MAS REPORT 97 – 1

TRANSACTIONS

MATHEMATICS, COMPUTER SCIENCE, STATISTICS DIVISION

MISSISSIPPI ACADEMY OF SCIENCES

SIXTY-FIRST MEETING

FEBRUARY 20 & 21, 1997

BILOXI, MISSISSIPPI

Network Routing and Unit/Task Scheduling Examples.

Dr. Andrew W. Harrell
U.S. Army Waterways Experiment Station
Mobility Systems Division
Geotechnical Laboratory
Vicksburg, MS. 39181

This methodology has applications to both logistics haul problems and the deployment of possible tactical unit movements. Two different C-code versions of variations of the basic Edmonds-Karp/Dinic network maximum flow/min-cost[a] algorithms were written. These different versions enable a user to solve routing and scheduling problems interactively. The programs optimize the traffic flow (throughput) across a network of movement corridors on an arc digitized raster graphic program. They also schedule units to move through the network to perform tasks.

1. **Introduction.** With the development of computer graphics techniques for displaying arc digitized raster graphic (ADRG) information, new ways of representing unit movement, and aircraft or ship routing have been developed. Problems exist in giving these programs the capability to represent the effects of various types of on- and off-road obstacles, underwater mines, road damage/degradation, and bridge interdiction on the movement rates and routing possibilities across the ADRG. First, programs must be written to define and store route movement networks and arrays of obstacles. For this paper, we will assume these programs already exist along with the avenues of approach or movement corridors that they help determine. Harrell [14],[15],[16],[17],[18] gives a partial description of some current techniques of doing this.

Similarly,in developing programs to compute network flow[b] rates which identify the critical nodes in the solution, it is important to compute the maximal or min-cost flows in terms of an ordered list of paths from the start node (or set of nodes) to the goal node (or set of nodes). The solution can be displayed just as a logic programming interpreter displays in turn the list of

[a] The maximal flow problem is the problem of determining what is the greatest number of vehicles/hour that can travel through a movement network at a given time. It is computed by designing an algorithm to optimize the assignments of flows to edges in the network. The min-cost flow problem is the problem of determining from all the possible flows which realize a given network maximal flow value, those which do it with minimal cost.

[b] An assignment of flow-rates to some or all of the edges in a given network. Each flow-rate has to be less than or equal to the flow-capacity of its edge.

predicate[c] variable identifications which satisfy the specified goal. The question then becomes whether this approach is feasible in terms of search time bounds and how it is implemented. This question is answered in the paper Harrell [17] which contains proofs of several theorems giving time bounds for the performance of the algorithm in terms of a polynomial formula involving the number of vertices and edges in the network. Some assumptions about the number of incoming and outgoing edges at any vertex are required in order to prove these theorems. But, in the situation in which the network is created interactively on the arc digitized raster graphic the user has the ability to model the representation of the situation in this manner.

This paper contains six sections. Section 1 is the introduction. In section 2 a short history of the development of the algorithm and the motivation for its use is given. In Section 3 an explanation of how this algorithm can be used to solve the maximum flow problems associated with certain types of networks is discussed. And, an example of its use to analyze Corps-level tactical off-road vehicular mobility in Germany. In section 4 simple modifications to this algorithm are presented that can be used to solve the corresponding min-cost flow problem. In section 5 there is a discussion of the appropriateness of these algorithms for solving transportation and assignment problems. An example of its application to analyze a theater level logistics problem in Korea is given. The final section contains a summary.

2. **History of the Problem.** The motivation for using network analysis for the problems discussed above follow from the inability of modelling the situation only in terms of a series of routes along parallel movement corridors. Using only individual mobility corridors it is harder to tell how the network throughput is affected by the cross-corridor movement possibilities. The complexity of the problem of the analysis is much more than that involved in such a simplistic approach to the problem. The min-cost network solution solves the problem of matching units to tasks in order to implement a logistics or obstacle emplacement plan. It allows the user to evaluate the effect of removing road segments or movement corridors from the overall routing. The first requirement to accomplish this is the determination of a road network or series of unit movement mobility corridors on the arc digitized raster graphic. A maximum flow network solution for off-road movement corridors allow determination of the delay effect of obstacles by measuring throughput with and without obstacles.[d] When an obstacle

---

[c] A logical relation which affects one or more objects or variables.

[d] In order to define this term, flow rates along the edges in the network are required. This refers to the number of vehicles/hour that can pass over a given edge. It can be calculated as [1/(time it takes a group of vehicles to traverse the edge)]*number of vehicles in the group. See the report Harrell [13]

plan is being analyzed and weapon sites are added, the min-cost network solution allows the determination of the overall effect on the moving units being exposed to enemy fire. This effect follows from the firing rates and line-of-site capabilities of the weapon which are emplaced in the defensive plan. For a logistics plan the importance of bomb damage repair to the accomplishment of the overall haul requirement can be evaluated. Some progress in understanding the tactical problem in this way was accomplished by planners in the US Department of Army Studies [7],[8],[9].

In 1986 computer programs were written in PROLOG[e] to do network maximal flow vehicle throughput and minimal cost routing. The algorithms were based on a depth-first search version of a networking computational procedure developed by Edmonds/Karp [10]. In 1989 these programs were put into an early interactive Turbo Pascal version of the U.S. Waterways Experiment Station (WES) Mobility Division's Condensed Army Mobility Modelling System (CAMMS). The use of the programs was demonstrated for then US Army V Corps area in Germany using mobility speed prediction and terrain data. Corps level tactical movement plans were analyzed and the results published in WES report GL-89-4 (Harrell [16]).Input from the U.S.Army Engineer School was used for obstacle breaching times and for a list of standard obstacles. In 1990 the algorithms were rewritten and put into an improved C language version of CAMMS. Mobility information for areas of the U.S. Army National Training Center (NTC) in California was collected and converted to arc digitized raster graphic format. Members of the engineer staff at the U.S. Army Fort Riley Kansas Mechanized Infantry Division were trained in its use. The computer programs were put in the Army's tactical command and control software (ATCCS) programs, taken in the back of a High Performance Multi-Purpose Mobility Wheeled Vehicle (HMMWV) to a mechanized division rotation training cycle at the NTC. In 1991 an Army Research Office paper,"A Logic Programming Approach to Network Flow Problems",(Harrell [14]) was written which determines the computational complexity of these versions of the max-flow, min-cost algorithms. From 1991 to 1994 the present version of the U.S. Army Engineer School Obstacle Planner Software (OPS) was written by personnel in the WES Mobility Systems division. These versions of OPS did not have network analysis capability. They had however an automated mobility corridor generation algorithm which used a unit movement raster grid version of an A* search algorithm (see the reference by Nilsson [16] and McKinley[19] for a description of how this search algorithm works). In 1995 a new C-language version of the network algorithm was written which has better memory allocation routes and other

---

for ways of defining vehicular and unit movement flow rates along edges in an arc digitized raster graph movement network.

