Application of Routing Technologies to Rural Snow and Ice Control

EDWARD HASLAM AND JEFF R. WRIGHT

The design of routes for intrastate highway snow and ice control is perhaps the most difficult and complex of all public-sector routing problems. In addition to the random and usually unevenly distributed effects of a snow event, service must be provided rapidly, equitably, and simultaneously across the network. The task is made more difficult by the presence of multiple and conflicting objectives on the part of maintenance engineers responsible for this service. The design of snow removal routes is addressed from the perspective of multiple objective optimization. The strengths and weaknesses of several mathematical programming approaches are discussed and an efficient heuristic routing methodology is proposed. Experience with the analysis of a portion of the Indiana highway network is described.

Many public services are provided along predetermined routes with the overall effectiveness of those activities determined, at least in large part, by the efficiency of those routes. The design of service routes is characterized by multiple and conflicting objectives, many of which are difficult to quantify. Perhaps the most complex service activity for which careful route planning is important in wintertime snow and ice control on our nation's intrastate highway system.

Snow and ice control during the winter months in northern states is a major operation. The Indiana Department of Highways (INDOT), for example, must routinely maintain some 11,414 mi of roadway throughout the state. Because each traffic lane must receive service, this requirement translates into more than 29,000 lane-mi overall. The resources needed for this operation include nearly 1,500 trained personnel and some 1,088 maintenance vehicles. The cost is enormous. Over $15 million was budgeted by Indiana to support the operation during the 1987–1988 winter season and these costs are expected to increase by 20 percent for the 1988–1989 season.

The management of the operation is complex because of several factors. Unlike other route-oriented public-sector activities, snow removal must be initiated simultaneously throughout the entire event region, which, for Indiana, may mean as many as 1,200 vehicles commencing service at the same time. In addition to the snow removal fleet, a team of radio operators and mobile supervisory units is also required. The operation often extends through more than one worker shift, creating additional personnel management problems as well (*1*).

In addition to the magnitude of the overall operation, uncertainties as to the duration and severity of the snow emergency pose special problems. State roads are categorized into three classifications on the basis of historical average daily traffic (ADT). Class I roads (major traffic arteries including Interstates and their associated ramps and roads with ADT greater than 5,000) receive continuous service including plowing and the application of chemicals and abrasives as needed to keep the road surface wet and bare. Class II roads (routes having ADT between 1,000 and 5,000) receive continuous plowing and sufficient chemicals and abrasives to maintain a bare wet pavement in the center portion of the roadway. Class III roads (ADT less than 1,000) receive enough service to keep the routes passable, with chemical treatment only for hills, curves, and intersections. Any and all routes created should maintain as much route class continuity as possible; that is, each route should be created to service only one class of road. However, because of uncertainties associated with snow episodes, these criteria serve only as general guidelines; actual service levels are often determined while the operation is in progress.

Following a snow episode, extensive clean-up activities must be instituted, consisting of additional plowing and spot application of chemicals to remove all remaining snow from driving surfaces. This operation also includes clearing shoulder areas and special servicing of drains, overpasses, and bridges; drifts; and the maintenance equipment itself. Timing is extremely important in all phases of the operation.

In Indiana, as in most states, snow and ice control is administered at the subdistrict and site level consistent with preestablished snow routes of which there are some 986 in Indiana at present. Though all vehicles begin operation from a specific site location, routes proper do not necessarily begin at a site location. In cases where a truck must travel to the starting point of a route, no service will be conducted during this time. Such travel is referred to as "deadhead" travel, with deadhead miles accounting for nearly 9,000 vehicle-mi per season. Though travel times will fluctuate with overall conditions, design operation calls for plowing intensities dependent on roadway classification. In general, a route must be completely serviced in 2 to 2.5 hr.

Finally, the operation is complicated by the presence of public traffic. Plowing and spreading is most effective within a range of speed for the maintenance vehicle. When vehicle movement is restricted, performance decreases and road conditions will further deteriorate, causing additional problems and potentially dangerous conditions.

In designing an overall management strategy for snow and ice control, a number of difficult questions may be posed: What is the best set of routes for maintenance vehicles so as to minimize overall deadhead miles (minimize excess cost)? What characteristics of individual routes are most important in terms of overall safety of operation? How best should ve-

E. Haslam, Corporate Research and Development, United Airlines, Chicago, Ill. 60666. J. R. Wright, Department of Civil Engineering, Purdue University, West Lafayette, Ind. 47907.

hicles be restocked with sand and chemicals during the operation? What contingencies should be provided to compensate for the uncertainties of storm intensity? What are the best allocations of vehicles to sites and routes to vehicles? Are the current administrative boundaries optimal or should the configuration of subdistricts be modified? Where should new facilities be constructed to best enhance overall system performance? These questions and more should be considered in the design of an effective strategy for conducting winter road maintenance.

The focus of this research is the design of an efficient algorithm for the specification of snow and ice control routes, given complete and precise information about the underlying target network. All road segment lengths and adjacencies are known, and the locations of vehicle and abrasive storage facilities are fixed. Furthermore, a partition of the network for service from those sites is assumed. The problem is one of assigning road segments within the partition to garage sites (vehicles) consistent with a service policy that sets a specific performance level on the basis of time of service to a particular road type. Following a review of the relevant vehicle routing literature, the details of a multiobjective heuristic procedure for solving this problem are described. An actual route design exercise is presented and computational experience is discussed.

ROUTE DESIGN TECHNOLOGY

There is an extensive body of literature addressing the topic of vehicle routing and scheduling. In this section, a review is offered of that portion of this literature relevant to the winter service vehicle routing problem. Bodin et al. (*2*) provide a textbook synopsis and more complete literature review of the entire field of routing and scheduling of vehicles and crews.

General Methodologies

The topic of vehicle routing and scheduling covers such activities as retail distribution, school bus routing, mail delivery, street sweeping and snow removal, waste collection, and communications system management. Yet given this diverse spectrum of applications, the entire field of study can be partitioned into two major categories: (a) routing vehicles when the only concern is developing a set of routes to satisfy demand at several points or streets, and (b) scheduling vehicles when the concern is satisfying demand given time windows or precedence relations with regard to the demand points. Some authors (*3*) choose to make the major partition of the topic at demand type. That is, does the demand for service originate from points, making it a node-covering problem, or does it exist along the length of the street, making it an arc-covering problem?

