Les Cahiers du GERAD ISSN: 0711–2440

Vehicle Routing for Urban Snow Plowing Operations

N. Perrier, A. Langevin,
A. Amaya

G–2006–33

May 2006

Vehicle Routing for Urban Snow Plowing Operations

Nathalie Perrier
André Langevin*
Alberto Amaya

Department of Mathematics and Industrial Engineering
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec) Canada H3C 3A7
{nathalie.perrier; andre.langevin; alberto.amaya}@polymtl.ca

* *and GERAD*

May 2006

Les Cahiers du GERAD

G–2006–33

Abstract

Winter road maintenance planning involves a variety of decision-making problems related to the routing of vehicles for spreading chemicals and abrasives, for plowing roadways and sidewalks, for loading snow into trucks, and for transporting snow to disposal sites. In this paper, we present a model and two heuristic solution approaches based on mathematical optimization for the routing of vehicles for snow plowing operations in urban areas. Given a district and a single depot where a number of vehicles are based, the problem is to determine a set of routes, each performed by a single vehicle that starts and ends at the district's depot, such that all road segments are serviced while satisfying a set of operational constraints and minimizing a time objective. The formulation models general precedence relation constraints with no assumption on class connectivity, different service and deadhead speed possibilities, separate pass requirements for multi-lane road segments, class upgrading possibilities, and vehicle-road segment dependencies. Several extensions, such as turn restrictions, load balancing constraints, and tandem service requirements, which are required in a real-life application, are also discussed. Two objectives are considered: a hierarchical objective and a makespan objective. The resulting model is based on a multi-commodity network flow structure to impose the connectivity of the route performed by each vehicle. The two solution strategies were tested on data from the City of Dieppe in Canada.

Key Words: Winter road maintenance; Snow removal; Arc routing; Chinese postman problem; Operations research.

Acknowledgments: We wish to thank Mr. Jacques LeBlanc, Director of Public Services of the City of Dieppe who provided us with the data used in the computational experiments and Gilles Cormier for his valuable comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada and by the Fonds FQRNT of the Quebec Ministry of Research, Science and Technology. This support is gratefully acknowledged.

Résumé

La planification de l'entretien hivernal des réseaux routiers implique une variété de problèmes reliés au routage des véhicules pour l'épandage de fondants et d'abrasifs, pour le déneigement des rues et des trottoirs, pour le chargement de la neige dans des camions et pour le transport de la neige vers des sites de déversement. Dans cet article, nous présentons un modèle d'optimisation et deux méthodes de résolution approximatives pour le routage des véhicules pour les opérations de déblaiement en milieu urbain. Étant donné un district, un dépôt et une flotte de véhicules, le problème consiste à déterminer un ensemble de tournées partant et revenant au dépôt tel que chaque segment de rue est desservi par un seul véhicule, tout en respectant plusieurs contraintes opérationnelles. La formulation inclut des contraintes générales de préséance, des vitesses de service et de passages à vide différentes, des passages répétés obligatoires pour les rues à voies multiples, la possibilité d'augmenter l'ordre de préséance des rues non prioritaires, et des restrictions sur les rues qui peuvent être desservies ou traversées par chaque type de véhicules. Plusieurs extensions, telles que des pénalités pour limiter l'utilisation de certains types de virages, des contraintes d'équilibre de durée des tournées et la possibilité de desservir certaines artères en tandem, sont également discutées. Deux objectifs différents sont considérés: un objectif hiérarchique et un objectif de temps d'achèvement des opérations. Le modèle est basé sur un ensemble de problèmes de flot dans un réseau pour imposer les contraintes de connectivité de chaque tournée de véhicule.Les deux méthodes de résolution sont testées avec les données fournies par la ville de Dieppe au Canada.

Mots clés : Entretien hivernal; réseaux routiers, déneigement; tournées sur les arcs; problème du postier chinois; recherche opérationnelle.

Introduction

Snow plowing operations are usually performed in almost all urban regions with frozen precipitation or significant snowfall. Though each storm is unique in duration, intensity, and composition, vehicle routes for plowing operations are generally fixed at the beginning of the winter season. To facilitate the management of the plowing operations, the geographical region (or network) is usually partitioned into non-overlapping subareas (subnetworks), called districts, each including one depot at which a number of vehicles are based. The traditional approach for the design of districts in the context of winter road maintenance consists in partitioning the road network into districts by assigning road segments to their closest depot. Kandula and Wright (1995, 1997) and Muyldermans et al. (2002, 2003) used this approach for designing districts for plowing and spreading operations. A similar approach for designing small clusters of streets in the context of snow disposal operations was developed by Labelle et al. (2002). For a recent survey of optimization models and algorithms for the design of districts for winter road maintenance, the reader is referred to the work of Perrier et al. (2006b,c).

In this paper, we address the problem of vehicle routing within each district borders for snow plowing operations. For each district, the vehicle routing problem consists of determining a set of routes, each served by a single vehicle that starts and ends at the district's depot location, such that all road segments are serviced, all the operational constraints are satisfied, and a time objective is minimized. In addition, the configuration of routes needs to conform to existing district boundaries. Routes crossing these boundaries must be avoided from an administrative standpoint. In rural regions, only a subset of all road segments requires service, whereas most urban areas assume that all road segments of the district network must be serviced. Most naturally, each road segment is usually associated with two traversal times, which are possibly dependent on the vehicle type: the time required to plow the road segment and the time of deadheading the road segment. *Deadheading* occurs when a plow must traverse a road segment without servicing it. In general, a shorter time is associated with deadheading. Traversal times for servicing and deadheading a road segment have already been considered by Haghani and Qiao (2001) and Benson et al. (1998).

Different operational constraints can be imposed on the snow plow routes. For example, since agencies have finite resources that generally do not allow the highest level of service on all road segments, they must then prioritize their response efforts. The most common criterion for prioritizing response efforts is traffic volume. Typically, the road segments of a district network are partitioned into classes based on traffic volume and must be serviced while respecting a *hierarchy*, or precedence relation, between classes. Each subgraph induced by a class can be connected or not depending on the topology of the district network and on the level of service policies involved. One type of hierarchy constraint, called *linear precedence relation*, requires a unique ordering relation between classes in a route. This is the case where all roads carrying the heaviest traffic must be serviced first, followed by those that carry medium traffic volume, and so on. Another type of hierarchy constraint, called *general precedence relation*, imposes a weak partial ordering relation between classes in a route. This is the situation where all roads having a large traffic volume must be serviced before those having a low traffic volume in a route, but medium-volume roads can be serviced either before or after some high-volume and low-volume roads. However, some agencies allow *class upgrading*, the possibility of servicing road segments of a class in any of the classes of higher priority, in order to reduce the service completion time of this class and/or the total completion time. Class upgrading is also necessary when deadheading unserviced road segments (i.e., traversing road segments without servicing them), gets extremely difficult if not simply impossible. If so, plows must service each road segment the first time they traverse it while disregarding the hierarchy constraint.

Also, each vehicle type can have a restriction on the road segments that it can service and road segments that it can traverse. This constraint for each vehicle type is called *vehicle-road segment dependency.* In plowing operations, the vehicle fleet may consist of a collection of vehicles with varying size, service speed, and shape. Vehicles from the larger vehicle type cannot traverse small alleys. Vehicles from the slower vehicle type cannot service roads having a large traffic volume (for example, rotary plows). Some road segments allow vehicles from a vehicle type to traverse but not service the road segment because the road segment is too narrow to conduct service (for example, displacement plows mounted on the front, side, or beneath their truck carriers).

Finally, since plowing operations are usually limited to one lane at a time, multi-lane road segments necessitate multiple separate passes. This contrasts with materials spreading operations where materials are spread onto the road through a spinner which can be adjusted so that more than one lane of a road segment can be treated in a single pass.

The time objective considered for the routing of vehicles for plowing operations is to minimize the completion time of the first priority class, then the time of the second class, etc. This objective is called the *hierarchical* or *lexicographic* objective, as opposed to the *makespan* objective which minimizes the time at which all vehicles return to the depot, i.e., the shortest time required to service all road segments plus the shortest travel time from the last serviced road segment to the depot. The hierarchical criterion is well suited for snow plowing operations where road segments of higher priority classes must be serviced as soon as possible even if this requires a longer overall time. Moreover, the hierarchical objective is particularly appropriate when class upgrading possibilities are allowed since the vehicle routing problem with makespan objective and class upgrading possibilities is equivalent to the vehicle routing problem with makespan objective and no hierarchy constraint. The hierarchical objective has previously been considered by Cabral et al. (2004) and Korteweg and Volgenant (2006). Perrier et al. (2006a) studied a vehicle routing problem with makespan objective and class upgrading possibilities, but they impose a tolerance level on the total distance of lower-class road segments that can be serviced prior to higher-class road segments.

In a previous paper (Perrier et al., 2006a), we proposed a two-phase constructive method for the problem of vehicle routing for urban snow plowing operations. The method was developed by focusing on the specific needs of a particular city and incorporates a wide range of constraints and possibilities such as linear precedence relations with no assumption on class connectivity, separate passes or tandem plow patterns for multi-lane road segments (two vehicles plowing at the same time almost side by side), vehicle-road segment dependencies, left turn restrictions, load balancing across routes, and class upgrading possibilities. However, the method supposes that every arc and every vehicle type is associated with a single traversal time no matter if the arc is traversed by the vehicle while servicing or deadheading. The first phase determines a partition of the arcs into clusters, each having approximately the same workload, with an adaptation of the technique proposed by Benavent et al. (1990) for the capacitated arc routing problem. A directed hierarchical rural postman problem with makespan objective and class upgrading possibilities is then solved heuristically on each cluster using an extension of a procedure introduced by Dror et al. (1987) for the HCPP. Test results indicated that the method produced sets of routes that dominate the existing set of routes of the city with respect to either makespan objective, total duration of the routes, total distance travelled, or total duration unbalance occurring between routes. However, to maintain or enhance service levels in many cities, the emphasis should be placed on service completion time (hierarchical objective) as opposed to the time at which the vehicles return to the depot (makespan objective). Moreover, several cities choose to have a general precedence relation between classes in a route and each vehicle type usually has different service and deadhead speeds.