[e] An artificial intelligence programming language which uses in its compiler or interpreter a backtracking, depth-first search algorithm in order to satisfy the goals of different logical predicates.

additional features. A C-code algorithm was written to determine, using a max-min metric, point streams of coordinate differences between mobility corridors. Algorithms were written to take unit weapon relative combat power scores from Fort Leavenworth war games and interpolate their values across a spatial area. These values were then used to provide an alternate way to generate mobility corridors which take account of weapon siting (through line-of-site count) and unit firepower scores. A preprocessor was written to filter out no-go cells from raster data, then automatically generate a grid movement network for routing and throughput analysis A breath-first path search version of the max-flow algorithm was written to improve the capacity of the network capable of being analyzed and speed of execution of different parts of the algorithm (this part of the algorithm now has the ability to use the method of pre-flows as explained in the reference Cormen [3]). C-language code was written to enable the min-cost version of the network algorithms to do unit/task scheduling and routing.

3. Methodology for Solving Maximal Flow problems and examples of network flow path routing and scheduling.

-- Off road movement corridors as they were in Germany 1988

-- Maximal flow throughput values can be used to measure obstacle effectiveness. This affects the capability of the defense to deal with attacking vehicles.

The network examples for this scenario is solved by a multiple start point, multiple end point, path search, goal seeking, backtracking[f] algorithm. The algorithm uses the depth-first search procedure explained above and extends partial paths by sorting non-cyclic routes at each new search stage. The report Harrell [16] lists the code in Prolog and Turbo Pascal for the algorithm. It solves both the max-flow and minimal cost throughput problem. Networks allow bi-directional edges. Unit/task resource scheduling problems can be solved by adding work time to the network nodes. The method differs from other depth-first search procedures in that it saves information about partial solution paths. This allows the same computer procedures to be used to solve both the max-flow and the min-cost flow problem.

Given that there is a method of generating all the shortest paths through the network, it is easy to modify the predicates to generate a list of maximal flow paths through the network. We now assume that each route segment has an associated maximal flow capacity. At each stage, simply choose the direction of maximal flow to expand the paths, and perform a sort on the maximal flow of the route segments instead of minimum length. Then, write a predicate that each time we reach the goal node with a route, go

[f] An algorithmic search scheme. In order to compute all the ways to satisfy a given goal it follows a series of branching paths. Then, upon exhausting future search possibilities,in order to start the search again, retraces its steps to the last previous branching decision.

back and subtract that routes' flow values from the network capacities. When the Prolog backtracking search doesn't generate any more solutions, all directed paths through the network have at least one edge which is already filled to capacity. One other way exists to increase the flow in the network. The paths which contain edges that point backward along the allowed route segments can be considered. Then, when the goal node is reached, proceed back to the source and modify the flow capacities, and add flow capacity along those segments, instead of subtracting it. As explained by Sedgewick [22] when the above procedure reaches a situation in which all paths have either full forward edges or nonempty back edges, then the Ford Fulkerson theorem says the maximal flow of the network has been reached. Many of the algorithms presently in use Goldberg [13] and Tarjan [24] for solving the maximal flow problem do not save lists of partial paths but instead use a labelling process to update information at each node. This increases computational speed, but makes it difficult to pick in order the main routes that contribute to the optimal solution. Since, it may be desirable to do sensitivity analyses which locate the points in the network which most affect all the possible solutions the above approach gives you more information after the process is completed. Thus, it is possible to print out in order of flow the paths which contributed to the max-min cut situation. This then can be used to plan barriers for the defense or attack routes for the offense. The description of the algorithm is illustrated below:

Step 1. Search for the best(maximal flow) route from the starting to the ending node. Only search in directions in which there is either a forward edge with positive unused flow capacity, or a backward edge with positive existing flow. If no route exists, terminate the algorithm.

Step 2. Subtract the value of that flow from the capacity slots in the route's definition and add the flows (or subtract the flows if headed in a backward direction along an edge). Go to step 1.

If flow rates are added to the edges the above procedure will generate the maximum vehicular flow across the network and compute the sensitivity of the solution to changes in flow rates at critical nodes. Off road flow capacities may be estimated from vehicular movement formations and speeds in the terrain corridor that each edge corresponds with. Suppose that the terrain will support a certain number of units and the movement speeds are computed from off-road vehicular speed prediction programs. A movement formation such as shown in figure 1 can be used and the vehicular flow rates computed according to the formula given above in footnote d. Then figure 2 shows a maximum flow solution and figure 3 shows the changes in the solution caused by changing the speeds on three edges.