The class of solution methodologies most relevant for this research fall under the classification of the Chinese postman problem (CPP). This classical operations research problem is one of finding the minimum cost cycle that visits every arc in a given network at least once. The basic CPP is well solved (*4*) but is computationally intractable on mixed networks (*5*). When a simple constraint such as a vehicle capacity restriction is imposed on the basic CPP on a directed or undirected network, the CPP becomes known as the "capacitated" CPP and is NP-hard (*6*). A polynomial time algorithm has been developed (*7*) for the CPP on a network with precedence relations on the arcs, which is of particular interest with regard to roadway clearing priorities for the snow removal problem.

The Chinese Postman Problem

In the world of network algorithms, the problem of snow removal is an arc-covering application in that the concern is covering every, or a subset of every, link or road in the network. The CPP is the basic mathematical programming formulation for this application; it is a classic problem statement within the operations research literature (*8*). As mentioned, the objective of the CPP is to find the least-cost path through the network that covers every arc at least once and starts and ends at the same node. For the snow removal problem, least cost is defined as minimum distance and arcs that are traversed more than once are tallied up as deadhead miles. Deadheads miles or arcs are segments traversed by a truck that is not servicing those segments either because they are serviced by another route or are used exclusively as access to a serviceable segment. An initial problem formulation was solved to find a CPP solution using one truck with infinite capacity starting from and returning to a common depot with the objective of covering every road in the network while minimizing total deadhead miles. Although this formulation is not a realistic representation of the actual multiple truck and depot problem, it does provide a lower bound on the solution and a good starting point for more complex models. The CPP mathematical program is a minimum-cost, flow-based formulation (*9*), as follows:

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}x_{ij} \qquad (1)$$

Subject to

$$\sum_{k=1}^{n} x_{ki} - \sum_{k=1}^{n} x_{ik} = 0 \qquad \text{for all } i, \text{ and} \qquad (2)$$

$$x_{ij} \geq 1 \text{ and integer} \qquad \text{for all } (i,j). \qquad (3)$$

In this formulation, the decision variables x_{ij} is the number of times the arc from Node i to Node j is covered in the optimal solution. The objective (Equation 1) seeks to minimize the total distance traveled, which is the summation of how many times an arc is covered, x_{ij}, times the length of the arc, d_{ij}. The first constraint set (Equation 2) enforces flow continuity for all n nodes in the networks by ensuring that every time a truck enters a node it must also leave that node. The final constraint set (Equation 3) states that every arc must be covered at least once and the answers must be integer; an arc being covered a fractional number of times has no meaning. It is a well known characteristic of this formulation that when solved as a continuous linear program using the simplex algorithm, an integer solution will result without the need for use of branching or cutting-plane methods.

The Rural Chinese Postman Problem

The first step toward model realism is to realize that INDOT is only responsible for a subset of all the roads in the state. That is, INDOT only has to service state roads, highways, and Interstates; however, its trucks may need to traverse a county road to provide service to a state road. Therefore, city streets and county roads should be included in the data, but with a flag designating them as merely nonrequired arcs that may be used to provide more efficient service or reduce deadheading miles. With the data so marked, a variant of the CPP known as the "rural postman problem (R–CPP) can be used to solve the problem. The formulation is identical to the CPP except for the addition of the following constraint:

$$x_{ij} \geq 0 \text{ and integer} \qquad \text{for all } (i,j) \in (A - R) \tag{4}$$

The modified formulation uses the notion of the set R, denoting the set of all arcs that INDOT is required to service, and the set A, which is the set of all of the arcs in the network. The additional constraint set (Equation 4) thus allows that all arcs that are not required to be serviced by INDOT do not have to be covered, but may be traversed. This discrete model will also terminate integer when solved as a linear program.

The Multiple Truck Rural Chinese Postman Problem

The previous model, R–CPP, used the idealized one truck of infinite capacity. Indiana's current snow control operations use multiple trucks departing and returning to a common depot of which there are several grouped into subdistricts that in turn are part of larger districts. If the resolution is restricted to one depot in any given subdistrict, the R–CPP may be extended to accommodate multiple vehicles and result in a formulation called the "M-rural postman problem" (MR–CPP). This mathematical program is a bit more complex, as follows:

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} d_{ij}\, x_{ij}^{m} \tag{5}$$

Subject to

$$\sum_{k=1}^{n} x_{ki}^{m} - \sum_{k=1}^{n} x_{ik}^{m} = 0 \qquad \text{for all } (i,m) \tag{6}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}\, x_{ij}^{m} \leq \text{MAXMILES} \qquad \text{for all } m \tag{7}$$

$$\sum_{k=1}^{m} x_{ij}^{m} \geq 1 \text{ and integer} \qquad \text{for all } (i,j) \in R \tag{8}$$

$$\sum_{k=1}^{m} x_{ij}^{m} \geq 0 \text{ and integer} \qquad \text{for all } (i,j) \in (A - R) \tag{9}$$

$$x_{\text{depot}}^{m} = 1 \qquad \text{for all } m \tag{10}$$

The MR–CPP has a modified decision variable x_{ij}^{m}, that identifies which trucks will cover each arc (i,j), so that there are now $m*a$ integer decision variables in the formulation, where a = number of arcs. The objective function (Equation 5) is the same: minimize total distance traveled by all m trucks. The continuity constraint (Equation 6) is the same flow continuity set, but now they are summed for every truck at each node. Vehicle capacity constraints (Equation 7) enforces vehicle capacity by setting the constant MAXMILES equal to the maximum number of miles each truck may travel. The next two constraint sets (Equation 8) and (Equation 9) are the same as before, but they must also be summed over all the vehicles. The final set of constraints (Equation 10) forces every vehicle to emanate from a designated depot.

