runway may be so damaged that some of the aircraft are not able to reach the runway. These aircraft will not be able to fly sorties until the taxiways are repaired enough to clear paths from the aircraft locations to the runway.

In this Note, the Taxiway Repair Schedule Problem is defined to be the problem of finding the optimal sequence for repairing the damaged taxiway sections, assuming that each section has a known repair time and that the sections are repaired one at a time. An optimal taxiway repair sequence minimizes the average time that aircraft have been denied access to the runway.

Two procedures are presented for selecting repair schedules: one, a branch-and-bound algorithm, actually determines an optimal schedule, but at a high computational cost (which becomes infeasibly high for large numbers of damaged taxiway sections); the second, a heuristic rule, is computationally simple but does not select optimal schedules for all problems. The two procedures are compared for 100 example problems.

ACKNOWLEDGMENT

The author is indebted to L. W. Miller, a Rand consultant, for several suggestions that improved the substance and content of this Note. In particular, he suggested the final version of the arc selection criterion for the heuristic rule. This criterion is (both conceptually and computationally) simpler than the one originally used and, on the average, selected better repair schedules for the 100 sample problems of Sec. IV than did the original criterion, although it selected two fewer optimal schedules.

CONTENTS

I. INTRODUCTION

A bombing attack on an airbase may so damage the system of taxiways that lead from aircraft shelter and parking locations to the base runway that some of the aircraft cannot reach the runway. The Taxiway Repair Schedule Problem, as defined below, is concerned with determining a repair schedule minimizing the average time that the aircraft do not have access to the runway through paths of undamaged or repaired taxiways. The more general problem of allocation of repair resources when both taxiways and runway are damaged is not considered.

Figure 1 depicts the aircraft shelters, taxiways, and runway of an example airbase similar to an actual NATO airbase. Typically, there are three separate squadron areas containing aircraft shelters or parking aprons with the aircraft locations in each squadron area interconnected by a set of taxiways. Each squadron area is connected directly to the runway or to parking aprons by two or more taxiways, and the parking aprons are joined to the runway by additional taxiways.

The complex of aircraft locations and taxiways are treated as a network. Nodes represent locations of aircraft or the intersections of taxiways with other taxiways or the runway (the runway is represented as a single node). Arcs represent the sections of taxiways between nodes. We associate with each node the number of aircraft in shelters or parking areas at the node and with each arc the repair time of the taxiway section represented by the arc. Figure 2 contains a simplified network representation for the airbase of Fig. 1. In the figure, node 1 represents the runway and the numbers alongside the nodes the numbers of