In this paper, we propose a formulation and two solution approaches based on a more general framework that can be adapted to the characteristics of several different cities. The model incorporates the hierarchical objective, general precedence relation constraints with no assumption on class connectivity, different service and deadhead speed possibilities, separate pass requirements for multi-lane road segments, class upgrading possibilities, and vehicle-road segment dependencies. Turn restrictions, load balancing constraints, and tandem service requirements are also enforced. The model is based on a multi-commodity network flow structure to impose the connectivity of the route performed by each vehicle with supplementary variables and constraints to model the hierarchical objective and is optimized with two constructive methods.

The rest of the paper is organized as follows. A brief review of literature is presented in the next section. In Section 2, a mathematical formulation of the problem is presented. Section 3 describes the two constructive methods. Computational experiments performed using data from the City of Dieppe, New Brunswick, Canada, are reported in Section 4 and conclusions are given in the last section.

1 Literature review

The vehicle routing problem treated in the present paper can be viewed as a multiple hierarchical Chinese postman problem (m-HCPP) with class upgrading possibilities and vehicle-road segment dependencies. The m-HCPP generalizes the hierarchical Chinese postman problem (HCPP), calling for the determination of a *single* route starting and ending at a depot and servicing *all* road segments of a network in such a way that the service hierarchy is satisfied and a time objective (makespan or hierarchical) is minimized. The HCPP is NP-hard (Dror et al., 1987), but can be solved in polynomial time if the precedence relation is linear and all subgraphs induced by the classes are connected. Dror et al. (1987), Ghiani and Improta (2000), and Korteweg and Volgenant (2006) have described exact algorithms for this case. The more realistic case, where the subgraph induced by a class is not connected, was first studied by Alfa and Liu (1988). The authors proposed a heuristic that first solves a rural postman problem on each subgraph induced by a class and then forms a giant tour satisfying the linear precedence relations. Later, Cabral et al. (2004) showed that it is possible to solve the HCPP with linear precedence relations and no assumption on class connectivity by transforming it into a rural postman problem. Gélinas (1992) described a dynamic programming algorithm for the HCPP with general precedence relations and class connectivity. Since the HCPP with no assumption on class connectivity is a special case of the m-HCPP, it follows that the m-HCPP is NP-hard. Hence, all algorithms developed for the solutions of m-HCPPs are heuristics.

One of the first heuristic algorithms developed for the solution of the vehicle routing problem for snow plowing operations is due to Moss (1970) who proposed a cluster-first, route-second approach to solve the vehicle routing problem for plowing and spreading operations in Centre County, Pennsylvania. Road segments are first organized into balanced sectors, and a vehicle route is obtained for each of them by solving a directed Chinese postman problem. The cluster phase tries to ensure that the graph generated by the edges of each sector is Eulerian to reduce deadheading in the routing phase.

Marks and Stricker (1971) presented two approaches for solving the problem of designing a set of m plow routes such that each road segment is cleared within either two or four passes, depending on its width, while minimizing the distance covered by deadheading trips. All plows are identical and multiple pass requirements are taken into account by duplicating each road segment as many times as the required number of passes on the road segment. The problem is modeled as a m-vehicle undirected Chinese postman problem. In the first approach, the transportation network is partitioned into m subnetworks by solving a districting problem, and a Chinese postman problem is solved for each of them using a decomposition heuristic. In the second approach, a unicursal graph is first derived from the original network, and arbitrarily partitioned into m mutually exclusive, collectively exhaustive subgraphs of approximately the same size so that an Eulerian cycle can be defined for each of them without additional duplication of edges. For details, see Stricker (1970). The authors also suggested three strategies to handle the hierarchy of the network when class connectivity is satisfied. The first strategy tries to allow the highest level of equipment usage on road segments of highest priority by multiplying the length of each road segment by its priority (with 1 being the highest priority) and solving a Chinese postman problem using these weighted lengths so as to favour the duplication of edges associated with road segments of highest priority. The second strategy solves a Chinese postman problem on each connected subgraph induced by the set of edges of a specific priority class and assigns exactly one vehicle to each postman tour. Finally, the last strategy generates several Eulerian cycles while disregarding road priorities, and chooses the cycle which best adheres to the hierarchy of the network.

The Bureau of Management Consulting, Transport Canada (1975), modeled a similar plow routing problem, with a homogeneous fleet of plows and multiple pass requirements for large road segments, as a m-vehicle undirected Chinese postman problem. Again, multiple pass requirements are taken into account by duplicating each road segment the required number of times. The problem is solved using a cluster first, route second heuristic, based on earlier work by Stricker (1970). The cluster phase breaks the original graph into small subgraphs according to several rules so as to enable routes with less deadheading. The route phase then solves an undirected Chinese postman problem in each subgraph and Fleury's algorithm (Kaufmann, p. 309, 1967) is used for determining an Eulerian cycle in the resulting Eulerian subgraph. The Bureau of Management Consulting also proposed to handle the hierarchy of the network and the direction of the traffic flow directly within Fleury's algorithm by selecting, at each iteration, the next edge of highest priority whose removal does not disconnect the Eulerian subgraph, while trying to respect the direction of the traffic flow.

Chernak et al. (1990) studied the problem of designing routes for two plows to clear the county roads in a district of Wicomico County, Maryland. The objective considered is to minimize the distance covered by deadheading trips, in addition to minimizing the plowing completion time. This problem is solved using a heuristic approach that constructs, for each plow, a primary route servicing roads of highest priority and a second route servicing the other roads.

A three-stage composite heuristic was proposed by Kandula and Wright (1997) for routing plows and spreaders in the state of Indiana. The heuristic takes into account class continuity and a maximum route duration for each class. *Class continuity* requires that each route services road segments with the same priority classification. In addition, both sides of a road segment must be serviced by the same vehicle. Given an undirected graph, the first phase identifies a set of seed nodes in sufficient number to respect the time limits, and then determines the maximum number of routes that can be constructed out of each seed node by means of an adaptation of the node scanning lower bound procedure introduced by Assad et al. (1997) for the capacitated Chinese postman problem. The second phase then constructs routes one at a time out of each seed node using a greedy optimality criterion. An improvement procedure that tries to reduce the distance covered by deadheading trips and the number of kilometers violating the class continuity constraints without exceeding the time limits is used last. Comparisons with the tabu search algorithm proposed by Wang and Wright (1994) for a vehicle routing problem for plowing and spreading operations on five networks of Indiana showed that the heuristic obtained the best solutions. However, it should be emphasized that the tabu search algorithm was stopped after a given number of iterations.

Finally, in a series of two papers, Salim et al. (2002a,b) proposed the SRAM (Snow Removal Asset Management) system to solve the vehicle routing problem for plowing and spreading operations in Black Hawk County, Iowa. The SRAM system can deal with service hierarchy and maximum route service times. Although the system relies in large part on decision rules drawn from interviews with experts, it also uses a simple constructive method that builds feasible routes one at a time for each class of roads using a greedy optimality criterion. Related field testing showed that the system reduced the total traversal time (service and deadheading) by 1.9–9.7% (depending on snowfall conditions) over the solution in use by the county.

While several models have been proposed for the m-HCPP in the context of snow plowing operations, a recent survey of models and algorithms for vehicle routing and fleet sizing for plowing and snow disposal (Perrier et al., 2005) indicates that very few have taken into account class upgrading possibilities and/or vehicle-road segment dependencies. One of the first efforts in this direction belongs to Haslam and Wright (1991) who developed an interactive route generation procedure for the plow routing problem at the Indiana Department of Transportation (INDOT), U.S. In this problem, routes of minimal total length that start and end at a given depot are sought and class continuity as well as maximum route length constraints must be satisfied. The route generation procedure starts by calculating a lower bound L_r on the number of routes to construct. The user then provides s seed nodes, $s \geq L_r$, with associated classes out of which feasible routes are constructed one at a time using a three-stage algorithm. Given a seed node and its class, the first stage of the algorithm constructs a feasible route made of a path from the seed node to the depot and another path in the reverse direction, without violating class continuity and maximum route length constraints. In the second stage, pairs of non-covered arcs of opposite direction are sequentially inserted into the route as long as class continuity and maximum route length permit. Finally, in the last stage, if arcs have not been covered, then the class continuity constraint is relaxed and the second stage is repeated by permitting class upgrading.

Wang and Wright (1994) described an interactive decision support system, called CASPER (Computer Aided System for Planning Efficient Routes), to assist planners at the Indiana Department of Transportation (INDOT) in the design of vehicle routes for plowing and spreading operations. The sectors are given and each of them contains exactly one depot. The system, which can accommodate service time windows, class continuity, and class upgrading, starts by calculating the number of routes to construct in a given sector for each class of roadways. For every class, the system builds the required number of vehicle routes starting and ending at the depot using a tabu search algorithm. An initial solution is obtained by means of a route growth heuristic described in Wang (1992), which is a refinement of the three-stage algorithm proposed by Haslam and Wright (1991). The system was tested on data from four northern districts of Indiana (Wang et al., 1995). On average, the system reduced the distance covered by deadheading trips and the number of routes by more than 4% and 7%, respectively, over the routing plan in use by INDOT.

Later, Campbell and Langevin (2000) described the commercially available vehicle routing software GeoRoute developed by the firm GIRO, based in Montreal, Canada, for postal delivery, winter maintenance, meter reading, street cleaning and waste collection applications. The GeoRoute software allows three types of winter road maintenance operations: plowing, spreading and snowblowing (for loading snow into trucks). The software can accommodate service time windows, service frequency, vehicle capacities, spreading rates, turn restrictions, vehicle-road segment dependencies, and both-sides service restrictions (servicing both sides of a road segment in a single route). GeoRoute uses a two-phase method similar to a cluster first, route second method, but constructs instead one route at a time. GeoRoute has been implemented in Ottawa, Canada (Miner, 1996, 1997) for snow plowing and in Suffolk County, United Kingdom (Guttridge, 2004) for salt spreading. Campbell and Langevin (2000) also report three implementations in the cities of Laval, Charlesbourg, and Nepean in Canada.

Very recently, Cabral et al. (2004) proposed a decomposition heuristic for the undirected HCPP with linear precedence relations and no assumption on class connectivity and hierarchical objective. The heuristic consists of sequentially solving the HCPP for each class, starting with the highest class, considering all traversed edges in any of the classes of lower priority as already serviced. As highlighted by Korteweg and Volgenant (2006), declaring a deadheading edge as already serviced can not increase total completion time, but may generate routes with a shorter time. Korteweg and Volgenant (2006) did not, however, provide a model or an algorithm to handle class upgrading possibilities.