Subtours

Solutions for the R–CPP and the MR–CPP may contain subtours. Subtour generation is the major stumbling block in the formulation of any type of exact routing algorithm. There are several different ways to address the problem of subtours, but for the current purpose, an exact solution for the R–CPP or MR–CPP, it is sufficient to discuss only one (*10*). The subtours could be eliminated if there existed constraints in the R–CPP or MR–CPP formulation that would not allow them to be formed in the first place. An example of such a set of subtour-breaking constraints is as follows (*6*):

$$\sum_{i \in R} \sum_{j \in R} x_{ij} - n^2 y_{1r} \leq |R| - 1$$

$$\text{for every nonempty subset } R \text{ of } \{2, 3, \ldots, n\} \tag{11}$$

$$\sum_{i \in R} \sum_{j \in R} x_{ij} + y_{2r} \geq 1 \tag{12}$$

$$y_{1r} + y_{2r} \leq 1 \tag{13}$$

$$y_{1r}, y_{2r} \in \{0,1\} \qquad r = 1, \ldots, (2^{n-1} - 1) \tag{14}$$

The problem with these constraints is that there are an exponential number of possible subtours that can be formed on the basis of the number of nodes; therefore there is an exponential number of possible constraints that could be added to the model to stop their formation. Two problems exist with these additional constraints: (a) for problems of any realistic size, say >100 nodes, the number of subtour breaking constraints grows to an outlandish size; and (b) these constraints force the model off of integer linear programming solutions. With the continuous linear program no longer guaranteed to terminate with an integer solution, the computational burden of iterating to an integer optimal solution can become prohibitive (*11*).

Conclusions

With the inclusion of subtour elimination constraints and the fact that there are $m*n$ integer variables in this formulation, the MR–CPP becomes even more hopeless as far as an exact solution by mathematical programming is concerned. But this formulation does provide an idea of the complexity and size of exact routing models as more real-world or problem constrains are imposed on the problem. Furthermore, an essential dimension of the problem is missing from the MR–CPP, the fact that there are multiple depots in each subdistrict. At a minimum, any design or analysis of routes and their configurations should be done at a subdistrict resolution. A multidepot MR–CPP formulation would require still more constraints and variables, and given the combinatorial explosion at the R–CPP and MR–CPP level, further listing and explanation are not necessary, given the purpose of this research.

What has been useful in creating and manipulating these exact formulations is the insights into the problem complexity they have unveiled. So far, no efficient method has been discovered to handle the different road classifications or priorities, given that each route should have class continuity. One way to handle this would be to partition the network into three subnetworks, one for each classification, and run a routing algorithm on each one. This procedure assumes that route class continuity is an absolutely binding constraint. (It is not, because the current set of INDOT routes violates route class continuity.) Therefore, this topic is better classified as a minor objective that could possibly be handled with a penalty function. This consideration leads to the point of identifying the true objective of this problem.

The public sector problem of snow and ice control is in reality a multiobjective problem. While minimizing cost, total distance, and number of trucks, INDOT also wishes to maximize the level of service to the public. This could be handled with some multiobjective optimization paradigm such as iterative trials, holding level of service constant and minimizing cost or maximizing efficiency (*12*).

A final point on these exact procedures is that although they are computationally impossible, they do provide the basic groundwork for formal heuristic methods such as linear programming column generation schemes. Lagrangian relaxation procedures, and others (*10,13*). These heuristic methods are appealing because they are based on the exact models and are therefore easy to justify and build the routes up from a trivial initial feasible solution to near optimal in a logical form.

Arc-Covering Applications and the Snow Removal Problem

The work by Marks and Stricker (*14*) addressed the problem of urban snow removal with the CPP as their base model. In their analysis of the snow plowing problem, they reveal the shortcomings of the CPP model with respect to real-world constraints such as the need for multiple plows and multiple-lane roads. The final and most restricting consideration mentioned is that of road priorities for which they provide several heuristic methods to overcome: (a) a weighting method that would make higher priority roads more attractive, (b) partitioning the network into three class-consistent subnetworks and solving each individually, and (c) ignoring priority and hope for a good solution. They conclude with an example problem in which they present a *cluster first–route second* heuristic that uses a CPP model for routing.

The algorithm provided by Lemieux and Campagna (*15*) is intended to enforce priorities by tracing an Eulerian circuit on a directed graph and picking higher-priority roads first whenever possible. This algorithm would have to be implemented with some heuristic procedure because it assumes that the truck can traverse all the arcs in the network.

Other arc-coving applications that use the CPP model or heuristic procedures include the routing of electric meter readers (*16*), waste collection (*17*), and routing street sweepers (*18–20*).

Finally, system studies (*21*) and simulations (*22,23*) have been conducted for the snow removal problem to help modularize this complex problem so that subsets of the problem may be solved separately and evaluated individually or used to drive another solution procedure. Of particular interest is the use of simulation to help the decision maker, human or machine, evaluate routes produced by some procedure, as described earlier. This evaluation could stimulate recommendations for route modifications or regeneration with additional constraints.

In developing a strategy for winter season snow and ice removal, the goal of the state highway authority is to provide efficient service within the constraints on available resources; plowing and abrasive spreading equipment, sand and salt supplies, and manpower. Although holding down overall cost is a primary consideration (*15*), the safety of the public is the major objective (*24*). Public safety in this context has two distinct but related components: (a) the condition of the road surface (*22*) and (b) the performance of the snow removal fleet during the operation. An effective snow removal operation is one that provides rapid and orderly snow removal and abrasive application without excessive interference with public transportation activity (*22,24,25*).

A MULTIOBJECTIVE HEURISTIC APPROACH

The CPP-based exact mathematical programming models presented in the previous section proved to be computationally infeasible as the sole solution procedure for generating snow removal routes. This was true for the case of one simple objective, the minimization of distance traveled, but with multiple objectives the exact models become all the more intractable and require alternative solution procedures.

A heuristic or nonexact procedure is an algorithm that will generate a feasible solution with no guarantee on optimality. The interested reader is directed to Chapter 7 of Parker and Rardin (*11*) for an overview and classification of nonexact algorithms with respect to discrete optimization. For the purpose of Indiana's snow route design problem, two possible heuristic approaches are investigated: (a) improving the existing route configuration, and (b) building routes exclusively from a definition of the network.

