The Two-Snowplow Routing Problem

Jonathan D. Robinson
Lester S. Ogawa
Scott G. Frickenstein
US Air Force Academy
Colorado Springs, CO 80840
Advisor: W.E. Skeith

Summary

We develop an efficient use of two trucks to sweep snow from roads in Wicomico County, MD. After making assumptions concerning specifics about the snowplows and the snow-removal district, we determine the best routes by using heuristic methods known as the Cluster-First, Route-Second method and the Route-First, Cluster-Second method. We implement these methods on a personal computer in conjunction with a depth-first search algorithm to generate Euler tours.

We then compare the results for six different general model formulations using the above heuristic, and make recommendations. We offer a short analysis of the model and algorithm, in which we discuss the strengths and weaknesses of our solution.

Snowplow Assumptions

We assume that the snowplows used in Wicomico County are similar to those used in Monument, CO. They vary in size, with the largest being allocated to the interstate and state highways, while smaller snowplows are used on smaller streets and roads. Once the highways are cleared, the larger snowplows assist with the smaller streets, as these tend to take longer due to the turns and reduced speed. To simplify the situation, we assume that the snowplows in Wicomico County are the small snowplows. This assumption is critical, in that the small snowplows can perform U-turns easily, while the larger snowplows cannot.

In talking to a snow plow operator, we learned that the speed of snowplows varies with the conditions. However, we assume that both snowplows travel at a constant velocity whether they are plowing or not. We also assume that the snowplows never break down and never get stuck.

We also assume the snowplows travel in the direction of traffic and that turning or going straight through an intersection takes the same amount of time.

Locale Assumptions

We assume that the snow-removal district will not be receiving more snowplows nor constructing more roads in the foreseeable future. This assumption limits the scope to a one-time analysis.

While the state highways may be used by the snowplows to get from point to point, the highways do not have to be plowed.

We assume that the map provided is accurate in both road inclusions and scale; and as no streets are labelled as having priority, we assume equal aggregate demand for all roads in the snow-removal district.

For lack of information, we assume that parking and other plowing obstructions are either disallowed totally or regulated so that no obstructions are present while the roads are being plowed. Similarly, we assume that time loss due to traffic flow control (i.e., stop signs and stoplights) is not a factor.

Lastly, we assume that the administration and personnel responsible for the snow removal do not have any preference for a specific routing policy.

Problem Overview and Analysis

The format of this problem can be simplified into a network of nodes and arcs; nodes represent the intersections or endpoints of the streets, while arcs represent the lanes of the streets between the intersections. Problems of this format are categorized as general routing problems or networks.

Servicing each road in the snow-removal district with multiple snowplows is equivalent to traversing each arc of a subnetwork of the entire network at least once. Therefore, multiple snowplow routing falls under the category of the *rural postman problem* [Orloff 1974a, 35]. In our problem, we are given two snowplows that plow a subset of the entire network.

In order to solve the rural postman problem, we must first define what we mean by efficient. As we would like to avoid the issue of time, we interpret an efficient route to be one that minimizes deadheading distance. *Deadheading distance* [Bodin and Kursh 1978] is the cumulative distance that the snowplows are not actually plowing snow from the county roads. This means that backtracking, using state highways, and returning to the origin from the termination point of the plowing routes are all considered deadheading. As distance is our measure of effectiveness, accuracy and precision are very important. In order to minimize inaccuracies, we digitized the problem map using AutoCAD, a computer-aided design system, to measure the distances along each arc.

Given our definition of efficiency, if it were possible to create a route for each vehicle which never backtracks and eventually returns to the vehicle's origin, such a route would be the most efficient (i.e., optimal) way to use the two trucks.

However, since a snowplow has to perform U-turns at some intersections, we must also take into account practical considerations, such as safety factors, which could jeopardize the implementation of an efficient route. For example, we want to minimize the number of U-turns at intersections of county roads and state highways, to minimize the hazard of the plow being hit by a fast-moving car while completing the turn. Furthermore, depending on whether the state highways have cloverleaf interchanges or overpasses, U-turns may be extremely difficult. On the other hand, U-turns on county roads can be done easily at intersections, due to flow control, especially in early morning or late evening.

We would also like the snowplows to finish at the same time. Otherwise, time inefficiency and inequity would result.

Therefore, in addition to a route being efficient in terms of deadheading, we must also try to minimize the number of U-turns that take place at intersections of the state highways and county roads, while also striving to keep the route distances as equal as possible.

Our objective, then, becomes the determination of an efficient, practical, and equitable routing plan for the two snowplows in the snow-removal district of Wicomico County.