2 Mathematical model

Formally, the problem of vehicle routing for urban snow plowing operations is defined on a strongly connected mixed graph $G = (V, A \cup E)$, where $V = \{v_0, v_1, \ldots, v_n\}$ is the vertex set, $A = \{(v_i, v_j) : v_i, v_j \in V \text{ and } i \neq j\}$ is the arc set, and $E = \{(v_i, v_j) : v_i, v_j \in V \text{ and } i < j\}$ is the edge set. Vertices $v_1, \ldots, v_n$ correspond to the road intersections, whereas vertex v_0 correspond to the depot at which are based m vehicles. Let $M = \{1, \ldots, m\}$ be the set of vehicles. Arcs and edges are used to represent one-way streets and multi-lane, two-way streets (one lane or more each way), respectively. For every arc and edge $(v_i, v_j) \in A \cup E$, let a_{ij}, e_{ij}, and e_{ji} be the number of circulation lanes associated with arc (v_i, v_j), edge (v_i, v_j) from v_i to v_j, and edge (v_i, v_j) from v_j to v_i, respectively. In plowing operations, since each lane must be serviced separately, each arc $(v_i, v_j) \in A$ is replaced by a_{ij} copies and each edge $(v_i, v_j) \in E$is replaced by e_{ij} arcs from v_i to v_j and by e_{ji} arcs from v_j to v_i. The resulting multigraph $G' = (V, A')$ is then directed. The arc set A' is partitioned into $\{A^1, A^2, \ldots, A^K\}$ with $A^1 \cup A^2 \cup \ldots \cup A^K = A'$ and $A^i \cap A^j = \emptyset$ for $i \neq j$, which induce the service hierarchy, i.e., all arcs of class A^i must be serviced before those of class A^{i+1}. Classes $1, \ldots, K-1$ represent road segments having a given priority whereas class K represents road segments that can be serviced anywhere in the sequence. For every class $p = 1, \ldots, K+1$, let $TMAX_p$ be a nonnegative real variable representing the service completion time of class p. Class $K+1$ is a fictitious class that allows to include the shortest travel path to the depot from the last serviced arc in class K for each vehicle. The graph G' is a multigraph, i.e., some arcs (v_i, v_j) may be replicated to model multi-lane road sections requiring separate servicing on each lane and road section widths requiring multiple servicing passes in addition to one-way streets requiring separate servicing on each side. Some arcs can be serviced by all types of vehicles, while others are restricted to certain types of vehicles only depending on the vehicle-road segment dependency requirements. For every vehicle $h \in M$, let $A_h \subseteq A$' be the subset of arcs in G' that can be serviced by vehicle h. With every vehicle $h \in M$ and every arc $(v_i, v_j) \in A_h$ are associated two positive durations s_{ij}^h and d_{ij}^h for the service and deadheading of arc (v_i, v_j) by vehicles h, respectively. For every vehicle $h \in M$, for every arc $(v_i, v_j) \in A_h$, and for every class $p = 1, \dots, K$, let x_{ij}^{ph} be a binary variable equal to 1 if and only if arc (v_i, v_j) is serviced in class p by vehicle h.

The mathematical formulation of the problem is based on a multi-commodity network flow problem to impose the connectivity of the route performed by each vehicle with supplementary variables and constraints. In this model, each commodity corresponds to a possible class-vehicle combination and shares the same directed graph $G'' = (V \cup \{v_a\}, A' \cup A_1 \cup A_2)$ constructed from G' where v_a is an artificial vertex, $A_1 = \{(v_a, v_i) : v_i \in V\}$ and $A_2 = \{(v_i, v_a) : v_i \in V\}$. An example of the construction of graph G" from G is illustrated in Figure 2.1. The arcs of A' and $A_1 \cup A_2$ are represented by dashed lines and dotted lines, respectively. The depot v_0 and the artificial vertex v_a are shown as dark and pale circles, respectively.

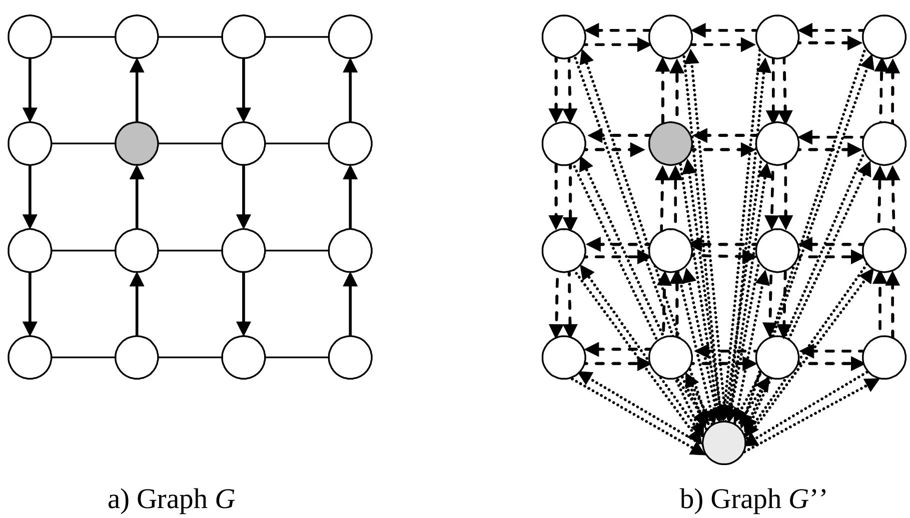


Figure 2.1: Construction of G'' from G

For every vehicle $h \in M$, for every arc $(v_i, v_j) \in A_h \cup A_1 \cup A_2$, and for every class $p = 1, \ldots, K+1$, let y_{ij}^{ph} be a nonnegative integer variable representing the number of times arc $(v_i,\ v_j)$ is traversed (while servicing or deadheading) in class p by vehicle h. For every vehicle $h \in M$, for every arc $(v_i,\ v_j) \in A_h \cup A_1 \cup A_2$, and for every class $p =$ 1,..., $K+1$, let w_{ij}^{ph} be a nonnegative real variable representing the flow on arc (v_i, v_j) associated with class p and vehicle h. Finally, for every class $p = 1, \ldots, K+1$ and for every vehicle $h \in M$, let t_p^h be a nonnegative real variable representing the service completion time of class p on route h. We include the t_p^h variables to clarify the formulation and the interpretation of results. The basic model for the problem of vehicle routing for urban snow plowing operations can be stated as follows:

Minimize

$$\sum_{p=1}^{K+1} M_p\, TMAX_p \tag{2.1}$$

subject to

$$TMAX_p \geq t_p^h \qquad (p = 1, \ldots, K+1, h \in M) \tag{2.2}$$

$$t_p^h = t_{p-1}^h + \sum_{(v_i,v_j)\in A_h} (s_{ij}^h x_{ij}^{ph} + d_{ij}^h (y_{ij}^{ph} - 1)) \qquad (p = 1, \ldots, K+1, h \in M) \tag{2.3}$$

$$t_0^h = 0 \qquad (h \in M) \tag{2.4}$$

$$\sum_{\substack{h\in M \\ (v_i,v_j)\in A_h}} \sum_{p=1}^{k} x_{ij}^{ph} = 1 \qquad ((v_i, v_j) \in A^k, k = 1, \ldots, K-1) \tag{2.5}$$

$$\sum_{\substack{h\in M \\ (v_i,v_j)\in A_h}} \sum_{p=1}^{K+1} x_{ij}^{ph} = 1 \qquad ((v_i, v_j) \in A^K) \tag{2.6}$$

$$\sum_{(v_i,v_j)\in A_h\cup A_1\cup A_2} y_{ij}^{ph} = \sum_{(v_i,v_j)\in A_h\cup A_1\cup A_2} y_{ji}^{ph} \qquad (v_i \in V \cup \{v_a\}, p = 1, \ldots, K+1, h \in M) \tag{2.7}$$

$$y_{ij}^{ph} \geq x_{ij}^{ph} \qquad ((v_i, v_j) \in A_h, p = 1, \ldots, K, h \in M) \tag{2.8}$$

$$\sum_{(v_i,v_j)\in A'\cup A_1\cup A_2} w_{ij}^{ph} = \sum_{(v_j,v_i)\in A'\cup A_1\cup A_2} w_{ji}^{ph} \qquad (v_i \in V \cup \{v_a\}, p = 1, \ldots, K+1, h \in M) \tag{2.9}$$

$$y_{ij}^{ph} \leq w_{ij}^{ph} \leq |A'|\, y_{ij}^{ph} \qquad ((v_i, v_j) \in A_h \cup A_1, p = 1, \ldots, K+1, h \in M) \tag{2.10}$$

$$y_{ij}^{ph} \leq w_{ia}^{ph} \qquad ((v_i, v_j) \in A_h, p = 1, \ldots, K+1, h \in M) \tag{2.11}$$

$$\sum_{v_i\in V} y_{ai}^{ph} = 1 \qquad (p = 1, \ldots, K+1, h \in M) \tag{2.12}$$

$$\sum_{v_i\in V} y_{ia}^{ph} = 1 \qquad (p = 1, \ldots, K+1, h \in M) \tag{2.13}$$

$$y_{ia}^{ph} = y_{ai}^{p+1,h} \qquad (v_i \in V, p = 1, \dots, K, h \in M) \quad (2.14)$$

$$y_{a0}^{1h} = 1 \qquad (h \in M) \quad (2.15)$$

$$y_{0a}^{K+1,h} = 1 \qquad (h \in M) \quad (2.16)$$

$$x_{ij}^{ph} \in \{0,1\} \qquad ((v_i, v_j) \in A_h, p = 1, \dots, K, h \in M) \quad (2.17)$$

$$y_{ij}^{ph} \geq 0 \text{ and integer} \qquad ((v_i, v_j) \in A_h \cup A_1 \cup A_2, \; p = 1, \dots, K+1, h \in M) \quad (2.18)$$

$$w_{ij}^{ph} \geq 0 \qquad ((v_i, v_j) \in A_h \cup A_1 \cup A_2, \; p = 1, \dots, K+1, h \in M) \quad (2.19)$$