The snow route design problem may be defined as follows: develop the best set of routes for a given depot, subject to a given minimum level of service and with adherence to route class continuity. Best can be defined as a minimum cost configuration in which cost is measured by the number of trucks needed and total number of miles traveled. Levels of service vary for each class type and are defined in the INDOT policy (*26*). Adherence to route class continuity is not a binding constraint; it may better be defined as a secondary goal. From discussions with INDOT personnel from various levels, the predominate objective seems to be minimization of deadhead miles, because of the negative public reaction resulting from a snow plow traveling any road while it is not plowing. The public would also like to see its respective streets plowed first, which translates into maximizing the level of service. Therefore, the public sector problem of designing snow routes is now better defined as minimizing cost (taxes) and maximizing the level of service (public satisfaction). Thus, the snow route design problem becomes one of having multiple and conflicting objectives.

There are many different ways to approach a multiple-criterion optimization problem, depending on the way the analyst and the decision maker wish to impose the decision maker's preference or utility on the problem (*27*). The decision maker may specify preferences for each of the objectives with respect to each other a priori, a posteriori, or interactively during the actual solving of the problem (*12*). All of these methods are based on a tractable initial single objective problem formulation and are therefore of little use to the solution procedure being developed in this research.

The multiple criteria theories referenced earlier are used in papers by Henig (*28*) and White (*29*) with respect to the classical network theory problem of finding a shortest path on a given network with the added consideration of multiple objectives. The multiple-objective shortest-path problem is important to this research because some heuristic routing procedures, including the one developed for this research, use or require a set of shortest paths at some point in the algorithm.

Several usable heuristic procedures are described for improving and designing routes.

Improvement Procedures

One class of heuristic approaches is to randomly or arbitrarily create routes and try to improve them. Because INDOT has a set of routes that have historically evolved and were created by experts, a heuristic procedure that tries to improve these existing routes is probably the best approach. Two such methods are a swap-improvement heuristic and a route elimination heuristic.

The swap heuristic answers the basic question: Can the current routes be improved? Indicators of improvement could be reduction of total deadheading, better enforcement of route class continuity, greater route compactness, and other factors that experienced highway engineering personnel may provide. A heuristic algorithm of this type requires the network data previously mentioned and the current routes to be digitally encoded in a similar format. It then tries to modify the routes on the basis of the objectives listed by swapping arcs between routes or cutting down on deadheading. The multiple objectives may be considered by either an a priori weighting vector that would cause swaps that improved the weighted objective function the most to be done first, or by an interactive swapping procedure that relied on the decision maker to specify preferred improvements. This method results in more efficient and easier-to-drive routes that in turn result in a better level of service.

The elimination heuristic answers the question: Are all the routes needed that are currently used to service the network? This method requires the same digitally encoded data as the improvement method. It ranks and sorts the current routes on the basis of total and deadheading length, total number of arcs it covers, and nodes it shares with a route of the same classification. The algorithm will then try to eliminate a route by breaking it up and distributing its arcs among other routes. In this case, an a posteriori analysis of the reduced set of routes will have to be performed by the decision maker to evaluate the feasibility of the new configuration. If successful, this method results in fewer needed trucks, thus reducing one of the major costs involved in snow and ice control.

Generation Procedures

Route generation procedures attempt to build routes from nothing more than the defined network. Of particular interest are the procedures outlined by Golden (*18*)—nearest neighbor, Clark, and Wright savings, insertion, nearest merger, and others—and the interactive set partitioning method developed by Cullen (*13*). An ad hoc route generation heuristic was developed and it will be the focus of the remainder of this section.

The seed node–based snow route generator is an ad hoc, nonexact procedure that reads in a definition of a network, prompts the user for a set of seed nodes on the basis of an empirical analysis, and tries to grow snow routes from the seed nodes, subject to the objectives and constraints outlined previously.

The underlying network for this algorithm is assumed to be directed; an arc for each lane with an associated distance and class. The data file has a format that includes an ID number, start node, and end note; length and class are provided for each line; and an ID number, number of arcs directed in and out, list of arc ID's for arcs directed in and list of arc ID's for arcs directed out are provided for each node. The algorithm stores the network data in two dynamically created arrays, one containing a structure or object that has the given data file information, the distance from and to the depot for the node, and two lists of pointers to each arc directed into and out of the node for each node in the network; and the other contains a structure or object that has the given data file information plus pointers to the start and end nodes. A complete discussion of the implementation of this scheme for network research is presented by Haslam (*30*).

The first step in the algorithm is to read in the network data from the data file previously mentioned and dynamically create the arrays of nodes and lines. In this step, all the node-to-arc and arc-to-node pointers found in these arrays are established. If n denotes the number of nodes and a denotes the number of arcs in the network, then computationally this first step is at worst $O(n*a)$. The next step is to calculate the remaining information needed by the algorithm, the all-to-all shortest paths by the Floyd-Warshall algorithm, known to be $O(n^3)$ in the worst case (*9*).

The next step in the algorithm is to do an empirical analysis on the network. This analysis consists of calculating the total number of lane-miles for each class of road, which, when divided by the maximum distance a truck may travel when servicing each class of road, yields the minimum number of trucks needed to service a given network. The user is also prompted for the location of the depot. This theoretical minimum is then used when prompting the user for the number of trucks the user would like to try to generate routes for. In the worst case, this step is O(a).

At this point, the user is asked to supply a seed node and its corresponding route classification for each truck. The user specifies the seed nodes on the basis of the spatial configuration of the network with regard to arc classifications. That is, seed nodes should be placed among clusters of arcs sharing a common classification. The placement of seed nodes is also driven by the experience of the user as the algorithm evolves through iterative trials. With this information, the array of route headers is initialized and each linked list has its first leg created. Each seed node is tested for feasibility by checking its distance from the depot against the maximum distance allowed for its route classification. In the worst case, this step is O(n) and ends the initialization and user interaction phase of the algorithm. The remaining steps constitute the procedures that attempt to generate or grow routes on the basis of the seed nodes provided.

The grow-to-depot and the grow-from-depot phases of the algorithm each contain two substeps and are essentially the same procedures, with one working in a reverse direction. The purpose of these phases is to try to grow the route back to the depot from a seed node and to try to grow the route from the depot to a seed node. For the grow-to-depot phase, the algorithm starts adding arcs to the solution that are not already covered, are of the same class as the route that is currently being built, do not force the current route over its maximum distance constraint, and push the route closer to the depot. This process continues until the route includes the depot or there are no candidate arcs that satisfy the given conditions. In the second step of this phase, if the route does not include the depot, then the shortest path back to the depot is added to the solution ignoring class and whether the arcs used have been covered before; it is added to the route as deadhead coverage. This procedure is repeated for all of the routes and until they all contain the depot.