Because no exact algorithm exists for the two-vehicle postman problem, we must rely on heuristic methods to get a near optimal solution. The two most popular heuristic methods are referred to as "Route-First, Cluster-Second" and "Cluster-First, Route-Second" [Bodin and Kursh 1978, 526]. The first model determines the most efficient route for the entire network (the *Chinese postman problem*) and then breaks the route up into N routes, one for each of the N vehicles. The second heuristic approach uses prior criteria to determine the subset of arcs to assign to each vehicle and then finds the most efficient routes for each of the M nonintersecting subsets.

As noted, the most efficient route is one that would cover every road in each direction exactly once and end up at the starting point. A route that meets the first criterion is referred to in graph theory as an Euler tour. An *Euler tour* is formally defined as "a cycle which traverses each arc of a graph once and only once" [Orloff 1974a, 45]. We interpret our network as a (connected) graph with intersections, dead ends, and garages corresponding to nodes, and directed lanes of travel corresponding to directed arcs. Each node has as many entering arcs as departing arcs, as all the roads are two-lane roads.

According to Theorem 3 of Orloff [1974a, 45], these observations guarantee the existence of an Euler tour of the graph. Thus, the problem becomes one of finding an Euler tour that ends at the origin, as this oute would have no deadheading at all.

One way to identify an Euler tour that ends where it begins is to enumerate the possible combinations until we find one. Before pursuing this option, however, we calculate a lower bound on the number of possible Euler tours of our graph. Harary [1969, 204] gives a formula for the total number T of Euler tours of P nodes:

$$T = c\prod_{i=1}^{P}(d_i - 1)!,$$

where

c is the common value of all cofactors in the matrix Mod (described in [Harary 1969]) and

d_i is the number of points adjacent to node v_i.

For our graph, c is nonnegative, as the number of trails must be nonnegative. But c cannot equal zero, as this would imply isolated points in the graph. As our graph is not disconnected, we have $c \geq 1$. Thus, we can use $c = 1$ to calculate a lower bound on the number of Euler tours.

For the Route-First, Cluster-Second method, we find that the number of points with $d_i = k$ is as follows:

$$\begin{array}{lr} k = 1: & 8 \\ k = 2: & 5 \\ k = 3: & 101 \\ k = 4: & 24 \\ k = 5: & 1 \end{array}$$

Thus, a lower bound on the number of possible Euler tours is

$$\begin{aligned} T &\geq (0!)8(1!)5(2!)101(3!)24(4!)1 \\ &\geq 2.8 \times 10^7. \end{aligned}$$

A comparable number of tours exist for the Cluster-First, Route-Second method, due to the different combinations of bisecting the network.

Due to this overwhelming number of possible tours, we must use another method to find an Euler route that terminates at its origin. Then, once we have determined the Euler tours, we can then focus upon trying to construct routes that are practical and equitable.

Model Formulation

The model of this network is a connected digraph with 140 nodes and 424 directed arcs. Formally, *connected* means that for any two points in the network, we can start at one point and choose a path to the other, and vice-versa. A *digraph* (directed graph) is defined as a graph in which all the arcs have a direction associated with them.

Mathematically, there exist several representations of network problems of this sort which would put the "picture" into a more computational format. In general, networks are converted into matrices. Information concerning arcs and nodes differs depending on the application. As the arcs are the most significant entities in the postman problem, we used a variation of the origin-termination matrix representation [Jensen and Barnes 1980, 93]. Unlike the true origin-termination matrix, we use a list format, since our network is neither dense nor complete. In addition, we include the length of the arc as well as the U-turn flag, since these are necessary for comparison of equitability and practicality.

The model can be simplified by recognizing that since state highways do not have to be plowed, we can remove them from the network. Initially, then, our model assumes no use of state highways, which reduces the model to 139 nodes and 374 directed arcs. We later relax this constraint by adding deadhead branches to form an H-network [Bodin and Kursh 1978, 528].

At the heart of our algorithm is the search for an Euler tour that terminates at its origin. One method of finding such an Euler tour (when one exists) is the depth-first search algorithm (Trémaux's algorithm [Even 1979, 53–54]). The depth-first search uses a simple set of rules to select arcs for travel which will cover every arc in the network only once and terminate at the origin. The rules are:

1. Initially, all arcs are unmarked. Set $V \leftarrow S$ (the vehicle's starting point).
2. If there are no unmarked arcs leaving V, go to Step 4.
3. Choose an unmarked arc leaving V, mark it with "1", and traverse it to its other endpoint U.
   (a) If any arc leaving U is marked, traverse the arc that returns to V and go to Step 2.
   (b) If no arc leaving U is marked, re-mark the arc from V to U with "2", set $V \leftarrow U$, and go to Step 2.
4. If no arc leaving V is marked with "2", halt.
5. Traverse an arc leaving V which is marked with "1" to its other endpoint U. Set $V \leftarrow U$, and go to Step 3.