$$TMAX_p \geq 0 \qquad (p = 1, \dots, K+1) \quad (2.20)$$

$$t_p^h \geq 0 \qquad (p = 0, \dots, K+1) \quad (2.21)$$

where $M_1 >> M_2 >> \ldots >> M_{K+1} = 1$. The objective function (2.1) minimizes the service completion time of the first priority class, then the completion time of the second class, and so on. As highlighted by Cabral et al. (2004), the notation ">>" means that in any feasible solution, the relation

$$M_p \, TMAX_p > \sum_{k=p+1}^{K+1} M_k \, TMAX_k$$

must be satisfied for $p = 1, \ldots, K$. Constraints (2.2) state that the maximum service completion time of a given class must be greater than or equal to the service completion time of this class on any route. Constraints (2.3) and (2.4) define the service completion time of each class on each route and the start time of each route, respectively. Constraints (2.5) and (2.6) assure that each arc of a given priority class is serviced either in this class or in any of the classes of higher priority by exactly one eligible vehicle and that all other arcs are serviced by exactly one eligible vehicle, respectively. A vehicle is *eligible* for a certain arc if it can service or deadhead this arc. Constraints (2.7) ensure route continuity for each possible class-vehicle combination. Constraints (2.8) state that an arc is serviced by an eligible vehicle in a given class only if it is traversed by the same vehicle in the same class. Flow conservation at every node for each class and for each vehicle is imposed by Constraints (2.9). Constraints (2.10) assure that the flow on every arc associated with a class and an eligible vehicle is positive if and only if this arc is traversed (while servicing or deadheading) in that class by that vehicle. Constraints (2.11) ensure that each partial route associated with a class and a vehicle does not contain any disconnected subtours. Constraints (2.12) and (2.13) require that each class-vehicle combination be associated with exactly two vertices of G: one start location at which the route must start service, called *start vertex*, and one end location at which the route must end service, called *end vertex*, respectively. Constraints (2.14) assure that the end vertex associated with a class and a vehicle corresponds to the start vertex associated with the next class and the same

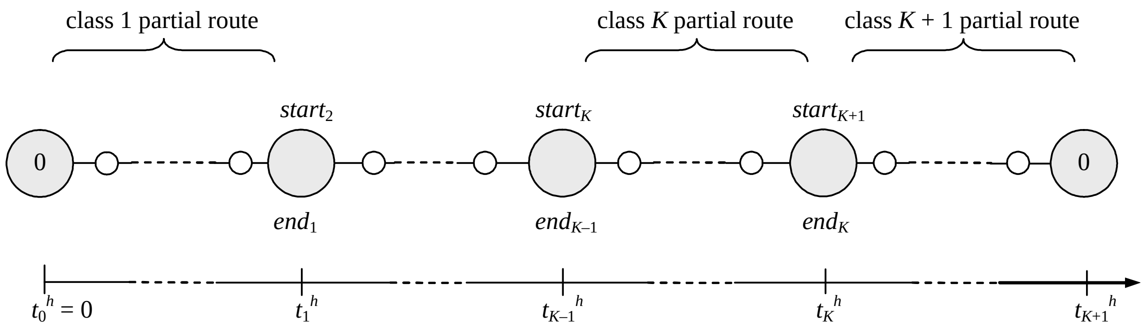


Figure 2.2: Schematic representation of a feasible route h in G"

vehicle. Constraints (2.15) and (2.16) require that each route starts and ends at the depot location, respectively. A schematic representation of a feasible route h in G" is illustrated in Figure 2.2. Recall that the partial route associated with class $K + 1$ corresponds to the shortest path from the last serviced arc by vehicle h in class K to the depot v_0.

Finally, all y_{ij}^{ph} variables must assume nonnegative integer values and all w_{ij}^{ph}, t_p^h, and $TMAX_p$ variables must assume nonnegative values, while x_{ij}^{ph} variables are restricted to be binary.

Proposition 1. *A feasible solution to the model (2.1)–(2.21) does not contain any disconnected subtours.*

Proof. We first show that the partial route associated with given vehicle h and class p does not contain any disconnected subtours. First, observe that for a given arc $(v_i,\, v_j) \in A_h \cup A_1$ serviced by vehicle h in class p, w_{ij}^{ph} must take on positive values if and only if $y_{ij}^{ph} = 1$. Thus, any arc $(v_i,\, v_j) \in A_h$ serviced by vehicle h in class p must be connected to the start vertex v_{start}^{ph} associated with vehicle h and class p ($v_{start}^{ph} \neq v_i$, $v_{start}^{ph} \neq v_j$) since

$$y_{ij}^{ph} = 1 \Rightarrow w_{i0}^{ph} \geq 1$$

and

$$\sum_{v_k \in V} y_{ak}^{ph} = 1 \Rightarrow w_{ak}^{ph} \geq 1, v_k = v_{start}^{ph} \text{ and } y_{ak}^{ph} = 0, v_k \neq v_{start}^{ph} \Rightarrow w_{ai}^{ph} = 0$$

imply that the flow variables associated with class p and vehicle h must define a directed path P from the start vertex v_{start}^{ph} to v_i to satisfy flow conservation at vertex v_i. This in turn implies that all arcs on P must be serviced or deadheaded by vehicle h in class p. Moreover, the partial route associated with vehicle h and class p must be connected to the depot v_0 since the partial route associated with vehicle h and every class $p = 1, \ldots, K+1$ does not contain any disconnected subtours and $v_{start}^{1h} = v_0$, $v_{end}^{1h} = v_{start}^{2h}, \ldots,$ $v_{end}^{p-1,h} =$

v_{start}^{ph}, $v_{end}^{ph} = v_{start}^{p+1,h}$,..., $v_{end}^{K+1,h} = v_0$. Thus, the route h does not contain any disconnected subtours. □

The multi-commodity network flow structure can also be used to model the contiguity constraints in a linear form for the design of sectors for snow disposal operations. Contiguity constraints require that sectors do not include distinct parts separated by other sectors. Non-contiguous sectors are undesirable from both administrative and operational standpoints given that deadheading trips would be necessary between the disjoint collections of road segments of each non-contiguous sector. For details, the reader is referred to the recent work of Perrier et al. (2006d).

The model (2.1)–(2.21) can be customized to deal with many additional situations. First, the model assumes that all types of turns made at intersections are allowed. However, in urban areas, vehicle routes must observe traffic rules such as the prohibition of making certain turns, mostly left turns and U-turns. More generally, even if they are not forbidden, the impact of undesirable turns, such as U-turns and turns across traffic lanes, is usually greater in routing snow plows as compared to spreading operations. Since most plows are designed to always cast the snow to the right side of the roadways, a left turn or a street crossing at an intersection results in a trail of snow in the middle of the intersection. Thus, the general guideline for constructing routes for snow plowing is that each plow should remain on the right side of a roadway using a block pattern by accomplishing a series of right turns to avoid compromising safety. To deal with these situations, a penalty can be imposed to each turn depending on its type (e.g., left, right, U-turn, and go straight). In snow plowing operations, right turns would be given the lowest penalty to provide safe roads. For each pair of arcs (v_i, v_j), $(v_j, v_k) \in A'$, denote $[(v_i, v_j), (v_j, v_k)]$ as the turn made going from arc (v_i, v_j) to arc (v_j, v_k) in G'. Let T denote the set of allowed turns in G'. For each turn $[(v_i, v_j), (v_j, v_k)] \in T$, for each class $p = 1, \ldots, K + 1$, and for each vehicle $h \in M$, let $n_{[ijk]}^{ph}$ be a nonnegative integer variable representing the number of times turn $[(v_i, v_j), (v_j, v_k)]$ is executed in class p by vehicle h. Then, the constraints

$$\sum_{(v_j,v_k)\in A'} n_{[ijk]}^{ph} = y_{ij}^{ph} \qquad ((v_i,v_j)\in A', p=1,\ldots,K, h\in M) \tag{2.22}$$

$$\sum_{(v_k,v_i)\in A'} n_{[kij]}^{ph} = y_{ij}^{ph} \qquad ((v_i,v_j)\in A', p=1,\ldots,K, h\in M) \tag{2.23}$$

must be added to model (2.1)–(2.21) to impose turn penalties. Constraints (2.22) ensure that the number of times a turn beginning with arc (v_i, v_j) is executed by a given vehicle in a given class corresponds to the number of times this arc is traversed by the same vehicle in that class. Constraints (2.23) serve the same purpose for turns that terminate with arc (v_i, v_j). Turn penalties are imposed by adding the term $\sum_{h\in M}\sum_{p=1}^{K}\sum_{[(v_i,v_j),(v_j,v_k)]\in T} p_{[ijk]}\, n_{[ijk]}^{ph}$ to the objective function (2.1), where $p_{[ijk]}$ is the penalty associated with turn $[(v_i, v_j), (v_j, v_k)]$. Examples of plow routing applications where turn penalties are explicitly taken into account are provided in Lemieux and Campagna (1984), Robinson et al. (1990), Gendreau et al. (1997), and Campbell and Langevin (2000).

Next, load balancing constraints can be introduced easily. Balancing the workload across routes means creating routes of approximately the same duration. If the minimum and maximum route durations are denoted by l and u, respectively, then the following constraints can be added to the original formulation:

$$l \leq \sum_{(v_i,v_j)\in A_h} \sum_{p=1}^{K+1} (s_{ij}^h x_{ij}^{ph} + d_{ij}^h (y_{ij}^{ph} - 1)) \leq u \qquad (h \in M) \tag{2.24}$$

Finally, the last situation concerns the inclusion of tandem service constraints. Since plowing operations are limited to one lane at a time, many agencies have developed tandem plow patterns in echelon formations for multi-lane road segments. Generally, these road segments must be serviced in any of the classes of higher priority. Let A^1 be the set of arcs that require tandem service. Given a set of pairs of vehicles to operate in parallel, the most popular approach is thus to assign each pair of vehicles to a single class 1 partial route starting at the depot and covering multi-lane road segments requiring tandem service. Let $A_{\text{tandem}} = \{(i_r,i_s), (i_t,i_u) \in A^1$: (i_r,i_s) and (i_t,i_u) necessitate tandem plow patterns$\}$ and let M_{tandem} be the set of pairwise compatible vehicles to operate in parallel such that each vehicle belongs to at most one pair. Then, adding the constraints

$$x_{rs}^{1h_1} = x_{tu}^{1h_2} \big((i_r, j_s) \in A_{\text{tandem}} \cap A_{h_1}, (i_t, j_u) \in A_{\text{tandem}} \cap A_{h_2}, (h_1, h_2) \in M_{\text{tandem}}\big) \tag{2.25}$$

ensures that pairs of arcs that necessitate tandem plow patterns are assigned to pairwise compatible vehicles in M_{tandem}.