The grow-from-depot phase works in the same manner, but moves in the reverse direction. In both phases, the worst case will be O($n*a$). These procedures are denoted as Stage 1 and Stage 2 growth by the algorithm.

The next step in the procedure, expand-the-routes, is denoted as Stage 3 growth by the algorithm. At this point in the procedure, each of the routes consists of a shortest path to and from the depot for each seed node containing as many feasible arcs as possible in the solution. The algorithm now tries to add all of the currently uncovered arcs to the solution by expanding the routes so that they cover the desired arcs. In this Stage 3 growth, an uncovered arc will only be added to a route if it is the same class as the route, shares at least one node with the route, and if its inclusion in the route will not cause the route to exceed its length limitation. If only one node of the candidate arc is shared with the route, then all arcs adjacent to the node not in the route are checked to see if they (a) contain a node in the route, and (b) do not violate any of the previously listed constraints. The State 3 generation thus adds uncovered arcs one at a time or in pairs to adjacent routes on the basis of class and length restrictions. In the worst case, this phase could be O(a^4).

The final step of the algorithm, denoted as Stage 4 growth, is a relaxed expansion procedure. This phase is the same as Stage 3 with the exception of relaxing strict class continuity. Any uncovered arcs left at this point become candidates for route inclusion as before, but now they must only have a class that is not greater than the route that is being considered. The idea behind this phase is that it is all right for a higher-class route to provide a better level of service to a lower-class route, but not vice versa. Again, in the worst case this phase could be O(a^4).

At all phases of the algorithm, printouts are provided so that the user may track the progress of the routes as they are generated. The user has to provide different sets of seed nodes until a feasible or acceptable solution is reached. This algorithm often is not able to cover every arc in the network, but by intelligently varying the seed nodes and the number of trucks used, the user should be able to converge on a solution in polynomial time.

Seed Node–Based Route Generation Example

An all Class I, two-lane network that was used as the base network for an example solution is shown in Figure 1.

On the basis of the empirical analysis, four seed nodes were chosen: Nodes 4, 11, 11, and 7. Node 11 was chosen twice because of its extreme distance from the depot. Figures 2–4 represent each stage of the route generation procedure.

For this example, the maximum route length was set at 45 mi. With 122 lane-mi total, the theoretical minimum number of trucks needed to clear the network is three. With seed nodes of 4, 11, 11, and 7, the four routes covered the entire network with a total of 162 mi traveled, or 40 mi of total deadhead. This simple example was presented to illustrate the proposed routing heuristic. In the following section, some results are described from the application of several different models and analysis procedures on an actual test site in Indiana.

RESULTS AND DISCUSSION

The methods discussed in the previous sections have been tested using data from the Fowler subdistrict. INDOT, project

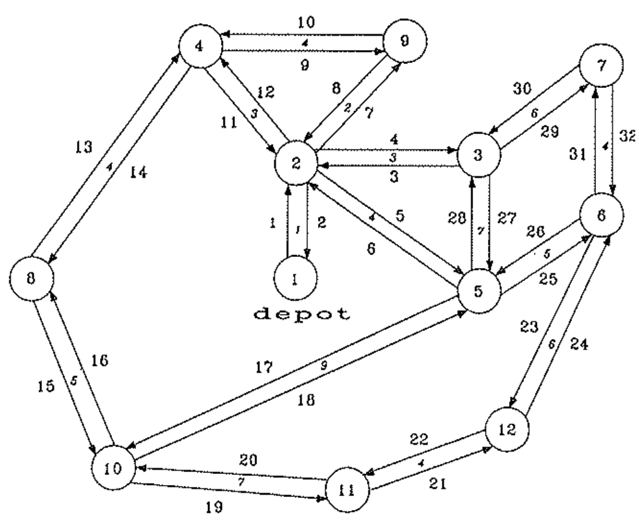

FIGURE 1 Network representation for example problem.

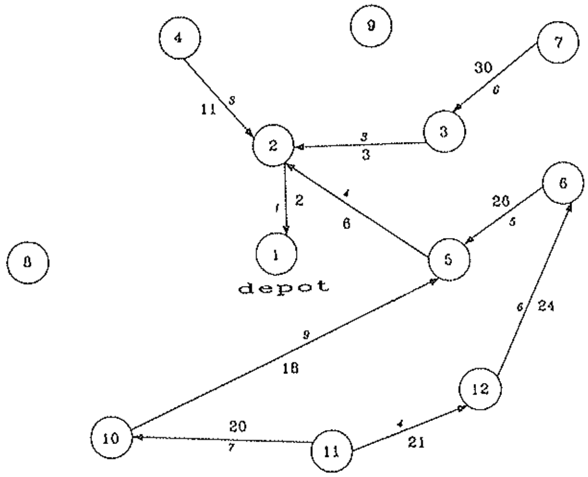

FIGURE 2 Solution after Stage 1 growth.

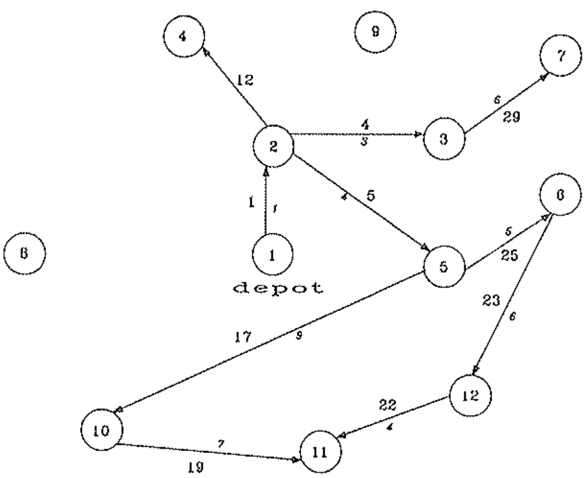

FIGURE 3 Solution after Stage 2 growth.