The depth-first search is an excellent algorithm for our model, as it is easily coded for a computer, and it can be used directly in conjunction with the origin-termination representation list we used to format our network. Appendices contain the origin-termination matrix, the node to node distances, and the program listing. [EDITOR'S NOTE: Omitted for space reasons.]

After formulating the algorithm, we tested it on a small subset of our graph, using only one vehicle, before using the algorithm on our entire graph. The search was then validated by tracing the complete tour for one vehicle plowing the entire region of Wicomico County. The original network was then divided into two subnetworks, each to be plowed by a different vehicle. In dividing the network, we favored a horizontal split, as each subnetwork could be easily accessed by the snowplows. We used the "eyeball method" to divide the network, attempting to equalize the distance travelled by each plow while minimizing the number of U-turns.

Results

In this section we discuss each of the six trial runs, three each from Route-First, Cluster-Second and from Cluster-First, Route-Second. Each of the Route-First, Cluster-Second runs begins with one of the trucks at one of the starting points and continues until this truck is at the second starting point. Here the second truck takes over and finishes the Euler tour (in reality, both trucks would start at the same time).

The objective of runs 1 and 4 is to clear all county roads, including all state roads in the Euler tour. Runs 2 and 5 clear only county roads and reflect the case of allowing U-turns at intersections of the state highways and the county roads. Runs 3 and 6 clear all county roads and use some state roads in order to minimize the number of U-turns.

For comparison, **Table 1** lists relevant information concerning each run.

Table 1.
Comparison of various routes.

Route	Total Legs	Mileage (miles)			U-turns		
		Path A	Path B	Total	Path A	Path B	Total
1A	424	210	106	316	63	19	82
1B	424	269	47	316	74	8	82
2A	374	151	109	260	54	27	81
2B	374	230	30	260	72	9	81
3A	398	6	278	284	0	71	71
3B	398	250	34	284	65	6	71
4	424	156	160	316	39	40	79
5	374	129	131	260	38	39	77
6	396	143	141	284	32	34	66

Analysis of Results

To analyze the data, we compare the efficiency, practicality, and equitability of each of the six trial runs. The main indicators are deadheading time, number of U-turns at state highways, and the distance travelled by each snowplow.

- **Runs 1–3:** The first six rows in Table 1 are the result of the Route-First, Cluster-Second heuristic. In general, the table shows that this method yields a higher number of U-turns and uneven mileage for the two trucks. These runs are impractical because many of the U-turns occur at state highways, and it is not equitable for one operator to plow many more miles than the other (the snow may melt by the time all plowing is done!). Given the time constraints and limited computing capability, it was impractical to try to find a more equitable plowing method using the depth-first search.

- **Runs 4–6:** These three runs result from the Cluster-First, Route-Second heuristic. In general, they are very equitable, as the distance division was made prior to the depth first search. Also, these runs finish at their origin and are much easier to implement than runs 1 through 3, as their region of service is distinct. Run 4 was ruled out due to its high mileage and high number of U-turns (especially on the state roads). Run 5 has the least mileage of the remaining two, so it is most efficient in terms of deadheading. Run 5 has roughly the same number of U-turns as Run 4, but more are on the state highways, making Run 5 more dangerous. Run 6 has moderate mileage compared to Runs 4 and 5, yet has significantly fewer U-turns. Additionally, only one of these U-turns is on a state road. In conclusion, Run 5 is most efficient (and therefore optimum) from the standpoint of deadheading (as no state roads are traversed), and Run 6 is most practical for the plow operator. Runs 5 and 6 are also equitable in terms of the distance travelled by each operator.

Therefore, the best heuristic routing plan is given by Run 5 if U-turns on the state highways are not a factor; if U-turns on the highway are a factor and should be avoided, then Run 6 is optimal. The specific routes and maps which describe the routing for Runs 5 and 6 are shown in **Figures 1** and **2**.