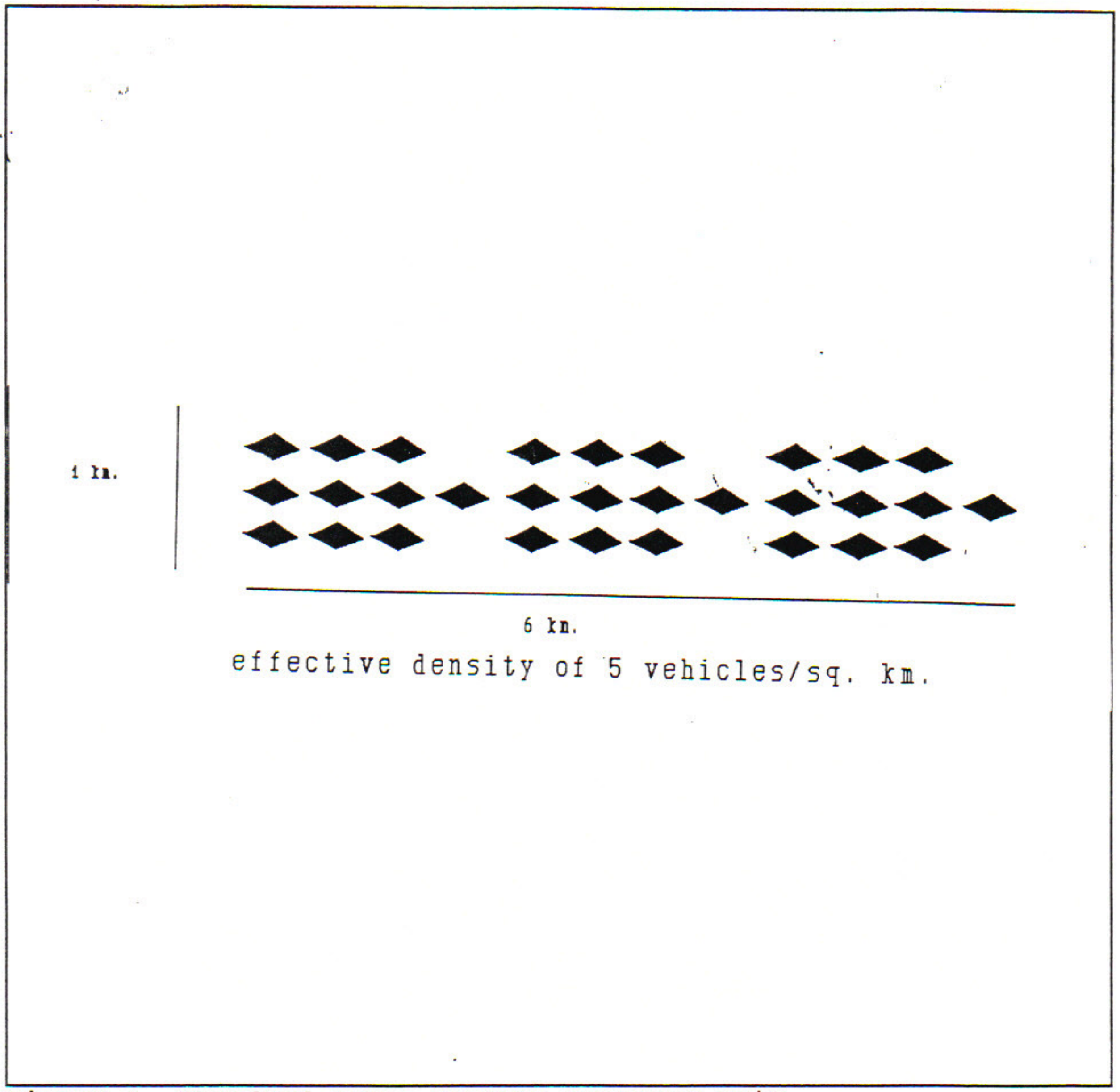

Figure 1. Standard cross country movement unit

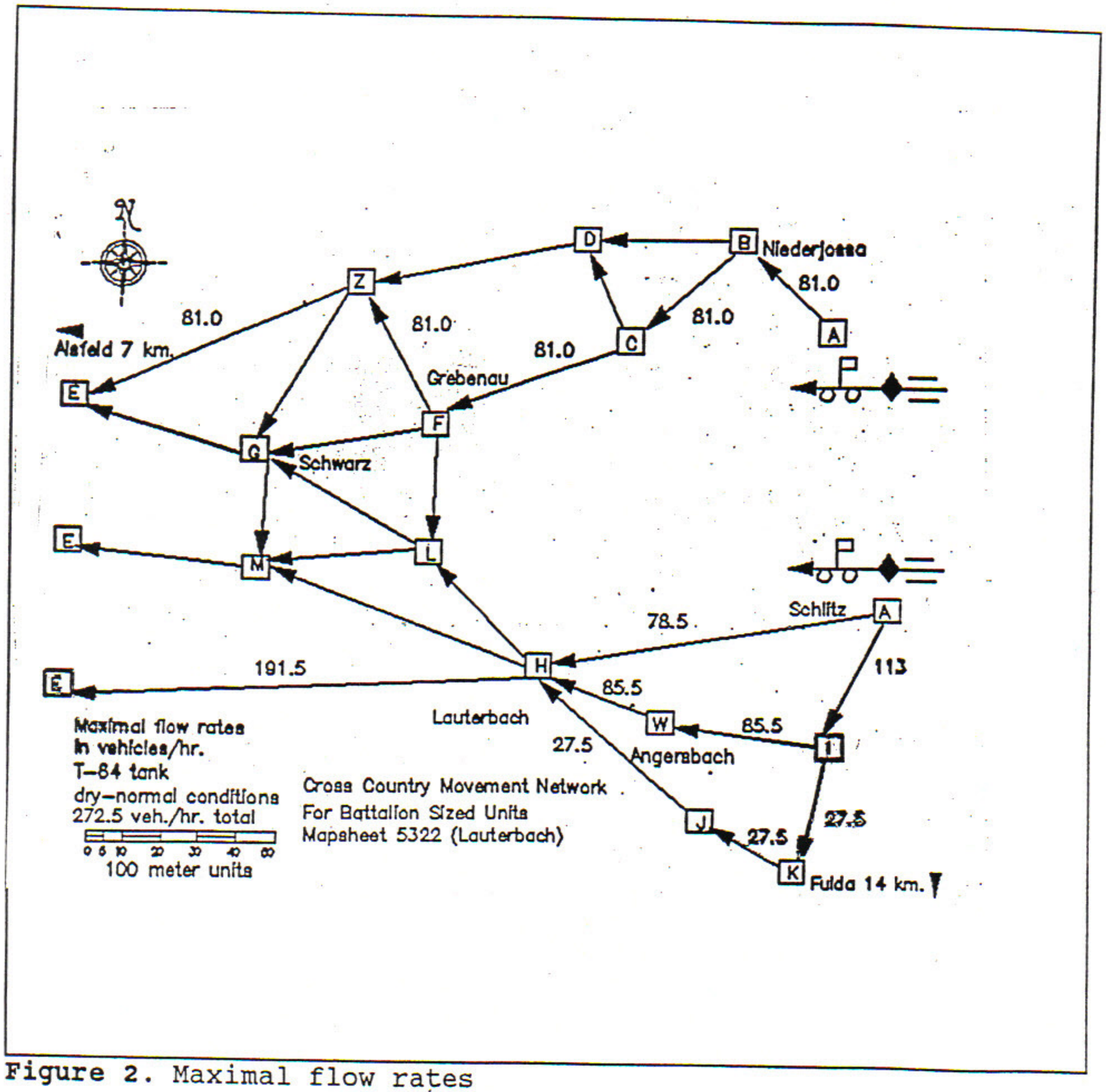

Figure 2. Maximal flow rates

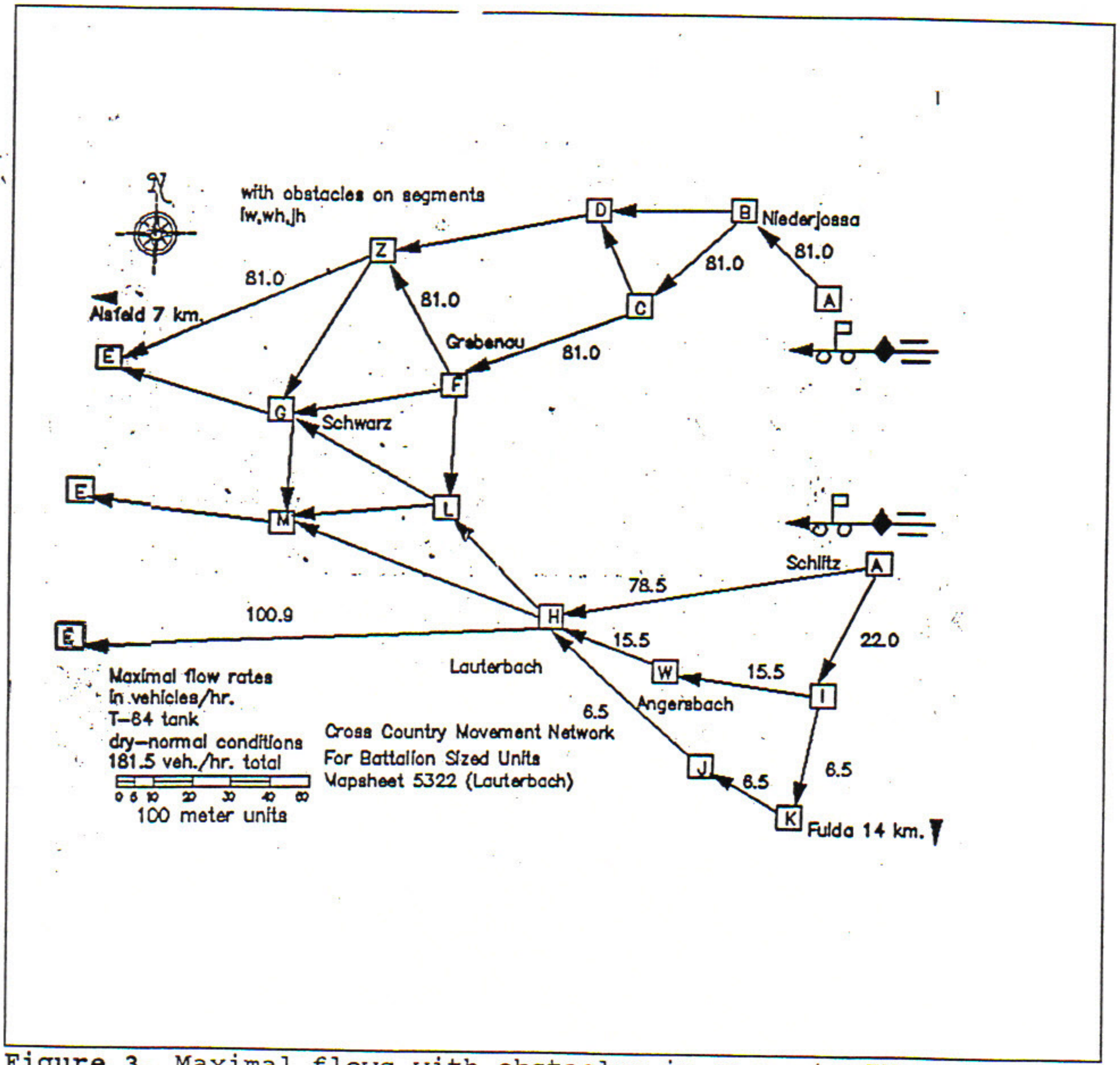

Figure 3. Maximal flows with obstacles in segments IW,WH,JH

Maximal flows without obstacles

```
MAXIMAL FLOWS

AIWHE flow veh/hr =   85.5
ABCFZE flow veh/hr =   81.0
AHE flow veh/hr =   78.5
AIKJHE flow veh/hr =   27.5
 total flow 272.5 veh./hr.
```

Maximal Flows with obstacles

```
MAXIMAL FLOWS

ABCFZE flow veh/hr =   81.0
AHE flow veh/hr =   78.5
AIWHE flow veh/hr =   15.5
AIKJHE flow veh/hr =    6.5
 total flow 181.5 veh./hr.
```

4. **Methodology for Solving the Minimal Cost Network Flow Problem** As a further benefit of the above approach, the procedures developed can be used to solve the minimal cost network flow problem. The minimal cost network flow problem is a generalization of the transportation network problem in operations research. In its formulation each edge is assumed to have a cost as well as a flow capacity associated. The problem then becomes to determine which of the several possible realizations of the maximal flow or throughput in the network has the associated minimal cost. For our purposes, the total cost of the flow will be considered as the sum of the flows along each edge multiplied by the traverse time. This measure of effectiveness is important in wargaming because it represents the amount of target exposure which is required for an offensive force to reach its objectives. As in the shortest path algorithm presented earlier, the algorithm that will be used to compute total cost of the flow expands the paths at each stage in the direction of shortest time. The resulting expanded path list will be sorted on time of the routes(in the

maximal flow algorithm the maximal flow possibilities were sorted on). Only those directions in which the maximum flow algorithm says there is either a forward edge with positive unused flow capacity, or a backward edge with positive existing flow should be chosen. As in the maximum flow algorithm, when the goal node is reached we proceed to go back to the source and subtract off the maximum flow that least cost incremental flow route can handle(in the maximal flow algorithm, the route is not necessarily the least cost incrementing flow).