3 Solution approaches

Even for small instances of the problem, model (2.1)–(2.25) contains a very large number of variables and constraints. We propose two constructive methods to solve this model. The first method, called *parallel algorithm*, constructs several routes in parallel by sequentially solving a multiple vehicle rural postman problem (m-RPP) with vehicle-road segment dependencies, turn restrictions, and load balancing constraints for each class $p = 1, \ldots, K$, considering all traversed arcs as already serviced. Recall that in the RPP, the arc set is partitioned into *required* and *non-required* arcs. The m-RPP consists of designing a set of m vehicle routes of least total cost, such that each route starts and ends at the depot and each required arc appears in at least one route and is serviced by exactly one vehicle. The second approach, called *cluster first, route second algorithm*, first determines a partition of the arcs into clusters, each having approximately the same workload. A hierarchical rural postman problem (HRPP) with class upgrading possibilities, vehicle-road segment dependencies, and turn restrictions is then solved on each cluster. We first present the parallel route constructive approach and then describe the cluster first, route second method.

3.1 Parallel algorithm

The parallel algorithm is based on a decomposition of the model into a set of K different subproblems. For every class $p = 1, \ldots, K$, the subproblem determines the $|M|$ best partial class p routes for the given start times when the partial class p routes must start service and for the given start locations at which the partial class p routes must start service through the solution of a m-RPP with vehicle-road segment dependencies, turn restrictions, and load balancing constraints, considering all arcs of class p traversed in any of the classes of higher priority as non-required arcs to allow class upgrading. For every vehicle $h \in M$, the time required to return to the depot, i.e., t_K^h plus the shortest travel time to the depot from the last serviced arc in class K by vehicle h, is determined last. For every class $p = 1, \ldots, K$, the m-RPP is of the following form:

Minimize

$$TMAX_p + \sum_{h \in M} \sum_{[(v_i,v_j),(v_j,v_k)] \in T} p_{[ijk]} n_{[ijk]}^{ph} \tag{3.1}$$

subject to

$$TMAX_p \geq t_p^h \qquad (h \in M) \tag{3.2}$$

$$t_p^h = s_p^h + \sum_{(v_i,v_j) \in A_h} (s_{ij}^h x_{ij}^{ph} + d_{ij}^h (y_{ij}^{ph} - 1)) \qquad (h \in M) \tag{3.3}$$

$$\sum_{\substack{h \in M \\ (v_i,v_j) \in A_h}} x_{ij}^{ph} = 1 \qquad ((v_i, v_j) \in A^p) \tag{3.4}$$

$$\sum_{(v_i,v_j) \in A_h \cup A_1 \cup A_2} y_{ij}^{ph} = \sum_{(v_i,v_j) \in A_h \cup A_1 \cup A_2} y_{ji}^{ph} \qquad (v_i \in V \cup \{v_a\}, h \in M) \tag{3.5}$$

$$y_{ij}^{ph} \geq x_{ij}^{ph} \qquad ((v_i, v_j) \in A_h, h \in M) \tag{3.6}$$

$$\sum_{(v_i,v_j) \in A' \cup A_1 \cup A_2} w_{ij}^{ph} = \sum_{(v_j,v_i) \in A' \cup A_1 \cup A_2} w_{ji}^{ph} \qquad (v_i \in V \cup \{v_a\}, h \in M) \tag{3.7}$$

$$y_{ij}^{ph} \leq w_{ij}^{ph} \leq |A'| \, y_{ij}^{ph} \qquad ((v_i, v_j) \in A_h \cup A_1, h \in M) \tag{3.8}$$

$$y_{ij}^{ph} \leq w_{ia}^{ph} \qquad ((v_i, v_j) \in A_h, h \in M) \tag{3.9}$$

$$\sum_{v_i \in V} y_{ai}^{ph} = 1 \qquad (h \in M) \tag{3.10}$$

$$\sum_{v_i \in V} y_{ia}^{ph} = 1 \qquad (h \in M) \tag{3.11}$$

$$y_{a,start_p^h}^{ph} = 1 \qquad (h \in M) \tag{3.12}$$

$$\sum_{(v_j,v_k) \in A'} n_{[ijk]}^{ph} = y_{ij}^{ph} \qquad ((v_i, v_j) \in A', h \in M) \tag{3.13}$$

$$\sum_{(v_k,v_i)\in A'} n^{ph}_{[kij]} = y^{ph}_{ij} \qquad ((v_i,v_j)\in A', h\in M) \qquad (3.14)$$

$$l \le \sum_{(v_i,v_j)\in A_h} (s^h_{ij}x^{ph}_{ij} + d^h_{ij}(y^{ph}_{ij}-1)) \le u \qquad (h\in M) \qquad (3.15)$$

$$x^{ph}_{ij}\in\{0,1\} \qquad ((v_i,v_j)\in A_h, h\in M) \qquad (3.16)$$

$$y^{ph}_{ij}\ge 0 \text{ and integer} \qquad ((v_i,v_j)\in A_h\cup A_1\cup A_2, h\in M) \qquad (3.17)$$

$$w^{ph}_{ij}\ge 0 \qquad ((v_i,v_j)\in A_h\cup A_1\cup A_2, h\in M) \qquad (3.18)$$

$$TMAX_p\ge 0 \qquad (3.19)$$

$$t^h_p\ge 0 \qquad (3.20)$$

For every class $p = 1,\ldots,K$, the objective function (3.1) minimizes the sum of the service completion time of class p and the penalties associated with turns. For any given class, constraint set (3.2) is identical to its counterpart (2.2) of the model (2.1)–(2.25). For every class $p = 1, \ldots, K$, constraints (3.3) define the service completion time of class p on each route given the start time s^h_p of class p on route h. For every class $p = 1, \ldots, K$ constraints (3.4) assure that each arc of class p is serviced by exactly one eligible vehicle. For any given class, constraint sets (3.5)–(3.11) are identical to their respective counterparts (2.7)–(2.13) of the model (2.1)–(2.26). For every class $p = 1, \ldots, K$ and for every vehicle $h \in M$, constraints (2.12) require that each partial class p, route h starts service at its start location $start^h_p$. For any given class, constraint sets (3.13)–(3.15) are identical to their respective counterparts (2.22), (2.23), and (2.24) of the model (2.1)–(2.25). Finally, if $p =$ 1, then the constraints (2.25) must be added to the model (3.1)–(3.20) to impose tandem plow patterns.

For every class $p = 1, \ldots, K$ and for every vehicle $h \in M$, let end^h_p represent the end location at which the partial class p, route h ends service. Given two routes $R^h_p = (start^h_p, \ldots, end^h_p)$ and $R^h_{p+1} = (start^h_{p+1} = end^h_p, \ldots, end^h_{p+1})$ representing the partial class p, route h and the partial class $p + 1$, route h in G, respectively, and having a common endpoint end^h_p in G, let $R^h_p + R^h_{p+1} = (start^h_p, \ldots, end^h_p = start^h_{p+1}, \ldots, end^h_{p+1})$ denote the union of the arcs of these two partial routes. For every vehicle $h \in M$, let R_h be the (possibly partial) route h in G, let SP^h_K be the shortest duration path from the last serviced arc in class K by vehicle h to the depot, and let sp^h_K be its travel time. The parallel algorithm can be described more precisely as follows.

1. Set $p = 1$. For every vehicle $h \in M$, set $s^h_p := 0$, and $start^h_p := v_0$. For every vehicle $h \in M$, set $R^h := \emptyset$.
2. If $p = 1$, add constraints (2.25) to model (3.1)–(3.20) and solve the resulting model. Otherwise, solve model (3.1)–(3.20). Let R^h_p be the resulting partial class p, route h. For every vehicle $h \in M$, declare all traversed arcs on R^h_p as already serviced and set $R_h := R_h + R^h_p$. Set $TMAX_p = \max_{h\in M} \{t^h_p\}$.

3. If $p = K$, go to Step 4. Otherwise, set $p = p + 1$. For every vehicle $h \in M$, set $s_p^h := t_{p-1}^h$, $start_p^h := end_{p-1}^h$, and return to Step 2.
4. For every vehicle $h \in M$, set $R_h := R_h + SP_K^h$ and $t_{K+1}^h = t_K^h + sp_K^h$. **Stop**.

3.2 Cluster first, route second algorithm

The cluster first, route second algorithm first determines a partition of the arcs to be serviced into compact clusters, each having approximately the same workload. Since vehicles plow at different speeds, the total workload is measured in time units (e.g., minutes). A vehicle route is then constructed in each cluster through the solution of a HRPP with class upgrading possibilities, vehicle-road segment dependencies, and turn restrictions. We adapted a technique proposed by Benavent et al. (1990) for partitioning the arcs into clusters. This technique determines the assignment of the arcs by solving a generalized assignment problem (GAP). It is inspired by the Fisher and Jaikumar (1981) generalized-assignment-based algorithm for the capacitated vehicle routing problem.

The algorithm starts by locating $|M|$ geographically dispersed arcs of A' to serve as seed arcs $s_1 \in A_1, \ldots, s_h \in A_h$ for the $|M|$ vehicles. The criterion for widely dispersing seed arcs over the graph G" is to maximize the product of the shortest paths among the seed arcs and the depot v_0. In particular, if seeds $s_1, \ldots, s_h$ have already been selected, seed arc s_{h+1} is chosen to maximize $\prod_{k=0,\ldots,h} (sp_{ak} + sp_{ka})$ over all arcs a. For every vehicle $h \in M$ and for every arc $(v_i, v_j) \in A_h$, let x_{ij}^h be a binary variable equal to 1 if and only if arc (v_i, v_j) is assigned to vehicle h, let d_{ijh} and d_{hij} represent the lengths of the shortest paths from arc (v_i, v_j) to arc s_h and from s_h to (v_i, v_j), respectively. Then the problem of assigning each arc of A' to exactly one of the $|M|$ vehicles can be formulated as a linear 0–1 integer program as follows.