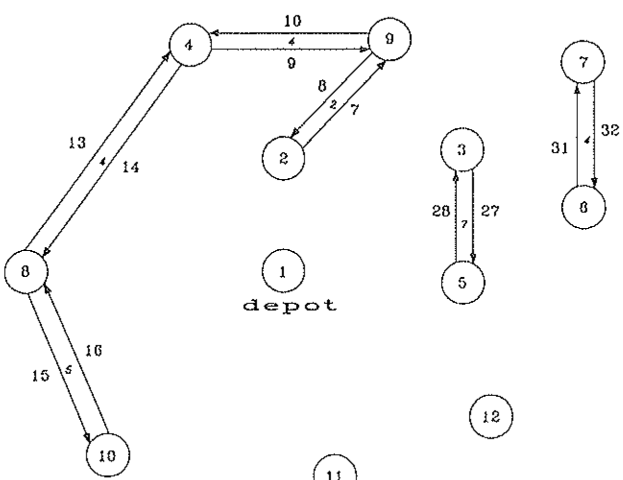

FIGURE 4 Solution after Stage 3 growth.

test site. At this point in the evolution of this project, the U.S. Geological Survey (*31*) digital data are still not fully compatible with the vehicle routing research of the previous sections, yet they are inherently important to any successful design or analysis previously outlined. To overcome this preliminary incompatibility some digital data was created with maps and a ruler. This effort has resulted in a 362-arc, 99-node link-node diagram that was hand encoded from a terminal. There is no precise indicator of the accuracy for these data. One limitation of the data is that only those roads currently used by INDOT in snow servicing were considered.

An empirical analysis of the Fowler subdistrict was undertaken to determine the minimum number of routes required to service the network. There are approximately 457 lane-mi of Class I roads, 271 lane-mi of Class II roads, and 188 lane-mi of Class III roads and Class I, II, and III routes cannot exceed 35, 50 and 65 mi, respectively. The Fowler subdistrict thus optimally (no deadheading) needs 13 Class I routes, 6 Class II routes and 3 Class III routes, or a total of 22 routes. The current INDOT configuration calls for a total of 26 routes or 4 extra routes more than the minimum needed. Given that there must exist some deadheading, a total of 26 routes is not far from a feasible optimal. This comes as no surprise given the fact that each set of routes was designed by a local expert, in terms of local geography and snow plow routing.

Exact Models: The CPP Formulation

A CPP was formulated and run using the test data. The result was a linear program with 362 variables, 99 flow continuity constraints, and an optimal solution of 905.9 lane-mi required to service the network. Only 2.39 mi were deadheading.

The digital data are currently the hand-encoded network. In the future, digital data will be used as described by Kurmas (*32*) and Wright et al. (*1*). Network data structures are two interlinked linked lists of nodes and arcs and their respective attributes created by a program that reads the raw digital data and generates the lists of structures. These linked lists of structures can be used to generate an XML model or a mathematical program, or to feed a heuristic algorithm. All mathematical programming formulations were solved using the XMP mathematical programming software (*33*).

The results of the CPP model run are presented in Table 1, Column 3. These results shed some light on the underlying structure of INDOT's network of roads and the accuracy of the network data. Out of a total of 903.5 lane-mi of roads that consisted of 362 arcs, only 2.39 mi (nine arcs) were needed to zero the polarity at every node in the network. Therefore, the state highway and road system taken as a completely directed network is a good underlying network for forming Euler or CPP tours. In Table 1, the difference in total required lane-miles between Columns 2 and 3 points out the shortcomings of relying too heavily on hand-scaling the map information and not having a digitally calculated and encoded data base for the network data. The exact mathematical models reviewed in the previous sections may be used as the platform for future route generation schemes in the following contexts: (a) they could be considered as the underlying model for a previously mentioned heuristic enumeration scheme, or (b) given an appropriate graphical interface, they could be

TABLE 1 METHODOLOGY EVALUATION PARAMETERS

Route Parameters (1)	Current Policy (2)	Theoretical Lower Bound via CPP (3)	Swap Heuristic Improvement (4)	Elimination Heuristic Elimination (5)
Total Required Lane Miles	915.4	903.5	915.4	915.4
Total Deadhead	175.8	2.39	135.2	160
Total Lane Miles	1091.2	905.9	1050.6	1075.4
# Routes	26	1	26	25

used without the exponential number of subtour breaking constraints and produce possible routes for a user to evaluate and impose needed constraints.

Heuristic Methods

The heuristic methods previously described hold the greatest promise for the actual generation of usable snow routes. Heuristics are especially well suited for problems that are a special case of a well-researched topic or algorithm, and snowplow routing with its road priorities and multiple objectives is a special and harder case of the general topic of vehicle routing and scheduling.

Improvement and Elimination Heuristics

The swap-improvement and elimination heuristics were evaluated with hand-encoded data with the routes as currently defined by INDOT as the data source. In the process of imposing the INDOT route definitions on the network definition, many shortcomings or conflicts were discovered between the two. Although many of the conflicts were resolved by editing the hand-encoded data, some still exist that will probably require ad hoc consideration. This is yet another justification for a digital network data base.

The computational effort and routing expertise required for the definition and implementation of the swap-improvement heuristic made its implementation prohibitive in the current project. This method requires the explicit definition of tradeoffs or preferences with respect to the objectives of minimizing cost and maximizing the level of service. These preferences could be defined by actual examples that are best demonstrated in a graphical decision support environment. In this environment, the decision maker could specify or evaluate swaps and then see the implications in the context of how the particular swaps affected the entire set of routes. Column 4 of Table 1, when compared with Column 1, demonstrates the effects of what would be a successful run of a heuristic of this type.

A rudimentary elimination heuristic was hand tested with the Fowler data and routes. This procedure evaluated routes with a relatively large number of deadhead miles, a relatively small number of arcs actually serviced, and at least one neighboring route of the same class. The result was the elimination of one route. In eliminating this route, three others were consequently modified. The removal of the route resulted in a loss of 15.8 deadhead miles and a corresponding surplus truck. The three modified routes all remain within the prescriptions set forth in the documented policy. Column 5 of Table 1 lists the results of this implementation of a elimination heuristic.