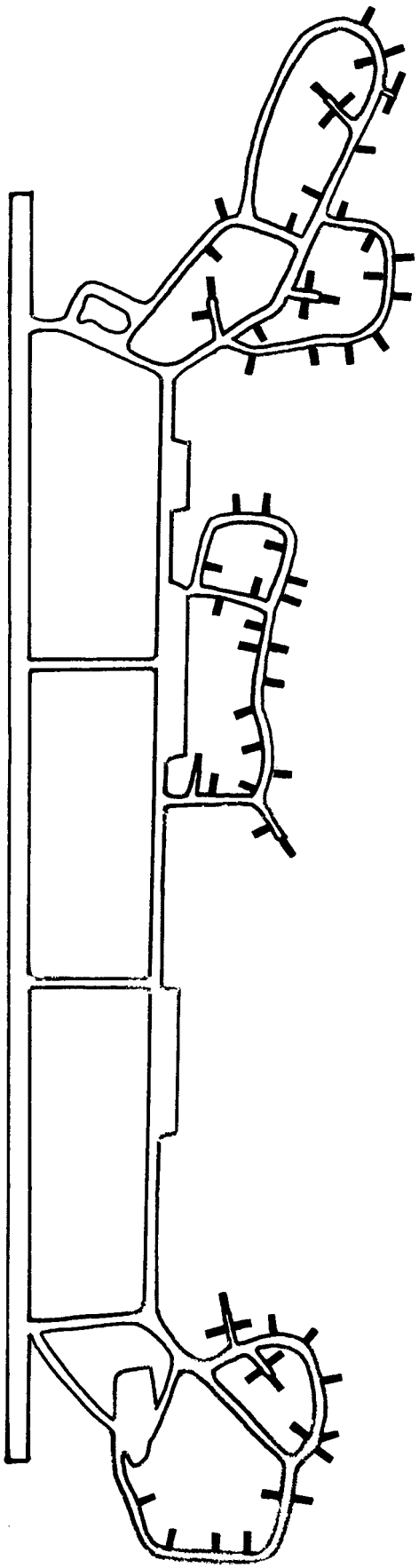

Fig. 1--Taxiway structure for the example airbase

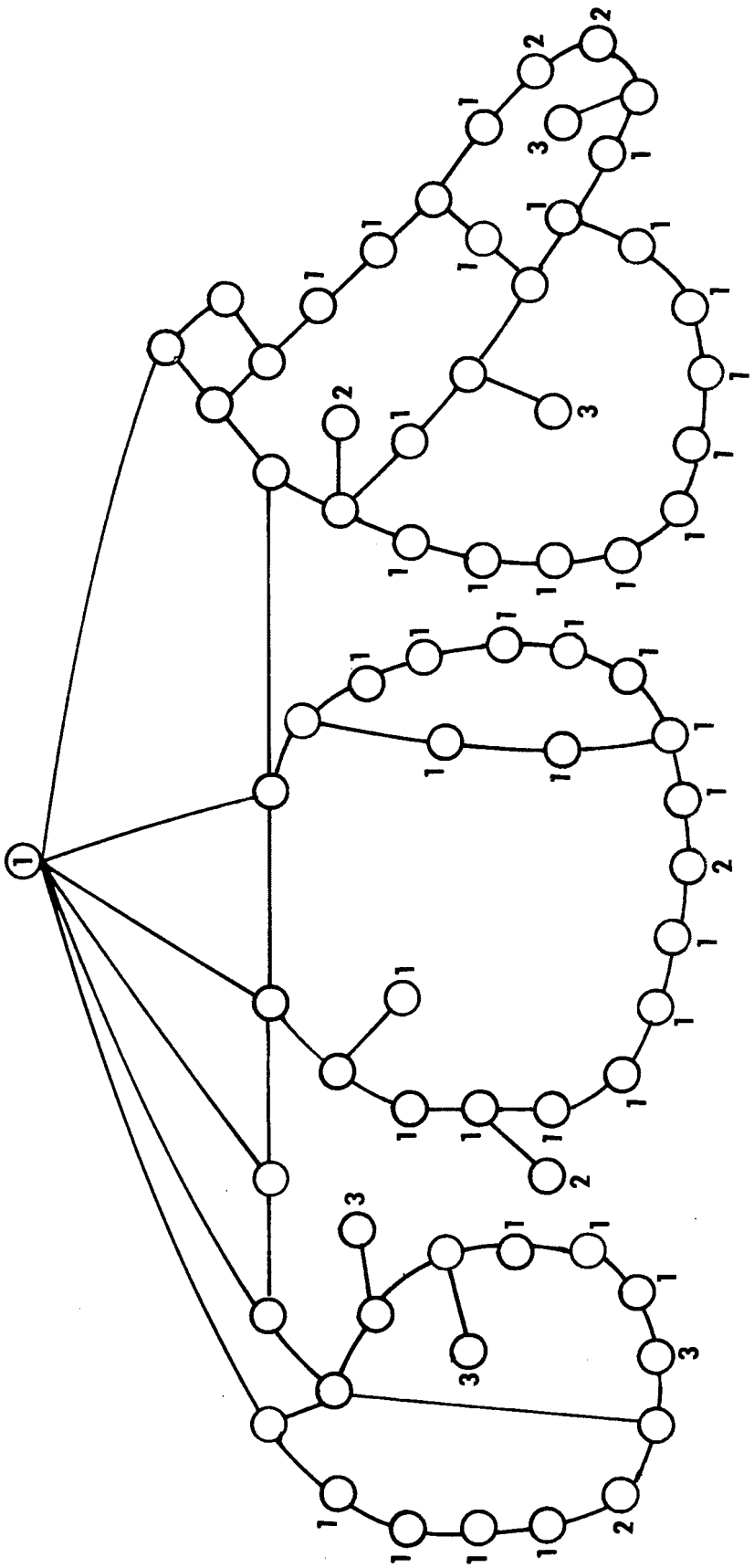

Fig. 2--Node-arc representation for the example airbase

aircraft at the nodes--small segments of taxiways (stubs) that connect the aircraft locations to taxiways have been ignored in this example.

Since an aircraft can not fly until a clear path of taxiways connects the aircraft location with the runway, a measure of the loss of productivity of an aircraft is the elapsed time before the aircraft has access to the runway after the bombing attack. It is assumed that each damaged arc has a repair time associated with it and that the arcs are repaired one at a time until all aircraft have access to the runway. The repair times are assumed to have integral values but the unit of time is arbitrary; it can simply be the number of bomb craters to be filled, assuming one bomb crater can be filled in one unit of time. The repair times are assumed to be independent of the order in which the arcs are repaired; the time to travel between arcs is assumed to be either insignificant compared with the repair times or independent of the distance between arcs. (The heuristic rule presented here could be easily modified to include repair times dependent upon the order in which the arcs are repaired.) The sequence of arcs repaired over time is a "repair schedule" and the sum of the times for each aircraft to have a path cleared to the runway as arcs are repaired is the "loss" associated with that repair schedule. The Taxiway Repair Schedule Problem is to find the repair schedule that has the minimum loss or, equivalently, that minimizes the average time before all the aircraft have access to the runway.

When there are several taxiway repair teams on an airbase, it may be necessary (or much more efficient) for each arc to be repaired by a

separate team. The repair schedule optimization problem is then different from the one formulated here, but the arc repair sequence selected by the heuristic rule of Sec. II can still be used as the order in which damaged arcs are assigned to repair teams.

If there are N arcs to be repaired, then there are N! different possible repair schedules. Since, for example, with N as small as 10 there are over 3 million different possible repair schedules, any algorithm that determines the optimal schedule must examine only a small portion of the repair schedules to be computationally feasible for very large N. The heuristic rule presented in Sec. II examines a maximum of N(N+1)/2 partial repair schedules; the branch-and-bound optimization algorithm of Sec. III partitions the total set of repair schedules into finer and finer subsets and, in general, needs to make calculations for only a fraction of the total subsets.

Considerable computation can be saved if it is recognized that one need consider only those repair schedules that open additional clear paths to the runway as each arc is repaired (the terminal nodes of each repaired arc have access to the runway at the time the arc is repaired). Suppose a schedule does not have this property. Then the first arc later in the repair sequence that did open an additional clear path to the runway could be placed in the repair sequence immediately ahead of an arc that did not, without increasing the loss; repeated rearrangements of the arc repair sequence of this type will eventually lead to a repair schedule that clears additional paths to the runway as each arc is repaired and has no greater loss than the original schedule.

Both the heuristic rule and the branch-and-bound algorithm automatically exclude repair schedules that do not create additional clear paths to the runway as each arc is repaired.

In the network representation for the taxiways of an airbase, an undamaged arc may be removed and its terminal nodes combined into a single node (including the aircraft from both nodes). After undamaged arcs are removed one at a time, a reduced network consisting of only damaged arcs will remain. If two or more arcs join the same nodes, the arcs with the larger repair times may be discarded until only one remains. This further reduces the size of the network (the computer program described in the appendix automatically performs both reductions). Hereafter, it is assumed that all arcs are damaged--the reduced network representation of the taxiway system is used.

II. THE HEURISTIC RULE

The heuristic rule determines an arc repair schedule by sequentially selecting arcs to be repaired. It works outward from the runway to the nodes that do not have access to the runway and uses a criterion for selecting the next arc to be repaired that approximates the (current) maximum of the number of aircraft that will have access to the runway per unit of repair time expended.

STEPS IN THE HEURISTIC RULE

While there are still nodes with aircraft not having access to the runway, select the next arc to be repaired by the following steps:

1. For each node k still without access to the runway, perform the following calculations:

    a. By using a shortest path algorithm, determine the minimum-repair-time path from the given node to the runway; ties are settled by selecting the first such path found.

    b. Set T_k equal to the total repair time for the arcs on the path of step 1.a and set A_k equal to the total number of aircraft at nodes on this path.

2. Find K, the value of k that maximizes A_k/T_k.

3. Working outward from the runway to node K along the minimum-repair-time path to node K, find the first arc on the path that has not yet been repaired. This is the next arc to be repaired.