The following theorem from Ford and Fulkerson [12, pages 121-122], which is also noted in Deo [6,page 394], insures that this process will generate the optimum solution.

Theorem: Let f be the minimal cost flow pattern of value w from start to finish. The flow pattern f' obtained by adding delta <= 0 to the flow in the forward edges of a minimal cost unsaturated path, and subtracting delta from the flow in the backward edges of the path is a minimal cost flow of value w + delta for the original network.

The same source code predicates can be used to implement this algorithm:

Step 1: Search for the best(minimal cost) route from the starting to the ending node. Only search in directions in which there is either a forward edge with positive unused flow capacity, or a backward edge with positive existing flow. If no such route exists, then terminate the algorithm.

Step 2: Subtract the value of that flow from the capacity slots in the route's definition and add the flows (or subtract the flows if going in a backward direction) along the edges. Go to Step 1.

As mentioned above the implementation of this algorithm uses the same basic predicates as the other two algorithms. Under the same assumptions used in the section on how to compute the maximal flow,the number of steps required to complete the algorithm is proportional to the number of vertices cubed.

If we again consider the network in figure 2, it is now possible to solve the problem of determining which of the several maximum flow solutions costs the less in the above sense.

Example of the method applied to a problem in NATO's defensive plan of Germany (1988).

Minimal cost flows for the same network and same vehicle/weather conditions as considered before are shown below in Figure 4. The maximum throughput for the minimal cost flows is the same as that which results from the maximal flow algorithm. The paths followed to achieve this throughput is, however, different in the minimal cost flows from those which result from running the maximal flow

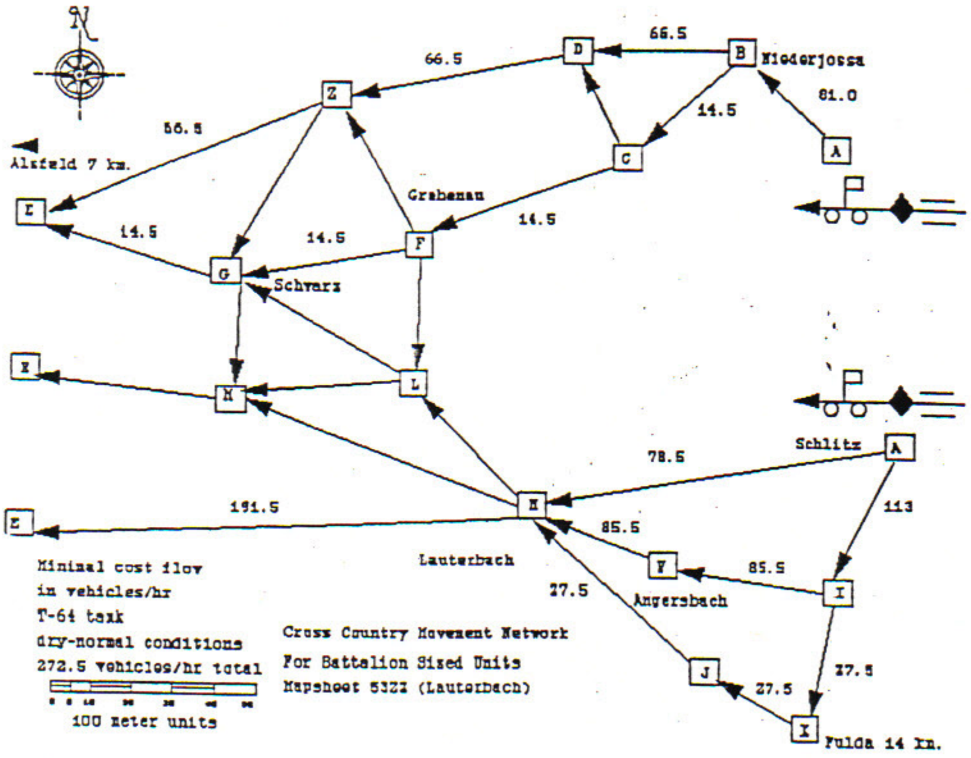

Figure 25. Minimal cost flows, dry-normal conditions

Figure 4. Minimal cost flows

algorithm. This is to be expected since in the minimal cost case the algorithm chooses the direction of shortest time to expand the search path. In the maximal flow case, the algorithm chooses the direction of maximum flow to expand the search paths. In the list of the minimum cost flow paths, we have included an average cost for each flow. This is defined to be the average of the sum of the amount of each flow path (vehicles) times the cost of it (minutes). Note that this average does not change much with and without the presence of obstacles. This is because what the obstacles affect (being employed over only a part of the network) is primarily the maximum throughput, and not the time through the network.

```
  MIN_COST FLOWS
          Minimum cost flows without obstacles


 AHE      flow veh/hr =     78.5 cost time(min)=    48.7
 AIWHE    flow veh/hr =     85.5 cost time(min)=    56.8
 AIKJHE   flow veh/hr =     27.5 cost time(min)=    66.2
 ABDZE    flow veh/hr =     66.5 cost time(min)=    67.2
 ABCFGE   flow veh/hr =     14.5 cost time(min)=    67.2

total flow 272.5 veh\hr. average cost 58.5 min\veh[g]
```