Seed Node–Based Snow Route Generation Heuristic

The ad hoc route generation heuristic developed for this research was tested on a subset of the Fowler Subdistrict data. The subset of data represents the arcs currently covered by one of the three depots in Fowler. There are 21 nodes and 54 arcs with all three classes of roads represented. Under the current INDOT route configurations, there are six routes used to cover this network, resulting in a total of 41.8 deadhead miles and 274.3 lane-mi traveled.

First, the heuristic could not cover all of the arcs in the network. However, the six routes currently used by INDOT contain two Class II routes that exceed the maximum allowable Class II length constraint by 10 to 13 mi each. The route generation heuristic currently contains no mechanism for relaxing the maximum distance constraints, but perhaps should, given the weak adherence to policy reflected by this example. For current purposes, a visual inspection of the routes produced by the heuristic and the underlying network indicates that the currently uncovered Arcs 41 and 42 would add 25.6 mi to any route to which they are added. Arcs 41 and 42 are adjacent to Route 4 (Class III, length = 58.9 mi) and Route 5 (Class II, length = 34.3 mi). With a relaxation of 15 mi for a Class II route, its maximum distance is now 60 mi, and the inclusion of Arcs 41 and 42 increases the length of Route 5 to 59.9 mi, but within the relaxed bound. Therefore, Arcs 41 and 42 are now covered by Route 5 with a new length of 59.9 mi.

This configuration of routes generated by the heuristic results in a total of 253.33 lane-mi traveled with 50.53 mi of deadhead travel and only one route violating its distance constraint. Therefore, it might be argued that although the heuristically generated routes contain more deadhead miles, they provide a higher level of service than the current INDOT routes. This demonstrates the tradeoff dilemma characteristic of multiple-objective problems, and is best evaluated by an appropriate decision maker.

The seed node–based snow route generator is an ad hoc heuristic procedure and consequently has several technical drawbacks:

1. Having the user specify the seed nodes used to generate the routes assumes that the user is familiar with the road network and thoroughly understands how the algorithm generates the routes. These requirements could be avoided by having the heuristic enumerate a set of candidate seed nodes and store the best sets of routes generated.
2. In its current form, the heuristic only looks for pairs of arcs to add to the solution when it is expanding the routes. In a rural setting, this is not a serious drawback because most of the network has zero-polarity nodes. In an urban setting with more complex interchanges and intersections this drawback may become a more serious problem. One way to overcome this deficiency would be to run a CPP algorithm on the original network and hand the heuristic the network with arcs added that ensured a CPP tour. The best way to remedy this fault would be to modify this part of the algorithm, a nontrivial task.
3. The only automated relaxation in the algorithm is that of route class continuity. There is a need for a route length relaxation procedure as was previously demonstrated and possibly for more interaction with the user while the problem constraints are being relaxed.
4. The algorithm is greedy. An enhancement for this would be to add an improvement module onto the procedure.
5. Other than finding shortest paths, the heuristic does no optimization. It is uncertain how this procedure might drive some sort of optimization model.

The major point in the heuristic's favor is that it is capable of generating feasible snow routes that, at least in one case, are comparable to existing implemented routes. Furthermore, it is quick, O(No. of arcs^4) in the worst case, and simple to understand and trace. Given its simplicity and problem-specific domain, it would be easy to modify to fit a particular snowplow route design problem.

A final point is that this heuristic algorithm is a relatively untested procedure that needs a few of the previously mentioned drawbacks ironed out before it could be compared with any other established vehicle routing solution procedure.

Future Directions

Throughout this final section, the need for accurate and comprehensive digital network data has been stressed. The reasons are twofold.

Without a common source of reliable data, the evaluation of any route generation scheme can be questioned. Many of these route design procedures may be able to find a better route configuration by traversing a local or county road that is not part of the state network, but will allow a state road or roads to be more efficiently serviced.

The second justification for digital data is perhaps the most important practical point unveiled in this research for INDOT. The Indiana Department of Highways would benefit greatly from establishing some form of a roadway information system (*34*). In the context of this research, such a system could be the backbone of a snowplow routing decision support system. Vehicle routing and scheduling systems have already been reported (*35–37*).

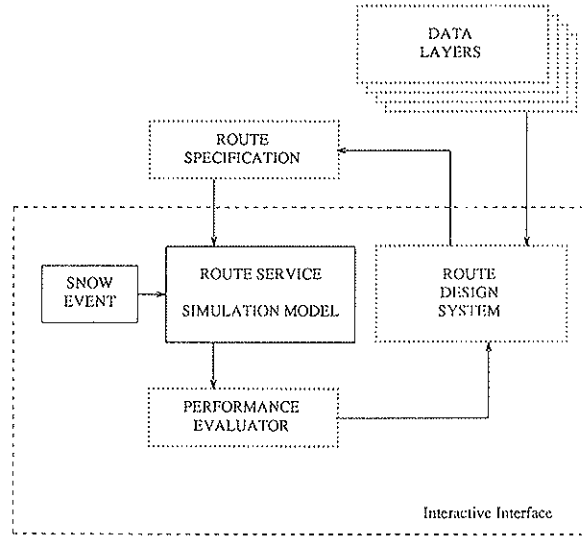

FIGURE 5 Schematic representation of route design system.

Set within the context of a decision support system as suggested by Figure 5, the complex and domain-dependent problem of snowplow route design could be simplified by intelligent user interaction and performance evaluation and analysis aids, such as a simulation model or a knowledge base.