AN EXAMPLE TAXIWAY REPAIR SCHEDULE PROBLEM

In Fig. 3, damaged arcs are indicated by including the arc repair time alongside the arc. The figure contains the same network representation as that of Fig. 2, but only the nodes are numbered that will remain after the network is reduced. Figure 4 contains the reduced network representation. The top number alongside each arc is the arc number and the bottom number is the arc repair time. The top number at each node is the node number, and the bottom number is the number of aircraft at the node. Table 1 summarizes the data for the reduced network.

Table 1

DATA FOR THE EXAMPLE PROBLEM

Node Number	Air-craft	Arc Number	From Node	To Node	Repair Time
2	2	1	1	2	1
3	1	2	1	5	3
4	15	3	1	6	2
5	18	4	1	7	2
6	2	5	2	3	1
7	19	6	2	4	1
8	9	7	3	4	1
		8	5	6	1
		9	7	8	1

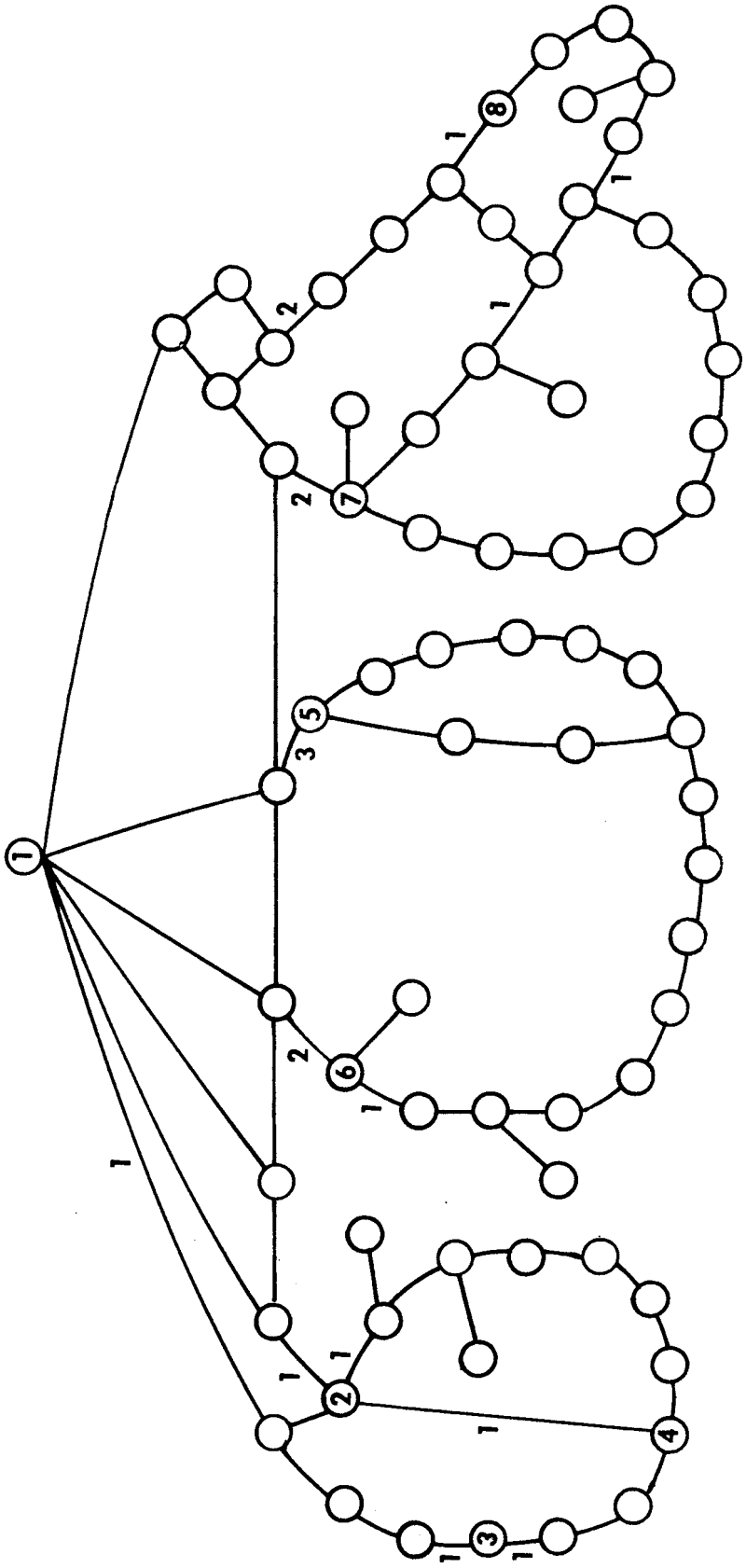

Fig. 3-- Nodes and arc repair times for the example problem

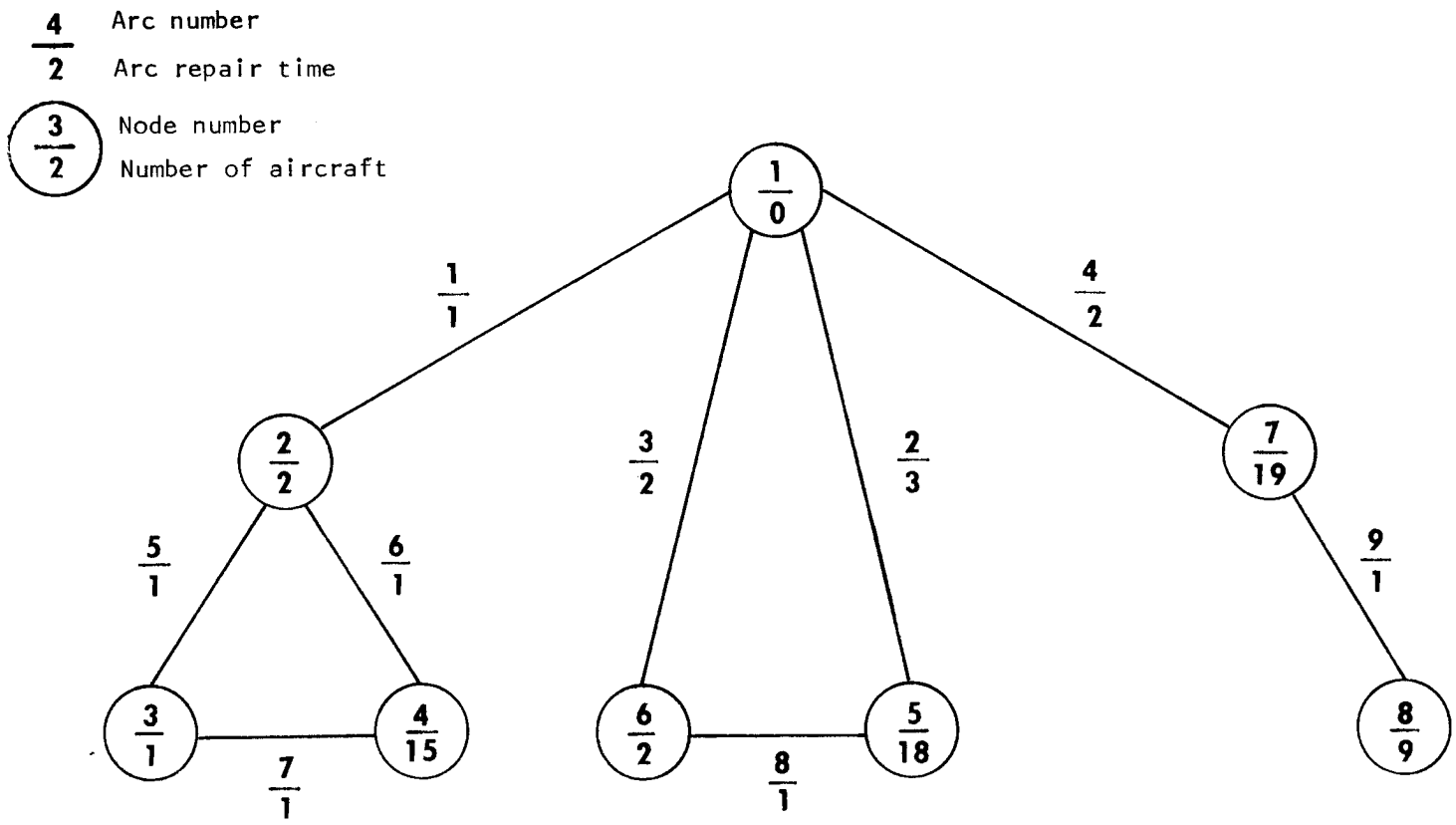

Fig. 4--Nodes, aircraft, arcs, and arc repair times for the reduced example problem

Table 2 summarizes the values obtained for the example problem during the steps of the heuristic rule and presents the arc repair sequence selected.

From Table 2, the arc repair schedule obtained by the heuristic rule is to repair arcs in the sequence 4916385 with a loss of 315 time units. From Sec. III, this is an optimal schedule. Note that this schedule does not include arcs 2 and 7 because all of the aircraft have clear paths to the runway without repair of these two arcs.

Table 2

REPAIR SCHEDULE SELECTED BY THE HEURISTIC RULE

Step	Nodes in Step	Values on the Minimum-repair-time Path					Repair Sequence Selected	Aircraft Reaching Runway	Cum. Time Loss
		Nodes	Arcs	A_k	T_k	A_k/T_k			
1	2	1,2	1	2	1	2.00			
	3	1,2,3	1,5	3	2	1.50			
	4	1,2,4	1,6	17	2	8.50			
	5	1,6,5	3,8	20	3	6.67			
	6	1,6	3	2	2	1.00			
	7	1,7	4	19	2	9.50	4	19	132
	8	1,7,8	4,9	28	3	9.33			
2	2	1,2	1	2	1	2.00			
	3	1,2,3	1,5	3	2	1.50			
	4	1,2,4	1,6	17	2	8.50			
	5	1,6,5	3,8	20	3	6.67			
	6	1,6	3	2	2	1.00			
	8	1,7,8	4,9	9	1	9.00	9	28	179