5. **Explanation of how this algorithm can be used to solve the general transportation network problem along with an example of its application to the logistics haul problem in a defensive plan of the Republic of Korea.** By the general transportation network problem the following is meant: N source points are located on a map as origins of logistical material. Each source has associated with it a supply of a[i] units of the material. In addition, there are m points designated as sinks or destinations, each destination point requires b[i] units of the material. Associated with each link in a network between the sources and the destinations there is a unit cost of transportation and a flow capacity. The problem is to determine the shipping or trucking pattern from origins to destinations that minimizes the total cost under the constraints imposed by the flow capacities on each link. When there are no more than one intermediate edges connecting the sources to the sinks the problem is the same as what is usually called in operations research the ordinary transportation problem or the assignment problem. Existing algorithms such as the Hungarian algorithm (Bazaraa, et.al. [3] and Ford and Fulkerson [12]) are adequate to solve the problem. And, the particular paths involved in scheduling or routing the solution follow immediately from the values of the

[g] This is the average of the second column of numbers. Each cost time is computed by dividing the flow path rate (vehicles/hr.) by the weighted sum of the number of vehicles in the formations for that flow rate. This is done by assigning widths (number of battalion sized units in formations as shown in figure 1) to the movement corridors (edges) in each flow path.

solution's edge flows. Also, in the case where there are intermediate edges but the scheduling or routing of the implementation of the min-cost solution is not required the Network Simplex Linear Programming approach is satisfactory. See the references Ahuja et. al. [1], Bazaraa et. al. [3], Ford and Fulkerson [9], and Bertsekes [8], Padberg [21] for the details of these approaches. This problem can be considered as a special case of the minimal cost network flow problem discussed in the previous section by defining n paths each with a flow capacity equal to a[i] from a starting point, and m paths each with a flow capacity equal to b[i] from the destinations to a notional ending points. The algorithm given to solve that problem will in the process of computing the maximal flow in the network just defined, produce the minimal shipping cost solution which satisfies most of the total requirements at the destinations. With simple modifications to the starting requirements for the search routines, the algorithm will produce solutions which satisfy the list of destinations in any given prioritized sequence.

Figure 5 shows a subset of a road network in Korea which was used as a supply network for a logistics exercise. Figure 6 shows the transportation network along with nodes,edges, and travel distances along the edges from a set of five ammunition supply points (source nodes 19,1,2,20,21,22) which are connected to three destinations (sink nodes 16,17,18). In Tables 1,2 logistics information about the network is given. The coordinates of the transportation network's sources and sinks are listed. The 5 Ton, 15 Ton, and 18 Ton truck hauling capacities for each route are also listed. These translate into network edge capacities for the minimal cost network problem. Vehicle speeds are assigned to each route. These translate into times to travel, which are the edge