Minimize

$$\sum_{h \in M} \sum_{(v_i,v_j) \in A_h} (d_{ijh} + d_{hij}) x_{ij}^h \tag{3.21}$$

subject to

$$\sum_{\substack{h \in M \\ (v_i v_j) \in A_h}} x_{ij}^h = 1 \qquad ((v_i, v_j) \in A') \tag{3.22}$$

$$L \leq \sum_{(v_i,v_j) \in A_h} s_{ij}^h x_{ij}^h \leq U \qquad (h \in M) \tag{3.23}$$

$$x_{rs}^{h_1} = x_{tu}^{h_2} \quad ((i_r, j_s) \in A_{\text{tandem}} \cap A_{h_1}, (i_t, j_u) \in A_{\text{tandem}} \cap A_{h_2}, (h_1, h_2) \in M_{\text{tandem}}) \tag{3.24}$$

$$x_{ij}^h \in \{0, 1\} \qquad ((v_i, v_j) \in A', h \in M) \tag{3.25}$$

where

$$L = \frac{\sum\limits_{(v_i v_j) \in A'} \left(\frac{\sum\limits_{k \in M} s_{ij}^h}{|M|} \right)}{|M|} - \alpha \frac{\sum\limits_{(v_i v_j) \in A'} \left(\frac{\sum\limits_{k \in M} s_{ij}^h}{|M|} \right)}{|M|},$$

$$U = \frac{\sum\limits_{(v_i v_j) \in A'} \left(\frac{\sum\limits_{k \in M} s_{ij}^h}{|M|} \right)}{|M|} + \alpha \frac{\sum\limits_{(v_i v_j) \in A'} \left(\frac{\sum\limits_{k \in M} s_{ij}^h}{|M|} \right)}{|M|}$$

and $0 \leq \alpha \leq 1$.

The objective function (3.21) minimizes the sum of all lengths of the shortest paths from the arcs in G" to the seed arcs and from the seed arcs to the arcs, so as to assess the compactness of every cluster. Constraints (3.22) assure that each arc is assigned to exactly one eligible vehicle. Constraints (3.23) impose a specified lower bound L and an upper bound U on the total workload of each vehicle. Pairs of arcs that necessitate tandem plow patterns by pairwise compatible vehicles are imposed by constraints (3.24).

Once the $|M|$ clusters have been determined, a HRPP with class upgrading possibilities, vehicle-road segment dependencies, and turn restrictions is then solved on each cluster. Model (2.1)–(2.23) can be used to this end. However, the assignment constraint sets (2.5) and (2.6) must be replaced with the constraints

$$\sum_{\substack{h \in M \\ (v_i, v_j) \in S_h}} \sum_{p=1}^{k} x_{ij}^{ph} = 1 \qquad ((v_i, v_j) \in A^k, k = 1, \dots, K-1) \tag{3.26}$$

$$\sum_{\substack{h \in M \\ (v_i, v_j) \in S_h}} \sum_{p=1}^{K+1} x_{ij}^{ph} = 1 \qquad ((v_i, v_j) \in A^K) \tag{3.27}$$

where S_h is the subset of required arcs assigned to vehicle h in the solution to the linear 0–1 integer model. Hence, each vehicle is now required to service only a subset of arcs. The route phase works by iteratively solving the resulting model for every vehicle $h \in M$, considering all traversed arcs on routes 1,..., $h - 1$ as already serviced in order to reduce the total completion time. The cluster first, route second algorithm can be summarized as follows.

1. *Cluster phase*
    1. Determine a set $\{s_1, \dots, s_{|M|}\}$ of $|M|$ seed arcs, selecting for each vehicle $h \in M$ an arc s_h such that the product of the sum of the shortest paths between s_h and the seed arcs $s_1, \dots, s_{h-1}$ and between s_h and the depot v_0 is maximum.

2. Solve model (3.21)–(3.25) to determine the assignment of each arc of A' to exactly one of the $|M|$ vehicles. For each vehicle $h \in M$, let S_h be the set of arcs assigned to vehicle h in the solution to the linear 0–1 integer model. Set $h = 1$.

2. *Route phase*

   a) Solve model (2.1)–(2.24) with constraint sets (2.5) and (2.6) replaced by constraint sets (3.26) and (3.27), taking S_h as an input, to construct the h-th vehicle route. Let R_h be the resulting route operated by vehicle h. Declare all traversed arcs on route h as non-required.

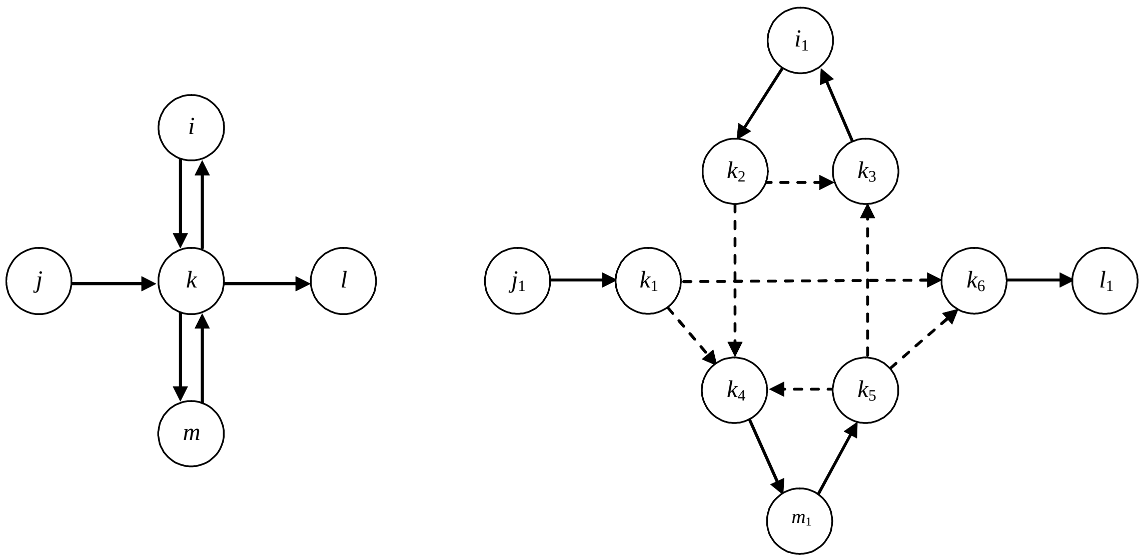
   b) If $h = |M|$, set $TMAX_p = \max_{h \in M} \{t_p^h\}$ for each class $p = 1, \ldots, K + 1$ and **stop**. Otherwise, set $h := h + 1$ and return to step 2a.

4 Computational experiments

To measure the performance of the proposed solution approaches, computational experiments were performed using data from the City of Dieppe, New Brunswick, Canada. All linear integer models were programmed using OPL studio 3.7 running with CPLEX 9.0 (after tuning the parameters) and the procedure to determine the set of seed arcs in the cluster phase of the cluster first, route second algorithm was coded in VBA using Microsoft Visual Basic 6.3. All experiments were performed on a Pentium 4 personal computer. A maximum running time of 3600 seconds was imposed for each linear integer programming problem. We first describe the data requirements and then give a summary of the results obtained.

4.1 Data requirements

The Dieppe data were extracted from a digitalized map stored in an image format and the current vehicle routes were obtained from Public Services of the City of Dieppe. We used Forestry GIS (fGIS) to extract the topology of the transportation network and road segment lengths from the graphical file. The road network of Dieppe involves 462 vertices and 1234 arcs partitioned into three classes A^1, A^2, and A^3 representing arterial streets, collecting streets, and local streets, respectively, with $|A^1| = 244$, $|A^2| = 229$, and $|A^3| = 761$. The subgraph induced by the set of arcs of class A^1 is Eulerian while the subgraphs induced by the set of arcs of classes A^2 and A^3 are not strongly connected. The City of Dieppe currently uses eight vehicles of three different types: one grader, two plows, and five loaders. The grader and the plows can clear 1,5 lanes in each pass, whereas the loaders can only clear one lane at a time. Moreover, the latter are restricted to class 2 and 3 streets, while the grader and the plows do not have any such restrictions. Each vehicle type has the same service and deadhead speeds. Thus, $s_{ij}^h = d_{ij}^h$ for every vehicle $h \in M$ and every arc $(v_i, v_j) \in A_h$. Plows travel at a speed of 25 km/h on class 1 and 2 streets and at 10 km/h on class 3 streets. The grader can travel at 20 km/h on class 1 and 2 streets and



a) Street intersection (k)

b) Same street intersection with replicated vertices and artificial arcs

Figure 4.1: Replicating vertices and introducing artificial arcs

at 10 km/h on class 3 streets. The loaders travel at a speed of 10 km/h. The precedence relation between classes in a route is linear, i.e. all arterial streets must be serviced before collecting streets and all collecting streets must be serviced before local streets, and the makespan objective is minimized. Linear precedence relations can be incorporated easily to the formulation by setting $A^K = \emptyset$.

The routes must also take into account forbidden left turns. These turns can be penalized by assigning them a high positive penalty and accounting for it in the model. However, as highlighted by Benavent and Soler (1999) and Corberán et al. (2002), a method that takes into account turn penalties, but not turn prohibitions, and tries to avoid the forbidden turns by assigning them a high penalty cannot guarantee to produce routes that avoid forbidden turns. A direct way of modelling forbidden turns is to transform the m-HCPP with forbidden left turns as a m-HRPP by adding artificial arcs to graph G'. For this, vertices have to be replicated. To illustrate, consider the street intersection v_k shown in Figure 4.1a, where the two left turns $[(v_i, v_k), (v_k, v_l)]$ and $[(v_j, v_k), (v_k, v_i)]$ are forbidden and all other turns are allowed. Figure 3.2b illustrates the replication of vertex v_k and the introduction of nine non-required arcs shown as dashed lines to represent straight crossings, right turns, left turns, and U-turns. The two forbidden left turns $[(v_i, v_k), (v_k, v_l)]$ and $[(v_j, v_k), (v_k, v_i)]$ can thus be avoided by eliminating their corresponding non-required arcs (k_2, k_6) and (k_1, k_3), respectively, in the augmented graph G', and the remaining non-required arcs have zero cost.