3	2	1,2	1	2	1	2.00			
	3	1,2,3	1,5	3	2	1.50			
	4	1,2,4	1,6	17	2	8.50	1	30	217
	5	1,6,5	3,8	20	3	6.67			
	6	1,6	3	2	2	1.00			
4	3	1,2,3	1,5	1	1	1.00			
	4	1,2,4	1,6	15	1	15.00	6	45	253
	5	1,6,5	3,8	20	3	6.67			
	6	1,6	3	2	2	1.00			
5	3	1,2,3	1,5	1	1	1.00			
	5	1,6,5	3,8	20	3	6.67	3	47	295
	6	1,6	3	2	2	1.00			
7	3	1,2,3	1,5	1	1	1.00			
	5	1,6,5	3,8	18	1	18.00	8	65	314
8	3	1,2,3	1,5	1	1	1.00	5	66	315

III. A BRANCH-AND-BOUND SOLUTION

Branch-and-bound procedures are enumerative schemes that set a framework for a structured search of the possible solutions for an optimal solution. In general, only a small fraction of the possible solutions need actually be enumerated, the remaining solutions being eliminated through the application of bounds establishing that such solutions cannot be optimal. The set of all possible solutions is partitioned successively into smaller and smaller subsets, and a lower bound (for the case of minimization considered here) is calculated for the loss associated with the solutions in each subset. After each partitioning, those subsets whose lower bound exceeds the loss of a known solution (or some otherwise determined upper bound on the minimum loss) are discarded. The partitioning continues until a solution is found whose loss is no greater than the lower bound of any remaining subset. This is an optimal solution.

A survey of branch-and-bound methods is given in Lawler and Wood (1966). Applications of branch and bound to scheduling problems may be found in Ignall and Schrage (1965) and Miller (1974).

The branch-and-bound solution process used here may be represented as a tree structure, as illustrated in Fig. 5. A node of order k represents all repair schedules having the same repair sequence for the first k repaired arcs; the node of order zero represents all repair schedules. Note that the repair schedules represented by the nodes of order k form a partition of the total set of repair schedules. In the figure, the partial repair schedule is the circled sequence of numbers.

