THE 'OPTIMAL' ROUTING OF VEHICLES USING THE GASP II SIMULATOR

M. Wayne Shiveley [1]

Edward M. Salley [2]

[1] Lehigh University

[2] Western Electric Co.

ABSTRACT

An integrated systems approach has been used to formulate a model for the analysis of routes and the routing of vehicles. The essential data to describe the available routes are stored in the form of a network; the directional relationships and the relative location of the routes are established with the aid of a grid. The GASP II simulator has been used to develop the model. The model utilizes the network to optimally route a vehicle from a specified starting point to a given destination; it employs a variety of techniques which includes dynamic programming, branch and bound, and simulation.

I. INTRODUCTION

There are numerous different types of vehicle routing problems; the problem that is described and solved in this report is one of the simplest types of routing, the identification of the shortest round trip between two points where restrictions are imposed on specified paths. The proposed approach consolidates many of the more sophisticated techniques into a simple basic procedure which can be easily understood and applied. The solution technique used is simulation; however, the size of the problem is controlled by the concept of branch and bound and the optimality of the solution can be proven by illustrating that the solution procedure is in fact identical to that of dynamic programming (1). The basic model was originally conceived because there existed a need to develop fast and efficient dispatching algorithms for large networks (3).

II. SOLUTION APPROACH

In order to provide a convenient method to identify the various routes and the restrictions on the use of each route, a network was developed. Each intersection of two or more routes is given a unique node number. The individual paths in the network can be uniquely identified by two node numbers. Depending on the direction being travelled, one node is a predecessor node and the other node is a successor; the node being travelled to is considered the successor node and the origin of the path is the predecessor node. There is no restriction placed on the value assigned to a node except that it must be unique. In addition to node numbers, each path must maintain pertinent related information which includes the length of the path and weight limitation on the path.

III. SIMULATION PROGRAM

In order to program the procedure a number of ordered sets of related pieces of information must be updated and maintained during the process. Each of these sets of information were treated as a file of information. The GASP II simulation language is a Fortran based language which supports the searching, reading, ordering, and updating of small files of information. The language provided the essential tools necessary to program the procedure. A detailed explanation of the model will not be attempted; however, the pertinent features of the algorithm are outlined.

The proposed algorithm can be described in eleven steps:

1. Read in the data necessary to define the network including a positive identifier for the starting and ending nodes.

2. Load into File 1 all paths in the network who have a predecessor node that coincides with the starting node; these entries are ordered based on the total distance from the starting node to this path's successor node.

3. Remove the first path (shortest total distance including this path) from File 1 if it is a feasible route then branch to step 5.

4. If there are still entries in File 1 go to step 3; otherwise go to step 8.

5. Store the path just removed from File 1 into File 2; File 2 contains the feasible and only the feasible routes.

6. If the path just stored in File 2 has a successor node equal to the destination node then branch to step 9.

7. Remove all entries from File 3 whose predecessor node is equivalent to the previous entry (stored in File 2) successor node; update the total distance travelled to reach this path and then store this entry in File 1, then branch to step 4.

8. Print a message which informs the user that there is no feasible routes to the ending node and then terminate the procedure.

9. Read from the list of paths in File 2 the first entry whose path's successor node is equal to the last path's predecessor node. Store this path so that it can be printed later.

10. If this new path's (from File 2) predecessor node is not the original starting node branch to step 9.

11. Print the paths stored in the order of their cumulative distance; this completes the solution.

IV. CONCLUSIONS

The solution to a simple problem involving the optimal routing of a vehicle from one point to another was found through the use of simulation. The same technique can be applied to problems where the map of a city is stored in machine readable form (4). It provides a feasible method to route ambulances and/or fire engines throughout a city or township. The optimal route can be identified with any number of constraints placed on the individual paths within the network.

BIBLIOGRAPHY

1. Bellman, R. E. Dynamic Programming, Princeton University Press, New Jersey, 1957.
2. Pritsker, A. Alan B., Kiviat, Philip J. Simulation with Gasp II, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969.
3. Ross, D. W. "Traffic Control and Highway Networks," Networks, 2:97-123, John Wiley & Sons, Inc., 1972.
4. Shiveley, M. W. "Computerized Storage and Analysis of Networks," a paper presented at the ORSA Convention (Spring) 1972.