Finally, some class 1 multi-lane road segments necessitate tandem plow patterns in echelon formations. The city requires that service in tandem should be accomplished by the two plows.

4.2 Summary of results

Four scenarios were used in the experiments. In the first scenario, class upgrading possibilities are forbidden and the hierarchical objective is minimized. In the second scenario, the hierarchical objective is still used, but the possibility to service road segments of a class in any of the classes of higher priority is now allowed. The third scenario considers the makespan objective instead of the hierarchical objective and prohibits class upgrading possibilities. Finally, the fourth scenario disregards the linear precedence relations between classes in each route. Since the parallel algorithm constructs feasible routes for each class independently, it cannot be employed for solving the hierarchy relaxation.

In Scenarios 1 and 3, class upgrading prohibitions are treated by replacing constraints (2.5) and (2.6) with the constraints

$$\sum_{\substack{h \in M \\ (v_i,v_j) \in A_h}} x_{ij}^{kh} = 1 \qquad ((v_i,v_j) \in A^k, k = 1, \ldots, K-1) \tag{4.1}$$

$$\sum_{\substack{h \in M \\ (v_i,v_j) \in A_h}} x_{ij}^{Kh} = 1 \qquad ((v_i,v_j) \in A^K) \tag{4.2}$$

respectively, and by not considering all traversed arcs as already serviced in steps 2 and 2a of the parallel and cluster first route second algorithms. If the makespan objective is considered instead of the hierarchical objective, then $TMAX$ may be a nonnegative real variable representing the time required to service all arcs of $A^1 \cup A^2 \cup \ldots \cup A^{K-1}$ plus the shortest travel time to the depot from the last serviced arc. One would then replace the objective function (2.1) by (4.3) and constraints (2.2) and (2.20) by (4.4) and (4.5), respectively.

$$TMAX \tag{4.3}$$

$$TMAX \geq t_{K+1}^h \qquad (h \in M) \tag{4.4}$$

$$TMAX \geq 0 \tag{4.5}$$

We were unable to obtain a feasible solution to the subproblem (3.1)–(3.20) for class 3 within the prescribed time limit. For this class, the m-RPP was solved with the following adaptation of the cluster first route second algorithm.

1. For every vehicle $h \in M$, let end_2^h be the seed vertex for vehicle h. Let R^3 be the set of non-serviced arcs of class 3. For every vehicle $h \in M$ and for every arc $(v_i,$

$v_j) \in A_h \cap R^3$, let x_{ij}^h be a binary variable equal to 1 if and only if arc $(v_i,\, v_j)$ is assigned to vehicle h, and let d_{hij} represent the length of the shortest path from seed vertex end_2^h to arc $(v_i,\, v_j)$. Solve the following linear 0–1 integer model to determine the assignment of each arc of R^3 to exactly one of the $|M|$ vehicles.
Minimize

$$\sum_{h \in M} \sum_{(v_i,v_j) \in A_h \cap R^3} d_{hij} x_{ij}^h \tag{4.6}$$

subject to

$$\sum_{\substack{h \in M \\ (v_i v_j) \in A_h}} x_{ij}^h = 1 \qquad ((v_i, v_j) \in R^3) \tag{4.7}$$

$$L \le t_2^h + \sum_{(v_i,v_j) \in A_h \cap R^3} s_{ij}^h x_{ij}^h \le U \qquad (h \in M) \tag{4.8}$$

$$x_{ij}^h \in \{0,1\} \qquad ((v_i, v_j) \in R^3, h \in M) \tag{4.9}$$

where$L = t_2^1 + r^1 - \alpha(t_2^1 + r^1)$ and $U = t_2^1 + r^1 + \alpha(t_2^1 + r^1), 0 \le \alpha \le 1$. For each vehicle $h \in M$, the time r^h for servicing all arcs of R^3 by vehicle h is determined such that the equations

$$t_2^{h_1} + r^{h_1} = t_2^{h_2} + r^{h_2} \qquad (h_1, h_2 \in M) \tag{4.10}$$

$$\sum_{h \in M} r^h = \sum_{(v_i v_j) \in R^3} \left(\frac{\sum\limits_{h \in M} s_{ij}^h}{|M|} \right) \tag{4.11}$$

are satisfied. For each vehicle $h \in M$, let S_h be the set of arcs of class 3 assigned to vehicle h in the solution to the linear 0–1 integer model. Set $h = 1$.

2. Set $A^3 = R^3$ and $A^k = \emptyset$, $k = 1,\, 2$, and solve model (2.1)–(2.24) with constraints (2.4) and (2.15) replaced with the constraints

$$t_0^h = t_2^h \qquad (h \in M) \tag{4.12}$$

$$y_{a,end_2^h}^{1h} = 1 \qquad (h \in M) \tag{4.13}$$

   and constraints (2.5) and (2.6) replaced with the constraints (3.26) and (3.27), taking S_h as an input. Let R_3^h be the resulting partial class 3 route operated by vehicle h. Declare all traversed arcs on route R_3^h as already serviced and set $R_h := R_h + R_3^h$.
3. If $h = |M|$, set $TMAX_3 = \max_{h \in M}\{t_3^h\}$ and go to step 4 of the parallel algorithm. Otherwise, set $h := h + 1$ and return to step 2.

Table 4.1 presents the service completion times obtained when solving the instance under the first three scenarios with the parallel algorithm and the four scenarios with the

Table 4.1: Completion times and percentage gaps

	Scenario			
	1	2	3	4
Completion time (h)				
Parallel algorithm				
$TMAX_{1P}$	1.2	1.2	1.2	–
$TMAX_{2P}$	2.0	1.9	2.0	–
$TMAX_{3P}$	5.0	5.3	5.0	–
$TMAX_{4P}$	5.7	5.7	5.2	–
Cluster first route second algorithm				
$TMAX_{1C}$	1.4	1.4	1.4	4.7
$TMAX_{2C}$	1.9	1.9	1.9	5.1
$TMAX_{3C}$	4.8	4.5	5.3	5.1
$TMAX_{4C}$	5.3	4.9	5.5	5.2
Percentage gap (%)				
$(TMAX_{1P} - TMAX_{1C})$ / $TMAX_{1C}$	–17.9	–15.3	–18.3	–
$(TMAX_{2P} - TMAX_{2C})$ / $TMAX_{2C}$	5.2	1.2	1.4	–
$(TMAX_{3P} - TMAX_{3C})$ / $TMAX_{3C}$	4.9	17.3	–5.7	–
$(TMAX_{4P} - TMAX_{4C})$ / $TMAX_{4C}$	8.9	15.6	–6.4	–

cluster first route second algorithm. The last four rows compare the completion times of both solution approaches and contain the percentage difference in completion time between the parallel and the cluster first route second algorithms, based, respectively, on the completion times of classes 1, 2, and 3, and on the total completion times (class 4). Some gaps are negative because the service completion time produced by the cluster first route second algorithm is sometimes longer than the service completion time produced by the parallel algorithm.

When the hierarchical objective is minimized (Scenarios 1 and 2), we observe that for classes 2, 3, and 4, the cluster first route second algorithm improve upon the parallel algorithm. For classes 3 and 4, up to 17.3% and 15.6% can be saved in completion time, respectively, by applying the cluster first route second algorithm. For class 2, the partial routing can still be carried out with slightly less time with the cluster first route second algorithm. However, for class 1, the partial routes constructed by the cluster first route second algorithm incur at least 15.3% more time than those found by the parallel algorithm. This performance is easily explained from the following observations: since the subgraph induced by the set of arcs of class A^1 is Eulerian, in the parallel algorithm, only the deadheading travel time to the first serviced arc from the depot will make each partial class 1 route more expensive. However, in the cluster first route second algorithm, if the subgraph induced by the set of arcs of class 1 assigned to a vehicle is neither Eulerian nor strongly connected, then a partial class 1 route with more deadheading will be created during the route phase. Thus, the cluster phase should ensure that the graph generated by the arcs of class 1 of each cluster is Eulerian to reduce deadheading in the routing

Table 4.2: Percentage gaps: Dieppe's method vs. parallel and cluster first route second algorithms

	Scenario			
	1	2	3	4
Percentage gap (%)				
$(TMAX_{1D} - TMAX_{1P})$ / $TMAX_{1P}$	137.5	131.7	137.5	–
$(TMAX_{2D} - TMAX_{2P})$ / $TMAX_{2P}$	172.2	183.0	172.2	–
$(TMAX_{3D} - TMAX_{3P})$ / $TMAX_{3P}$	2.2	–2.8	3.0	–
$(TMAX_{4D} - TMAX_{4P})$ / $TMAX_{4P}$	–2.8	–2.1	8.1	–
$(TMAX_{1D} - TMAX_{1C})$ / $TMAX_{1C}$	95.0	96.2	93.9	–40.6
$(TMAX_{2D} - TMAX_{2C})$ / $TMAX_{2C}$	186.3	186.3	176.1	6.3
$(TMAX_{3D} - TMAX_{3C})$ / $TMAX_{3C}$	7.2	14.0	–2.9	1.1
$(TMAX_{4D} - TMAX_{4C})$ / $TMAX_{4C}$	5.8	13.1	1.1	6.7

phase. When the makespan objective is minimized (Scenario 3), for all but one of the priority classes the completion time produced by the parallel algorithm is lower than that produced by the cluster first route second algorithm.

With respect to the computation times, we note that the cluster first route second algorithm solves faster than the parallel algorithm. For each scenario, the parallel and the cluster first route second algorithms require to solve 12 and 9 linear integer programming problems, respectively. In most of the 60 linear integer programming problems, the maximum running time of 3600 seconds was not sufficient to reach and prove optimality. Thus, the time it takes the cluster first route second algorithm to complete a single scenario is about 9 hours. This computing time seems reasonable given that decisions related to the routing of vehicles for plowing operations are generally updated every winter season.

Table 4.2 evaluates the quality of the solution produced by the City of Dieppe for winter 2004–2005 by comparison with the two solution approaches. We report the percentage difference in the completion time for the four classes. The City of Dieppe produced the following solution for winter 2004–2005: $TMAX_{1D} = 2.8$, $TMAX_{2D} = 5.4$, $TMAX_{3D} = 5.2$, and $TMAX_{4D} = 5.6$. The city's completion time of class 3 is shorter than that of class 2 because the city allows the service of all arcs of class 3 in the higher class 2.