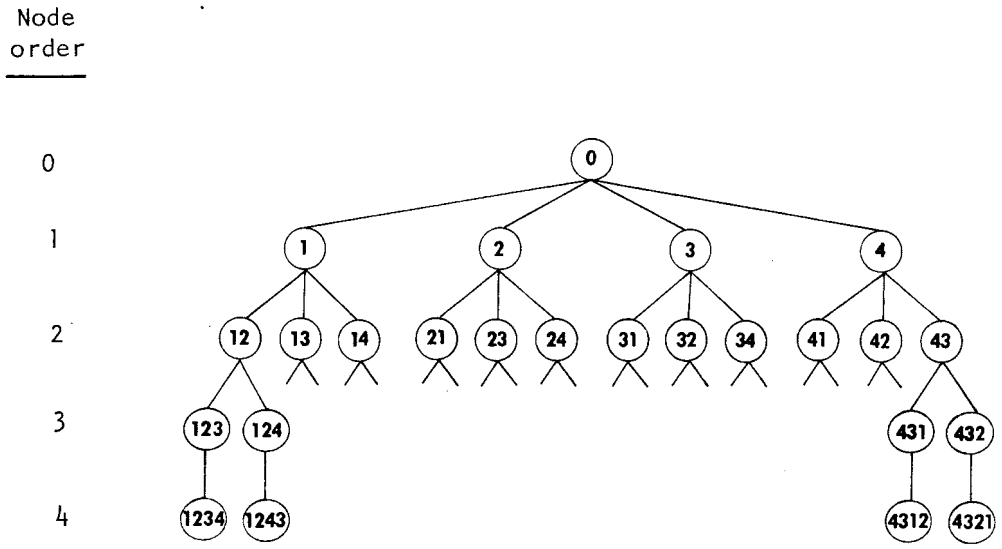

Fig. 5--Branch-and-bound tree for four damaged arcs

A node of order N - 1 represents only one complete repair schedule because once a sequence of N - 1 repaired arcs is specified there is only one possibility for the last arc to be repaired.

The partial repair schedule associated with a node in the tree is the partial repair schedule (PRS) of the node, and the subset of repair schedules represented by the node is the PRS class of the node.

LOWER BOUNDS FOR THE PRS CLASSES

The nodes of order k are formed by adding N - k + 1 new nodes for each node of order k - 1, by adding one arc to the partial repair schedule for each arc that is not yet included. (Thus, after each order is generated there are a total of $N(N - 1) \cdots (N - k + 1)$ nodes). The node immediately above a given node on the same branch will be called the predecessor of the given node and a node immediately below, a successor to the given node.

For each node of order k, a lower bound for the node and the value of the time loss for aircraft with access to the runway after completion of the PRS for the node are obtained from:

$$LB = L + (A - R)T + \Sigma P_j A_j \qquad (3.1)$$

$$L = L_0 + (R - R_0)T \qquad (3.2)$$

where LB = the lower bound on the losses for all repair schedules in the given PRS class,

L = the total loss for those aircraft with access to the runway after completion of the given PRS,

L_0 = the total loss for those aircraft with access to the runway after completion of the predecessor PRS of the given PRS (equals zero when k = 1),

R = the number of aircraft with access to the runway after completion of the given PRS,

R_0 = the number of aircraft with access to the runway after completion of the predecessor PRS of the given PRS (equals the number of aircraft with access to the runway without repairs when k = 1).

T = the repair time for for the arc repaired last in the given PRS,

A = the total number of aircraft at the airbase,

P_j = the repair time for the minimum-repair-time path from node j to the runway (after completion of the given PRS),

A_j = the number of aircraft at node j,

and the summation is over all taxiway nodes.

Because, when the lower bound LB is formed, the loss associated with each aircraft is the repair time of the given PRS plus the time for repairing the arcs in the minimum-repair-time path for the aircraft after the given PRS is completed, it follows that each LB is a lower bound for all repair schedules in the corresponding PRS class.

The PRSs associated with nodes of order N are complete repair schedules, and the lower bounds for the corresponding classes are the actual losses for those repair schedules. Any repair schedule of order N with the minimum loss is an optimal schedule.

PRUNING THE TREE (ELIMINATION OF PRS CLASSES)

When carried to completion, the successive partitionings of the set of repair schedules generates N! Nth order nodes. Obviously, it would be desirable to eliminate as many as possible of the branches (nodes plus their successor nodes) to avoid generating the entire tree. Any node (and its successor nodes) whose repair schedules can be shown not to be optimal can be discarded--any node whose lower bound is larger than the loss for any given repair schedule, e.g., the one selected by the heuristic rule. This is the "bound" part of the branch-and-bound procedure. In applying bounding, the loss for the repair schedule of the heuristic rule or any better repair schedule found is used if nodes of order N have been generated.

A node can be discarded if it can be shown by some other means that there is a better repair schedule--one with a smaller loss than any repair schedule represented by the node. A given node is "dominated" by another node if the latter node represents at least one repair schedule with a smaller loss than any repair schedule of the given node. (Thus, as used here, bounding is a particular case of dominance.) The following describes three dominance criteria that are used along with bounding to substantially reduce the number of nodes that are actually generated:

1. As discussed in Sec. I, any node with a PRS whose last-repaired arc does not add a clear path to the runway is dominated by some other node.

2. Any node with a PRS whose last-repaired arc has both of its terminal nodes already joined to the runway by clear paths not involving the just-repaired arc is dominated by some other node.
3. If two PRSs clear paths to the runway for the same taxiway nodes, the corresponding node with the larger lower bound is dominated by the one with the smaller lower bound; if the two lower bounds are equal an arbitrary one of the two nodes can be discarded. (Because the two PRSs clear paths for the same taxiway nodes, comparing lower bounds is equivalent to comparing the values of L from Eq. (3.2), the total loss for all aircraft with access to the runway after completion of a given PRS.)

BRANCHING STRATEGIES

The order in which the nodes of the branch-and-bound tree are generated is called the "branching strategy." Because all nodes are either added to the tree or discarded by the application of bounding or the dominance criteria, one branching strategy is better than another if it uses bounding and the dominance criteria to greater effect--finds an optimal solution with less computational cost.

The branching strategy used here, a "breadth-first" strategy, is to generate nodes of order k before those of order k + 1. This permits an efficient scheme to be used for applying the third dominance criterion (see the appendix). For all (remaining) nodes of order k, the successor nodes of order k + 1 are generated one at a time, using

bounding and the dominance criteria to eliminate as many nodes as possible. (The kth order node is discarded after all its k + 1st order successor nodes have been generated.)