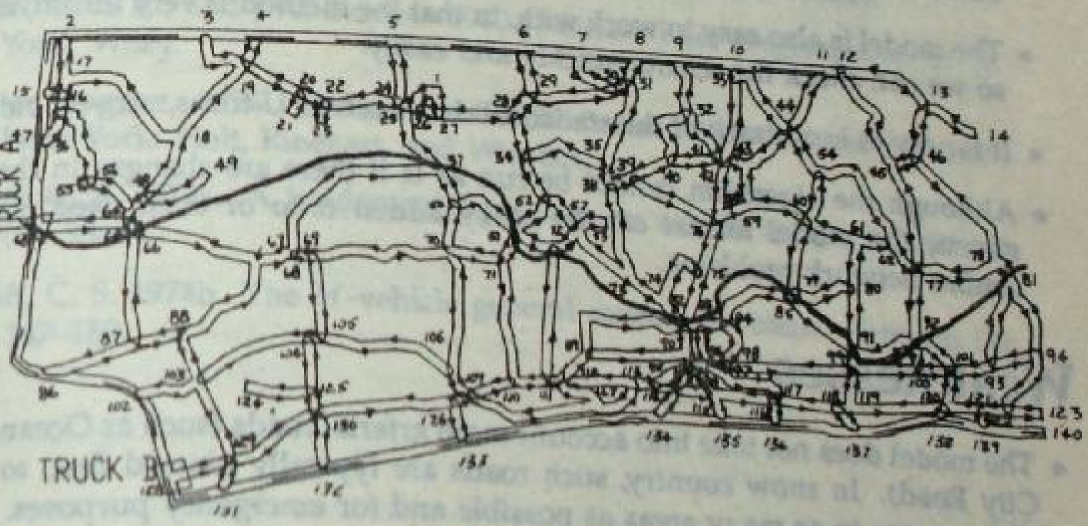

Figure 1. The most efficient route found.

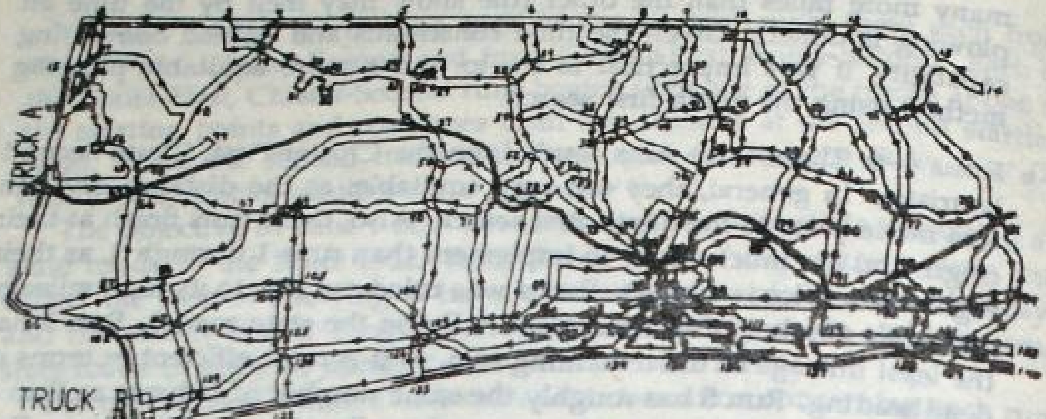

Figure 2. The most practical route found.

Strengths

- The model implements an algorithm that guarantees an efficient solution (no deadheading time). By definition of an Euler tour, there are no overlaps, and each snowplow finishes at its origin.
- One of the key strengths of the model is that it will always generate a solution that is efficient in terms of deadheading.
- The model is also easy to work with, in that the method is very intuitive, so we can check the results quickly and easily.
- It is easy to keep track of the statistical data such as U-turns, moves, etc.
- Although the algorithm cannot be run as is if there are changes in the streets, the model format can be easily added onto or even used for similar network problems.

Weaknesses

- The model does not take into account main arterial roads (such as Ocean City Road). In snow country, such roads are typically plowed first, to allow access to as many areas as possible and for emergency purposes.
- The model does not take into account traffic on the roads (which may impede plowing), stop signs/lights, or other possible unforeseen obstructions. Additionally, some roads may be inherently slower than others,

thereby failing our assumption of constant speed. In other words, our model assumes fairly ideal conditions! Thus, our solution may not, in reality, represent the fastest solution.

- A reanalysis would be necessary if the number of snowplows or roads changed.
- The depth-first search inherently incorporates many U-turns into the Euler tour. The algorithm is also not effective using the Route-First, Cluster-Second heuristic. This limits us to the Cluster-First, Route-Second heuristic method only.
- The model is very sensitive to how one bisects the network.

Conclusion

For near-ideal conditions, the model is suitable; for regular use, a more robust model is desired.

References

Bodin and Kursh. 1978. The sweeper routing problem. *Operations Research* 26(4): 525–537.

Even, Shimon. 1979. *Graph Algorithms*. Potomac, MD: Computer Science Press.

Harary, Frank. 1969. *Graph Theory*. Reading, MA: Addison-Wesley.

Jensen, Paul A., and J.W. Barnes. 1980. *Network Flow Programming*. New York: Wiley.

Lawler, Eugene. 1976. *Combinatorial Optimization: Networks and Matroids*. New York: Holt, Rinehart, and Winston.

Orloff, C. S. 1974a. A fundamental problem in vehicle routing. *Networks* (4): 35–64.

Orloff, C. S. 1974b. The *M*-vehicle general routing problem. *Networks* (4): 147–162.