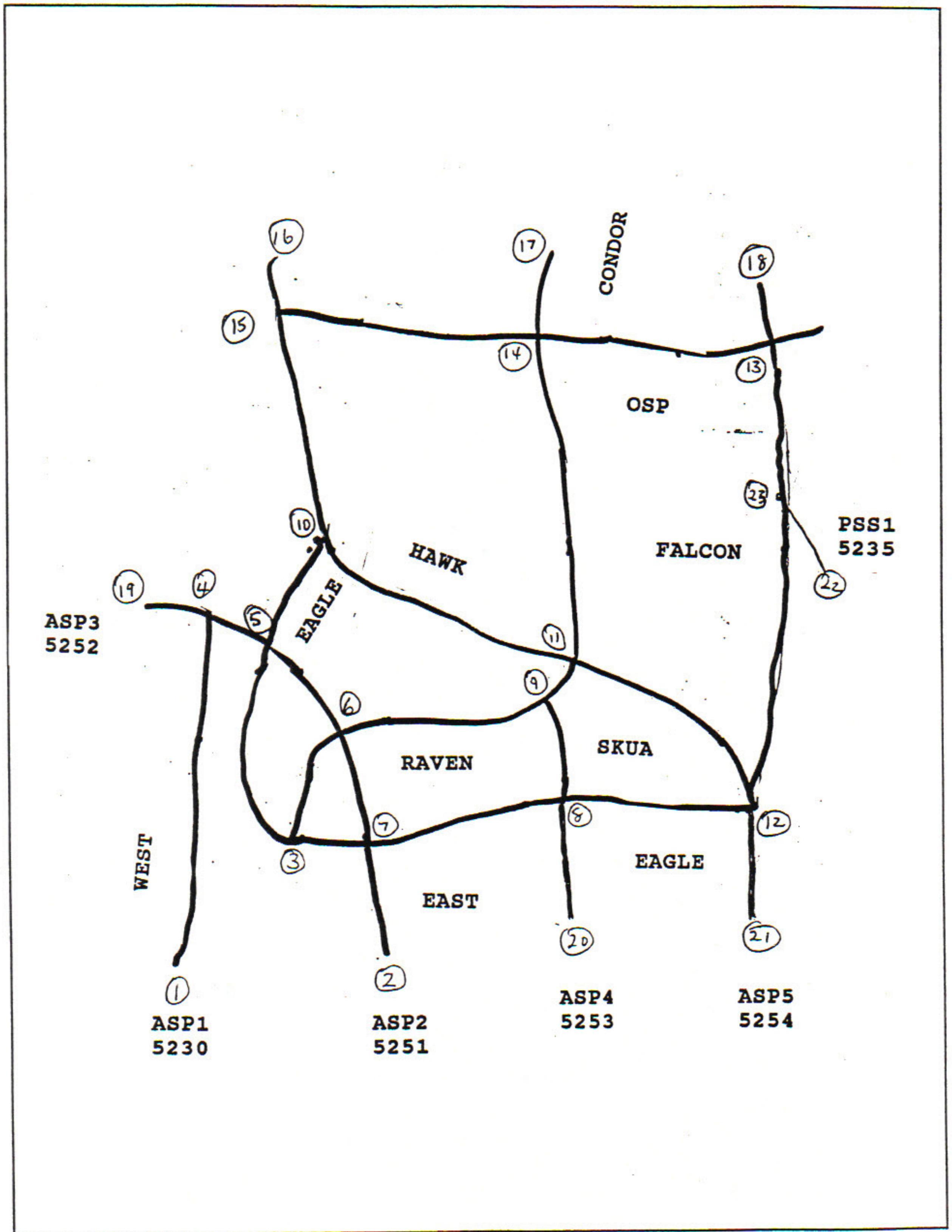

Figure 5 South Korea Transportation Network

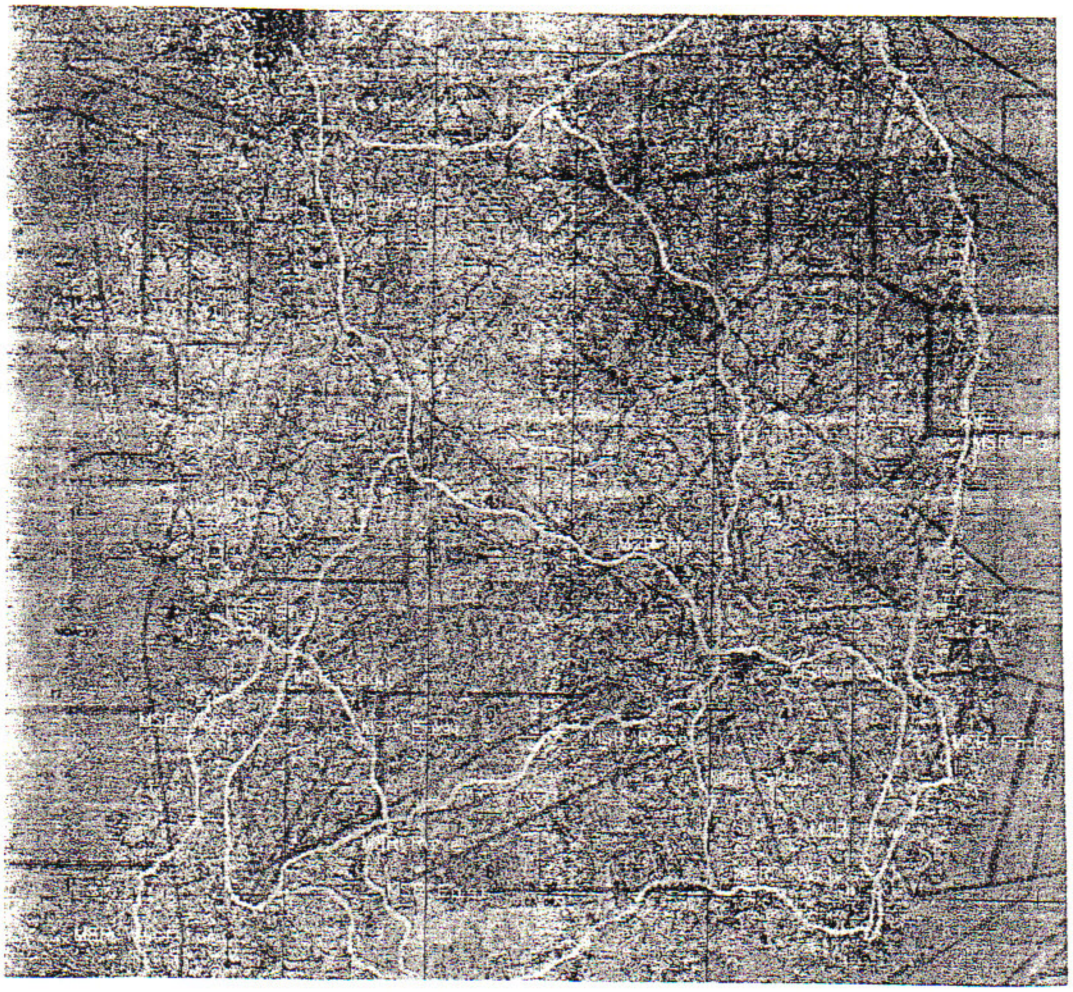

Figure 6 Arc Digitized Raster Graphic

Table 1

CLASS V STORAGE SITES		
Supply point number	Map coordinates	Name
5230	CP890770	ASP1
5251	CP890570	ASP2
5252	CP010925	ASP3
5253	CP030430	ASP4
5254	CP033440	ASP5
PORT STORAGE SITES		
5235	CP900455	PSS1

Table 2

Main Supply Route Capacity (1000 Short Tons/Day)						
Route	Length (km.)	5Ton	10Ton	18Ton	Speed (k.p.h)	Travel Time
Skua	63	92.1	165.8	257.9	50	75.6
Condor	198	76.7	138.1	214.8	50	237.6
Osprey	178	92.1	165.8	257.9	50	213.6
Raven	178	92.1	165.8	257.9	50	213.6
East	195	46	82.9	129.3	50	234
West	152	46	92.9	129.3	50	182.4
Eagle	402	46	82.9	129.3	50	482.4
Hawk	410	92.1	165.8	257.9	50	492
Falcon	450	46	82.9	129.3	50	540

costs for the minimal cost network problem. The results of the analysis are shown in Tables 3 and 4. The sinks for the routes are held constan~~c~~t and the sources varied inside of and between the two tables in order to investigate the best way of implementing the total logistics haul problem.

Table 3

Results of Logistic Haul Throughput Analysis for the Korean Theater		
5 Ton Trucks		
3 Nodes 16,17,18 are Sinks		
Sources	Total Throughput in 1000 Short Tons' (ST)	Best Routings [nodes] traverse time (mn)
2 21	92= 46 + 46	[21 12 11 14 17] 440 [2 7 6 9 11 14 17] 483
2 20 21	138= 46 + 46 + 46	[20 8 9 11 14 17] 428 [21 12 11 14 17] 440 [2 7 6 5 10 15 16] 486
1 2 22 20 21	184 = 46 + 46 + 46 + 46	[22 23 13 18] 320 [20 8 9 11 14 17] 428 [21 12 11 14 17] 440 [2 7 6 5 10 15 16] 486

Table 4

Results of the Logistics Haul Throughput Analysis for the Korean Theater		
10 Ton Trucks		
3 Nodes 16,17,17 are sinks		
Sources	Total Throughput 1000 Short Tons (ST)	Best Routings [nodes] traverse time (mn)
2 22	166 = 83 + 83	[22 23 13 18] 320 [2 7 6 9 11 14 17] 483
2 20 22	249 = 83 + 83 + 55 + 28	[22 23 13 18] 320 [20 8 9 11 14 17] 428 [2 7 6 9 11 14 17] 483 [2 7 6 5 10 15 16 486
1 2 22 20 21	387 = 83 + 83 + 55 + 83 + 28 + 55	[22 23 13 18] 320 [20 8 9 11 14 17] 428 [21 12 11 14 17] 440 [2 7 6 5 10 15 16] 486 [21 12 11 10 15 16] 564 [1 4 5 6 9 11 10 15 16] 750

The best routings are listed for two, three, and five source nodes and the same three sink nodes. Total throughput is given along which the best routing solutions which construct this throughput. The routings contain a list of the nodes traversed from source to sink and are listed in order of traverse time. The cost measure of effectiveness (the sum of the edge travel times multiplied by the flow rates) are listed under the traverse times.