One alternative branching strategy, a "depth-first" strategy, starts by generating the nodes of order 1. These nodes are then ordered by the values of their lower bounds. Then the successor nodes to the node with the smallest lower bound are generated and added to the list, the predecessor node and any dominated nodes having been discarded. This process is repeated until a node appears lowest on the list that has a complete repair schedule. This repair schedule is optimal because the PRS classes represented by the nodes remaining on the list have lower bounds larger than that of the repair schedule (whose lower bound is its actual loss). This branching strategy obtains complete schedules earlier in the procedure than does the selected strategy, so it may prune more nodes in applying bounding than the selected strategy when the heuristic rule has selected a poor repair schedule.

The alternative branching strategy has not been compared with the selected branching strategy because the computer program described in the appendix carried out only the selected branching strategy.

A SIMPLIFIED ALGORITHM WHEN THE REDUCED TAXIWAY NETWORK HAS NO LOOPS

When the reduced network representation for the damaged taxiway network has no loops, the calculation of the lower bound may be simplified. Although this has no general applicability to the taxiway repair schedule problem (there will normally be loops), the simplification is presented here as it may be useful in other similar scheduling problems.

By far the largest computational cost of the branch-and-bound algorithm is the determination of the P_js of Eq. (3.1), which requires solving a shortest path problem for each node added to the branch-and-bound tree. When the reduced node-arc representation for the taxiway structure has no loops (reduces to a tree), there is only one possible path to the runway for each taxiway node. This fact may be utilized to determine each lower bound by only one multiplication and three additions and subtractions as each node is generated from its predecessor node. The remainder of the algorithm is unchanged except that the second dominance criterion is no longer relevant because there are no loops.

The lower bound for the PRS class represented by each node is determined as follows from that of its predecessor node: Initially, calculate a lower bound for all repair schedules (for the PRS class of order zero) as

$$LB_0 = \Sigma A_j P_j$$

where LB_0 = the (initial) lower bound for each repair schedule,

A_j = the number of aircraft at node j,

P_j = the minimum-repair-time path from node j to node 1 (the runway) without any repairs.

The lower bound for every other node in the branch-and-bound tree is calculated from

$$LB = LB_0 + (A - R - C)T$$

where LB = the lower bound for the given PRS class,

LB_0 = the lower bound for the PRS class of the predecessor node of the given node,

A = the number of aircraft on the airbase,

R = the number of aircraft with access to the runway after completion of the predecessor PRS of the given PRS,

C = the number of aircraft whose paths to the runway include the arc added in forming the given PRS from its predecessor PRS.

THE OPTIMAL SOLUTION FOR THE EXAMPLE PROBLEM

Figure 6 contains nodes of the branch-and-bound tree for the example problem of Fig. 3 and Table 1. To save space in the figure, each node is identified by the number of the last arc in the PRS; the PRS is obtained, in reverse order, by moving upward from predecessor node to predecessor node to the node of order zero. Only those nodes for which lower bounds were calculated are included in the figure (nodes eliminated by the first two dominance criteria are not included). Nodes in the figure with no successor nodes are eliminated by bounding or the third dominance criterion. Table 3 summarizes the node elimination process for those nodes. In the table "Branched from" indicates nodes for which successor nodes were generated; "Dom. by X" indicates nodes dominated by node "X" according to the third dominance criterion; and "LB > UB" indicates that the lower bound for the node is larger than the loss (315 time units) for the repair schedule of the heuristic rule of Sec. II.

The last two entries in Table 3 contain optimal repair schedules for the example problem, arc repair sequences 4916385 and 4916387, and the minimum loss, 315 time units.