REFERENCES

1. J. R. Wright, E. P. Haslam, P. Kurmas, K. Buehler, and S. Benabdallah. *Development of a Prototype System for Snow Route Design and Management*. Purdue University, West Lafayette, Ind., 1988.
2. L. Bodin, B. L. Golden, A. Assad, and M. Ball. The State of the Art in the Routing and Scheduling of Vehicles and Crews. *Computers & Operations Research*, Vol. 10, No. 2, Pergamon Press Ltd., 1983, p. 211.
3. R. C. Larson and A. R. Odoni. *Urban Operations Research*, Prentice-Hall, Englewood Cliffs, N.J., 1981, pp. 383–427.
4. E. Minieka. The Chinese Postman Problem for Mixed Networks. *Management Science*, Vol. 25, No. 7, July 1979, pp. 643–648.
5. C. H. Papadimitriou. On the Complexity of Edge Traversing. *Journal of the ACM*, Vol. 23, No. 3, July 1976, pp. 544–554.
6. B. L. Golden and R. T. Wong. Capacitated Arc Routing Problems. *Networks*, Vol. 11, 1981, pp. 305–315.
7. M. Dror, H. Stern, and P. Trudeau. Postman Tour on a Graph with Precedence Relation on Arcs. *Networks*, Vol. 17, pp. 283–294, 1987.
8. K. Mei-ko. Graphic Programming Using Odd and Even Points. *Chinese Mathematics*, Vol. 1, 1962, pp. 237–277.
9. R. G. Parker and R. L. Rardin. *IE 537—Class Notes*. Perdue University, West Lafayette, Ind., 1986.
10. B. L. Golden. Introduction To Recent Advances In Vehicle Routing Methods. In *Transportation Planning Models*, M. Florian, ed. Elsevier, North Holland, 1984, pp. 383–418.
11. R. G. Parker and R. L. Rardin. *Discrete Optimization*, Academic Press, San Diego, Calif., 1988.
12. R. E. Steuer. Multiple Criteria Optimization: Theory, Computation, and Application. John Wiley, New York, 1986.
13. F. H. Cullen, Jr. *Set Partitioning Based Heuristics For Interactive Routing*. Ph.D. dissertation, School of Industrial and Systems Engineering. Georgia Institute of Technology, Feb. 1984, p. 216.
14. D. H. Marks and R. Stricker. Routing for Public Service Vehicles. *ASCE Journal of the Urban Planning and Development Division*, Vol. 97, Dec. 1971, pp. 165–178.

15. P. F. Lemieux and L. Campagna. The Snow Ploughing Problem Solved by a Graph Theory Algorithm. *Civil Engineering Systems*, Vol 1, Dec. 1984, pp. 337–341.
16. H. I. Stern, and M. Dror. Routing Electric Meter Readers. *Computers and Operations Research*, Vol. 6, 1979, pp. 209–223.
17. E. J. Beltrami and L. D. Bodin. Networks and Vehicle Routing for Municipal Waste Collection. *Networks*, Vol. 4, pp. 65–94, 1974.
18. C. Mandl. *Applied Network Optimization*. Academic Press, London, 1979.
19. L. D. Bodin and S. J. Kursh. A Detailed Description of a Computer System for the Routing and Scheduling of Street Sweepers. *Computers & Operations Research*, Vol. 6, Pergamon Press Ltd., Great Britain, 1979, pp. 181–198.
20. T. M. Liebling. Routing Problems for Street Cleaning and Snow Removal. *Models for Environmental Pollution Control*, R. Deininger, ed. Ann Arbor Science Publishers, Ann Arbor, Mich., 1973, pp. 363–374.
21. D. L. Minsk. A Systems Study of Snow Removal. In *Special Report 185: Snow Removal and Ice Control Research*, TRB, National Research Council, Washington, D.C., 1979, pp. 220–225.
22. W. B. Tucker and G. M. Clohan. Computer Simulation of Urban Snow Removal. In *Special Report 185: Snow Removal and Ice Control Research*, National Research Council, Washington, D.C., 1979, pp. 293–302.
23. R. England. Computer Analysis Ensures a Clean Sweep. *Surveyor*, Vol. 6, May 1982, p. 15.
24. G. L. Russell and H. K. Sorenson. A Value Engineering Study of Snow and Ice Control. In *Special Report 185: Snow Removal and Ice Control Research*, TRB, National Research Council, Washington, D.C., 1979, pp. 66–73.
25. T. M. Cook and B. S. Alprin. Snow and Ice Removal in an Urban Environment. *Management Science* Vol. 23, No. 3, 1976, pp. 227–234.
26. *INDOT 1985–86 Snow Packet*. Public Service Office, Indiana Department of Highways, Dec. 1985.
27. G. W. Evans. An Overview of Techniques for Solving Multiobjective Mathematical Programs. *Management Science*, Vol. 30, No. 11, The Institute of Management Sciences, Nov. 1984, pp. 1268–1282.
28. M. I. Henig. The Shortest Path Problem With Two Objective Functions. *European Journal of Operations Research*, Vol. 25, Elsevier, 1985, pp. 281–291.
29. D. J. White. The Set of Efficient Solutions for Multiple Objective Shortest-Path Problems. *Computers and Operations Research*, Vol. 9, No. 2, 1982, pp. 101–107.
30. E. P. Haslam. *The Application of Routing Technologies to the Problem of Snow Removal*. M.S. thesis, School of Civil Engineering, Purdue University, West Lafayette, Ind., 1988.
31. *Digital Line Graphs from 1: 100,000 Scale Maps*. Data Users Guide 2, U.S. Geological Survey, Reston, Va., 1985.
32. P. Kurmas. *Processing of Digital Line Graph Data for Decision Support Applications*. M.S. thesis, Purdue University, West Lafayette, Ind., 1988.
33. R. E. Marsten. The Design of the XMP Linear Programming Library. *Transactions on Mathematical Software*, Vol. 7, No. 4, Dec. 1981.
34. F. L. Mannering and W. P. Kilareski. The Common Structure of Geo-Based Data for Roadway Information Systems. *ITE Journal*, July 1986, pp. 43–49.
35. P. Duchessi, B. Salvatore, and J. P. Seagle. Artificial Intelligence and the Management Science Practitioner: Knowledge Enhancements to a Decision Support System for Vehicle Routing. *Interfaces*, Vol. 18, No. 2, The Institute of Management Sciences, March–April 1988, pp. 85–93.
36. B. L. Golden and A. A. Assad. Perspectives on Vehicle Routing: Exciting New Developments. *Operations Research*, Vol. 34, No. 5, Sept.–Oct. 1986, pp. 803–810.
37. J. Potvin, G. Lapalme, and J. Rousseau. ALTO: A Computer System for the Design and Experimentation of Routing Algorithms. *Publication 525*, Centre de Recherche sur les Transports, University of Montreal, June 1987.

Publication of this paper sponsored by Committee on Winter Maintenance.