6. **Summary** We have given definitions and examples of some scheduling and network terminology. We have explained the motivation in terms of the particular application areas discussed of using the particular algorithms we have selected. Along with this we have discussed several ways in which sorted priority queue[h] depth-first searches can be used to solve shortest path,network maximal flow, and min-cost network flow problems. Then we have demonstrated the usefulness of these approaches in terms of the several specific military planning applications. More details of how to implement these algorithms and examples of some of the C and Prolog language code used are contained in the references cited in the earlier sections. Since the logical structure of the implementation of the coding of these algorithms arise in many different research areas (such as expert system design, geographical information system searches, and communications network routing) there should be several other uses in the future of these type of artificial intelligence networking algorithms.

Acknowledgements The tests described and the resulting data presented herein, unless otherwise noted were obtained from research conducted under the MILITARY RESEARCH DEVELOPMENT TEST AND EVALUATION PROGRAM of the United States Army Corps of Engineers by the U.S. Army Waterways Experiment Station. Permission was granted by the Chief of Engineers to publish this information. I wish to thank Mr. Newell R. Murphy, Jr. and Mr. David Horner for their valuable comments and suggestions which have improved the exposition of this paper.

[h] A queue which is sorted after each new element enters it and which then places the resulting first element at the top (head) of the queue..

BIBLIOGRAPHY

[1] Ahuja, R.K., Magnanti , T. L., Orlin, J.B.,1993 Network Flows, John Wiley and Sons, Prentiss Hall, New York.

[2] Borlund Int.,Turbo Prolog,Version 2.0,User's Guide,, 1988, Scotts Valley, CA.

[3] Bazaraa, M.S. and Jarvis,1977, J.J. Linear Programming and Network Flows, John Wiley and Sons, New York.

[4] Bratko, Ivan,Prolog Programming for Artificial Intelligence,1986, Addison Wesley Publishing Co.

[5] Cormen,T.H, Leiserson, C.E., and Rivest, R. L., 1990, Algorithms, MIT Press, Cambridge, MA.

[6] Deo,Narsingh,Graph Thory with Applications to Engineering and Computer Science.,1974,Prentice Hall,Englewood Cliffs,NJ.

[7] Department of the Army, US Army Engineer Studies Center,1981, "Update of an Assessment of the Engineer Plan in Support of V Corps Oplan 33001", Secret,Washington DC.

[8] Department of the Army,1983, US Army Engineer Studies Center, "Middle East Targeting Assessment", Secret-Noforn, Washington, DC.

[9] Department of the Army, 1982, US Army Engineer Studies Center, "An Evaluation of the Adequacy of the Obstacle Plan Suppporting CINCUNC/CFC OPLAN 5027", Secret-Noforn Except ROK.

[10] Edmonds,J. and Karp,R.M.,"Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems", Journal for the Association of Computing Machinery, Volume 19, number 2,pp.248-264, New York, N.Y.

[11] Bertsekes, D.P.,1991, Linear Network Optimization,MIT Press, Cambridge, MA.

[12] Ford,L.R. and Fulkerson,D.R.,1962,Flows in Networks, Princeton University Press, Princeton, N.J.

[13] Goldberg,A.E. and Tarjan,R.E.,1988,"A New Approach to the Maximum-Flow Problem",Journal for the Association of Computing Machinery, volume 35,number 4,pp. 921-940,New York, N.Y.

[14] Harrell,A.W.,1988,"The Concept of Individual Vehicular and Unit Mobility and its Effect on Wargaming",Proceedings 27th Army Operations Research Symposium,Volume II,pp. 3-993 to 3-1006.

[15] Harrell,A.W.,1989,"The Concept of Individual Vehicular and Unit Mobility and its Effect on Wargaming",Proceedings 56th Military Operations Research Symposium.

[16] Harrell,A.W.,1989,"Evaluating the Effect of Off-Road Obstacles on Unit Movement",TR-GL-89-4,Waterways Experiment Station,Vicksburg, MS.

[17] Harrell, A.W.,1991,"A Logic Programming Approach to Network Flow Algorithms", Army Research Office Report number 91-1,Transactions of the Eight Army Conference on Applied Mathematics and Computing.

[18] Harrell, A.W.,1996, Examples of Network Routing and Unit/Task Scheduling Problems, 5th S.I.A.M. Conference on Optimization, Society for Industrial and Applied Mathematics.

[19] McKinley, G.B.,1990, "Automated Selection and Evaluation of Avenues of Approach", TR-GL-90-5,Waterways Experiment Station, Vicksburg, MS.

[20] Nilsson, N.J., 1980,"Principles of Artificial Intelligence",Tioga Publishing Co., Palo Alto, CA.

[21] Padberg, M, 1995,Linear Optimization and Extensions,Springer-Verlag, Berlin-Heidelberg.

[22] Sedgewick, Robert, 1993, Algorithms, Addison Wesley, Reading MA.

[23] Sterling,Leon and Shapiro,Ehud, 1986,The Art of Prolog, The MIT Press, Cambridge, MA.

[24] Tarjan,R.E.,1983,Data Structures and Network Algorithms, Society for Industrial and Applied Mathematics, Philadephia, PA.

[25] Winston, Patrick Henry,1984, Artificial Intelligence 2nd ed.,Addison Wesley, Reading MA.

[26] ......,1984, Lisp 2nd ed., Reading MA.