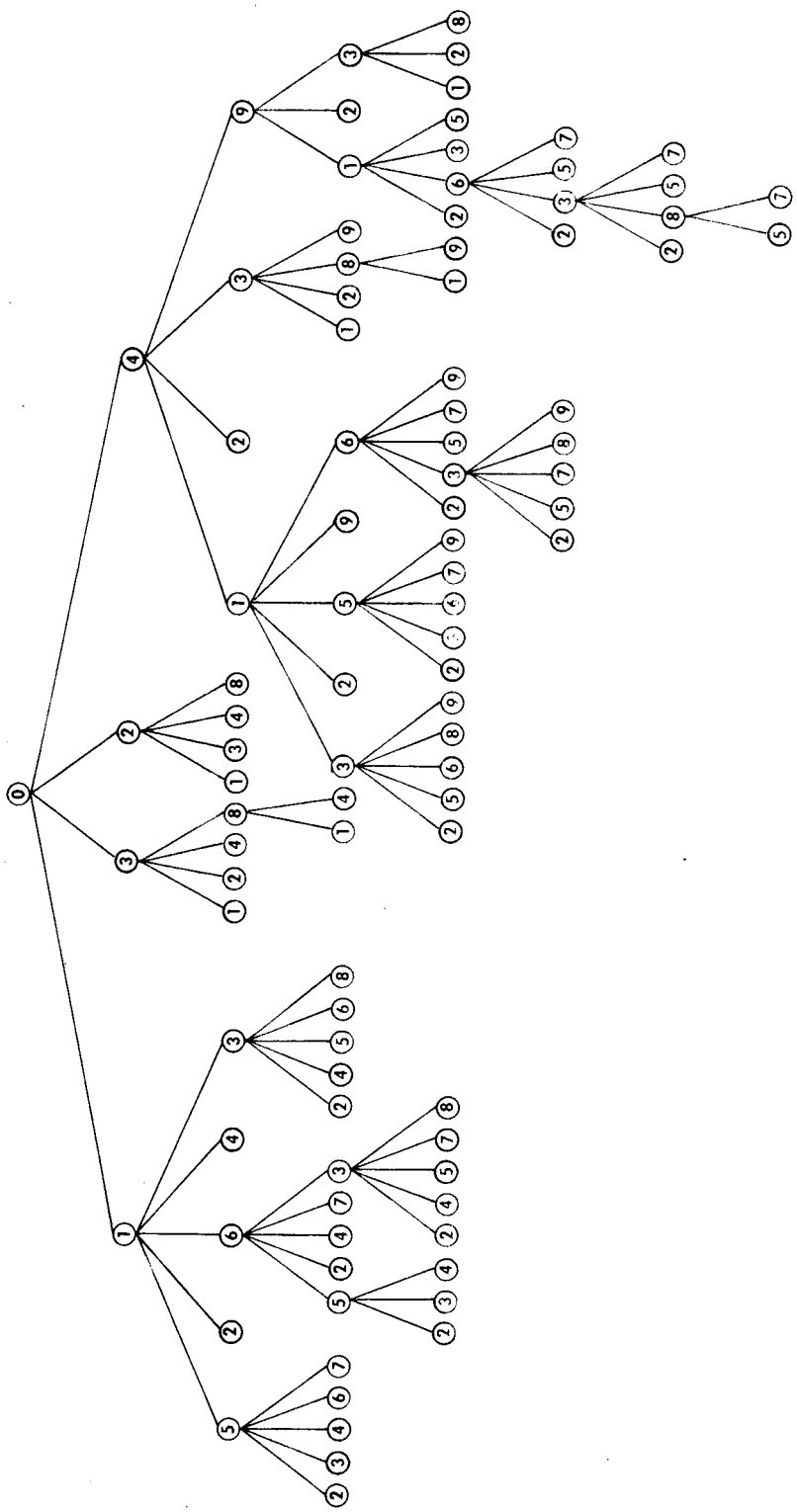

Fig. 6--Branch-and-bound tree for the example problem

Table 3

PRS CLASSES FOR WHICH LOWER BOUNDS WERE CALCULATED

PRS	LB	Disposition	PRS	LB	Disposition
1	205	Branched from	492	329	LB > UB
2	299	Branched from	493	307	Branched from
3	249	Branched from	1632	435	LB > UB
4	233	Branched from	1634	350	LB > UB
12	341	LB > UB	1635	358	LB > UB
13	293	Branched from	1637	358	LB > UB
14	277	Dom. by 41	1638	341	LB > UB
15	268	Branched from	1652	390	LB > UB
16	254	Branched from	1653	358	LB > UB
21	329	LB > UB	1654	342	LB > UB
23	393	LB > UB	4132	432	LB > UB
24	339	LB > UB	4135	354	LB > UB
28	345	LB > UB	4136	340	LB > UB
31	295	Dom. By 13	4138	337	LB > UB
32	423	LB > UB	4139	346	LB > UB
34	321	LB > UB	4152	382	LB > UB
38	295	Branched from	4153	354	LB > UB
41	262	Branched from	4156	335	LB > UB
42	318	LB > UB	4157	335	LB > UB
43	287	Branched from	4159	341	LB > UB
49	271	Branched from	4162	326	LB > UB
132	461	LB > UB	4163	312	Branched from
134	361	LB > UB	4165	321	LB > UB
135	354	LB > UB	4167	321	LB > UB
136	340	LB > UB	4169	313	Dom. by 4916
138	337	LB > UB	4381	323	LB > UB
152	401	LB > UB	4389	332	LB > UB
153	354	LB > UB	4912	343	LB > UB
154	338	LB > UB	4913	323	LB > UB
156	316	LB > UB	4915	326	LB > UB
157	316	LB > UB	4916	312	Branched from
162	345	LB > UB	4931	325	LB > UB
163	312	Branched from	4932	397	LB > UB
164	296	Dom. by 416	4938	325	LB > UB
165	302	Branched from	41632	378	LB > UB
167	302	Dom. by 165	41635	339	LB > UB
381	323	LB > UB	41637	339	LB > UB
384	331	LB > UB	41638	322	LB > UB
412	341	LB > UB	41639	331	LB > UB
413	312	Branched from	49162	319	LB > UB
415	306	Branched from	49163	314	Branched from
416	292	Branched from	49165	332	LB > UB
419	298	Dom. by 491	49167	332	LB > UB
431	314	Dom. by 413	491632	353	LB > UB
432	404	LB > UB	491635	332	LB > UB
438	314	Branched from	491637	332	LB > UB
439	323	LB > UB	491638	315	Branched from
491	291	Branched from	4916385	315	Optimal sol.
			4916387	315	Optimal sol.

IV. RESULTS FOR EXAMPLE PROBLEMS

The branch-and-bound algorithm is too expensive computationally to be practical as a general purpose procedure (at least for a large number of damaged arcs). Although the heuristic rule is efficient, it is an ad hoc procedure with no guarantee that it will produce optimal or near-optimal solutions for all taxiway repair schedule problems. For some empirical evidence that the heuristic rule provides good solutions, the optimal and heuristic rule solutions were compared for 100 example problems.

The example problems were generated by Monte Carlo sampling of the damaged arcs and their repair times, with the taxiway node-arc representation and aircraft locations of Fig. 2 for the airbase of Fig. 1. As the taxiway system for this airbase is more complex than that of most airbases, the set of example problems should provide a representative replacement set for the range of problems generated by actual attacks against NATO airbases.

Each problem was created by first selecting damaged arcs at random from a specified subset of the arcs. Seventy-five of the problems had the damaged arcs chosen from arcs in the individual aircraft squadron areas--10 problems with 10 damaged arcs in each of the three squadron areas and 15 problems with 20 damaged arcs in each of the three squadron areas (except that the left-most squadron area has only 18 arcs). In the remaining 25 problems, 20 damaged arcs were chosen at random from the entire set of 82 arcs. The repair time for each damaged arc was selected uniformly at random over the range of 1 to 5 time units.

The optimal repair schedule provided by the branch-and-bound algorithm and the repair schedule of the heuristic rule were obtained for each example problem by use of the computer program described in the appendix. Table 4 summarizes the set of example problems and the results of the comparison between the losses for the optimal repair schedules and the losses for the repair schedules of the heuristic rule.

From the last entry in the table, the heuristic rule produces the optimal solution in 59 percent of the problems and, over all the example problems, the losses for the heuristic rule solutions average less than 1 percent higher than the losses for the optimal solutions.

Table 4

COMPARISON OF THE HEURISTIC RULE AND OPTIMAL REPAIR SCHEDULE LOSSES FOR ONE HUNDRED EXAMPLE PROBLEMS

Squadron Area Damaged	Taxiway Arc Numbers	Number of Arcs Damaged	Number of Problems	Both Solns. Iden.	Opt. Sol. Average Loss	Repair Schedule Loss Differences	
						Max.(%)	Ave.(%)
Left	7-24	10	10	7	136	2	0.3
Center	27-49	10	10	4	162	7	1.2
Right	51-80	10	10	6	158	8	1.1
Left	7-24	18	15	6	390	3	0.7
Center	27-49	20	15	8	452	3	0.8
Right	51-80	20	15	11	517	2	0.2
All	1-82	20	25	17	324	10	1.1
Summary			100	59	330	10	0.8

Appendix

SOME COMPUTER PROGRAM DETAILS

A FORTRAN computer program was written combining the heuristic rule of Sec. II and the branch-and-bound algorithm of Sec. III. Although the heuristic rule is computationally quite efficient, the branch-and- bound algorithm may (potentially) have to evaluate a very large number of nodes in the branch-and-bound tree. Several computational algorithms and list processing schemes were used to help minimize the program computer memory and computation time required for the branch- and-bound algorithm.

The computer program first performs the network reduction process (described in Sec. I) that eliminates undamaged arcs and multiple arcs with the same terminal nodes (keeping one such arc with the shortest repair time). The remaining nodes and arcs are renumbered and the reduced taxiway network is used for both the heuristic rule and the branch-and-bound algorithm.

Both procedures have steps requiring the determination of the minimum-repair-time paths from all other nodes in the network to node 1 (the runway). This is equivalent to finding the shortest paths to a given node from all other nodes in a network. Both procedures use a FORTRAN subroutine for a "label correcting" algorithm, the steps of which are (in terms of repair times):

1. Assign a label D(i) (a value larger than the repair time of any minimum-repair-time path) to each node i, except for node 1, where D(1) = 0.

2. Search for an arc such that $R(i,j) + D(j) < D(i)$ for any nodes i and j, where $R(i,j)$ is the repair time of arc (i,j). Replace $D(i)$ by the value of $R(i,j) + D(j)$.
3. If any arc is found in Step 2, repeat the search. When no more can be found, terminate the algorithm. The last value of $D(i)$ is the repair time of the minimum-repair-time path from node i to node 1, and the last arc found for each node is part of the minimum-repair-time path to node 1 for that node.

Label-correcting algorithms have the desirable property of simultaneously generating the minimum-repair-time paths from all nodes to the first node. Starting from a given node and moving from one node to the next node (through the last arc used for each node at step 2) until node 1 is reached, the minimum-repair-time path is generated for the starting node.

Label-correcting algorithms differ in how the search in step 2 is carried out. This Note uses the one in Golden (1976). The speed of this version of the algorithm compared very favorably with several other shortest path algorithms for a class of sparse networks (networks with a small number of arcs at each node, as in taxiway networks) for a sample set of problems with up to a thousand nodes (Denardo and Fox, 1979).

Step 2 is as follows: starting with node 1 as the first member, a list of "active" nodes is formed and used sequentially, from the top of the list, as the node j of step 2. Any node i that is not currently active (in the list) and whose label is decreased at step 2 is added to the bottom of the list. After being used in step 2, node j becomes inactive and is deleted from the top of the list (it may subsequently

become active again and added back to the bottom of the list). When the list is empty the algorithm is complete and the value of the label at a given node is the repair time of the minimum-repair-time path from that node to node 1.

Any kth order node (PRS class) in the branch-and-bound tree that is not eliminated by the second dominance criterion of Sec. III has k arcs in its PRS that have cleared paths to the runway for k taxiway nodes. One word of memory (32 bits) is used to indicate the arcs in the PRS (the PRS arc-word) and one word to indicate the taxiway nodes with cleared paths to the runway (the PRS node-word). A one in the bit position corresponding to the arc number in the arc-word indicates that the arc is in the PRS, a zero indicates the contrary. The PRS node-word indicates the taxiway nodes with cleared paths to the runway in a similar fashion. Thus, for the branch-and-bound algorithm, the maximum number of both arcs and nodes in the reduced taxiway network is 32 in the current version of the computer program. (For the heuristic rule, the practical limits for the numbers of arcs and nodes are much larger; they have been arbitrarily set to 300 for both arcs and nodes in the computer program.)

Thirty-two half-words of memory are used for each PRS to indicate the node numbers of the nodes already having access to the runway and the arc numbers for the adjoining arcs not in the PRS but that have a terminal node cleared to the runway by the PRS. For a kth order PRS, the first k entries contain the node numbers of the nodes and entries k + 1 up to 32 the arc numbers of any contiguous arcs. This list is used to directly generate for each PRS class the successor PRS classes that are not eliminated by the first dominance criterion of Sec. III. The

PRSs of the (surviving) successor PRS classes are generated by adding one of the adjoining arcs to the PRS, adding the taxiway node cleared by the added arc to the k nodes cleared to the runway by the PRS, and determining any new adjoining arcs for the added node. A successor PRS class is eliminated by the second dominance criterion if both terminal nodes of the added arc already have paths cleared to the runway by the predecessor PRS.

The PRS node-word representation conserves memory and allows for an efficient application of the third dominance criterion of Sec. III. All PRSs clearing paths to the runway for the same nodes have the same numerical values for their node-words. As the PRS classes of order k are generated their PRSs are placed in an ordered list (called a heap), using the numerical value of the PRS node-word for the ordering. With a subroutine for the Heapsort algorithm of (Williams, 1964), the entries in the heap with the same numerical value are found in order, and all PRS classes with the same PRS node-word value are discarded except for one with the smallest loss.

REFERENCES

Denardo E. V., and B. L. Fox, "Shortest-Route Methods: 1. Reaching, Pruning, and Buckets," Operations Research, Vol. 27, No. 1, January 1979.

Golden, B., Shortest-Path Algorithms: A Comparison," Operations Research, Vol. 24, No. 1, January 1976.

Ignall, E., and L. Schrage, "Application of the Branch-and-Bound Technique to Some Flow-Shop Scheduling Problems," Operations Research, Vol. 13, No. 3, May 1965.

Lawler, E. L., and D. E. Wood, "Branch-and-Bound Methods: A Survey," Operations Research, Vol. 14, No. 4, July 1966.

Miller, L. W., "Branch-and-Bound and Heuristic Approaches to a Sequencing Problem with Team-size Requirements," AIEE Transactions, Vol. 6, No. 3, September 1974.

Williams, J. W. J., "Algorithm 232: Heapsort," Communications of the ACM, Vol. 7, No. 6, June 1964.