These results indicate that the two solution approaches can produce better routes than the city's method, in terms of service completion times. In fact, the parallel algorithm with the makespan objective (Scenario 3) reduces the time to service all arcs of A^1, A^2, and A^3 and the time required to service all arcs and return to the depot by more than 137%, 172%, 3%, and 8%, respectively, over the routing plan in use by the city. Moreover, the cluster first route second algorithm with the hierarchical objective and class upgrading possibilities (Scenario 2) cuts the service completion time of classes 1, 2, and 3 and the total completion time by more than 96%, 186%, 14%, and 13%, respectively, over the existing

Table 4.3: Percentage gaps: Makespan objective vs. hierarchical objective

Percentage gap (%)	Parallel algorithm	Cluster first route second algorithm
gap_1	0.0	0.6
gap_2	0.0	3.7
gap_3	–0.8	10.4
gap_4	–10.0	4.6

Table 4.4: Percentage gaps: No class upgrading vs. class upgrading

Percentage gap (%)	Parallel algorithm	Cluster first route second algorithm
gap_1	0.0	0.6
gap_2	4.0	0.0
gap_3	–4.9	6.3
gap_4	0.7	6.9

plan. The large decreases in completion times of higher priority classes result mainly from the parallel and cluster first route second routes better satisfying the hierarchy constraint.

We also analyzed the effects on service completion times of minimizing the hierarchical objective, allowing class upgrading possibilities, and satisfying the linear precedence relations between classes in each route. Table 4.3 presents, for every class $p = 1,\ldots, 4$, the relative difference gap_pbetween the value of $TMAX_p$ produced by the parallel (cluster first route second) algorithm with makespan objective (Scenario 3) and the value of $TMAX_p$ produced by the parallel (cluster first route second) algorithm with hierarchical objective (Scenario 1).

These results indicate that minimizing the hierarchical objective can reduce the service completion time of higher priority classes or increase the total completion time. With the parallel algorithm, the completion times of classes 1 and 2 remain the same because the makespan objective cannot be considered for solving subproblem (3.1)–(3.20) for these classes. Table 4.4 indicates, for every class $p = 1,\ldots, 4$, the relative difference gap_pbetween the value of $TMAX_p$ produced by the parallel (cluster first route second) algorithm with hierarchical objective and no class upgrading possibilities (Scenario 1) and the value of $TMAX_p$ produced by the parallel (cluster first route second) algorithm with hierarchical objective and class upgrading possibilities (Scenario 2).

Clearly, permitting class upgrading possibilities can reduce both the service completion time of higher priority classes and the total completion time. Table 4.5 illustrates the benefits of imposing linear precedence relations between classes in each route. The gap corresponds to the relative difference between the service completion time produced by the cluster first route second algorithm when the hierarchy constraint is relaxed (Scenario 4)

Table 4.5: Percentage gaps: No service hierarchy vs. service hierarchy

Percentage gap (%)	Scenario 1	Scenario 2	Scenario 3
gap_1	228.1	230.0	226.3
gap_2	169.3	169.3	159.7
gap_3	6.1	12.8	–3.9
gap_4	–0.8	6.0	–5.2

Table 4.6: Percentage gaps: Nine variants from (Perrier et al., 2006a) vs. cluster first route second algorithm (Scenario 2)

	Variant								
Percentage gap (%)	V1	V2	V3	V4	V5	V6	V7	V8	V9
gap_1	30.0	–0.1	141.0	30.0	182.3	266.5	243.6	–0.1	–0.1
gap_2	35.5	162.8	70.6	139.9	237.0	137.1	131.4	105.7	30.6
gap_3	24.2	5.4	18.1	2.8	32.1	5.2	26.9	7.4	19.2
gap_4	18.1	8.5	9.5	–1.5	42.9	5.3	20.6	1.5	10.6

and the service completion time produced by the cluster first route second algorithm with service hierarchy (Scenario 1, 2, or 3).

These results show, not surprisingly, that the linear precedence relations have a very positive influence on the service completion time of higher priority classes. In particular, completion times are reduced considerably for the first two classes.

In our previous paper (Perrier et al., 2006a), we proposed a cluster first route second method for the snow plow routing problem in the City of Dieppe. Nine variants were considered by changing the lower and upper bounds L and U on the total workload of each vehicle and the tolerance level on the total distance of class 3 road segments that can be serviced prior to higher-class road segments. According to these results, the cluster first route second algorithm with hierarchical objective and class upgrading possibilities (Scenario 2) presented here can produce a set of routes that mostly dominate the sets of routes produced by the other approach. Further comparisons with this method would thus be pointless. Table 4.6 presents, for every class $p = 1,\ldots, 4$, the relative difference gap_p between the value of $TMAX_p$ produced by a given variant and the value of $TMAX_p$ produced by the cluster first route second algorithm with the hierarchical objective and class upgrading possibilities (Scenario 2).

5 Conclusions

We have proposed a basic model and two solution approaches for the problem of vehicle routing in the context of snow plowing operations. This problem can be viewed as a multiple hierarchical Chinese postman problem with class upgrading possibilities and vehicle-road segment dependencies. The proposed model incorporates a wide variety of operational constraints and can also be customized to deal with many additional situations. The two constructive methods can produce sets of routes that dominate the existing routing plan of City of Dieppe with respect to service completion times in a few hours of computing time. This performance is satisfactory given the fact that the model need only be solved once every winter season. A faster solution approach would however be required so that the model can be used in real-time to determine the changes to be made following an equipment breakdown or weather change. Also, several extensions are still needed to make the model more useful in practice. For example, each road segment should be associated with three traversal times, which are possibly dependent on the vehicle type: the time required to plow the road segment, the time of deadheading the road segment if it has not yet been plowed, and the time of deadheading the road segment if it has already been plowed. Other extensions concern the requirement to service only a subset of arcs, thus leading to a multiple hierarchical rural postman problem, and the possibility of servicing multi-lane road segments requiring tandem service anywhere in the sequence, in order to reduce service completion times.

References

Alfa A.S., Liu D.Q. (1988) Postman routing problem in a hierarchical network. *Engineering Optimization* 114:127–138.

Assad A.A., Pearn W.L., Golden B.L. (1997) The capacitated Chinese postman problem: lower bounds and solvable cases. *American Journal of Mathematical and Management Sciences* 7:63–88.

Benavent E., Campos A., Corberán A., Mota E. (1990) The capacitated arc routing problem: a heuristic algorithm. *Qûestiió* 14, 107–22.

Benavent E., Soler. D. (1999) The directed rural postman problem with turn penalties. *Transportation Science* 33, 408–18.

Benson D.E., Bander J.L., White C.C. (1998) A planning and operational decision support system for winter storm maintenance in an ITS environment. In: *Proceedings of the 1998 IEEE International Conference on Intelligent Vehicles.* The University of Michigan: Ann Arbor, 673–675.

Bureau of Management Consulting. (1975) *Improving snow clearing effectiveness in Canadian municipalities.* Catalogue No T48-9/1975, Transportation Development Agency, Ministry of Transport, Canada.

Cabral E.A., Gendreau M., Ghiani G., Laporte G. (2004) Solving the hierarchical Chinese postman problem as a rural postman problem. *European Journal of Operational Research* 155, 44–50.

Campbell J.F., Langevin A. (2000) Roadway snow and ice control. In: Dror M, editor. *Arc routing: theory, solutions and applications.* Boston, MA: Kluwer, 389–418.

Chernak R., Kustiner L.E., Phillips L. (1990) The snowplow problem. *The UMAP Journal* 11, 241–50.

Corberán A., Martí R., Martínez E., Soler D. (2002) The rural postman problem on mixed graphs with turn penalties. *Computers & Operations Research* 29, 887–903.

Dror M., Stern H., Trudeau P. (1987) Postman tour on a graph with precedence relation on arcs. *Networks* 17, 283–94.

Fisher M., Jaikumar R. (1981) A generalized assignment heuristic for vehicle routing. *Networks* 11, 109–124.

Gélinas É. (1992) *Le problème du postier chinois avec contraintes générales de préséance.* M.Sc.A. Dissertation, École Polytechnique de Montréal, Canada.

Gendreau M., Laporte G., Yelle S. (1997) Efficient routing of service vehicles. *Engineering Optimization* 28, 263–71.

Ghiani G., Improta G. (2000) An algorithm for the hierarchical Chinese postman problem. *Operations Research Letters* 26, 27–32.

Guttridge A. (2004) *Rebuilding a winter maintenance service. The role of route optimization.* PSR Group Ltd, (http://www.psrgroup.on.ca/WPGuttridge.htm).

Haghani A., Qiao H. (2001) Decision support system for snow emergency vehicle routing. *Transportation Research Record* 1771, 172–78.

Haslam E., Wright J.R. (1991) Application of routing technologies to rural snow and ice control. *Transportation Research Record* 1304, 202–11.

Kandula P., Wright J.R. (1995) Optimal design of maintenance districts. *Transportation Research Record* 1509, 6–14.

Kandula P., Wright J.R. (1997) Designing network partitions to improve maintenance routing. *Journal of Infrastructure Systems* 3, 160–8.

Kaufmann A. (1967) *Graphs, dynamic programming and finite games.* New York: Academic Press.

Korteweg P., Volgenant T. (2006) On the hierarchical Chinese postman problem with linear ordered classes. *European Journal of Operational Research* 169, 41–52.

Lemieux P.F., Campagna L. (1984) The snow ploughing problem solved by a graph theory algorithm. *Civil Engineering Systems* 1, 337–41.

Marks H.D., Stricker R. (1971) Routing for public service vehicles. *Journal of the Urban Plannig and Development Division* 97, 165–78.

Miner W.M. (1997) Winter maintenance is often waiting game. *Better Roads* 67, 26–30.

Miner W.M., Bretherton S. (1996) Route optimization for winter maintenance activities. *1995–1996 field trial of roadway snow plowing.* Draft, 1996 TAC Annual Conference.

Moss C.R. (1970) *A routing methodology for snow plows and cindering trucks.* PennDOT project 68-5, Pennsylvania Transportation and Traffic Safety Center, Pennsylvania State University.

Muyldermans L., Cattrysse D., Van Oudheusden D., Lotan T. (2002) Districting for salt spreading operations. *European Journal of Operational Research* 139, 521–32.

Muyldermans L., Cattrysse D., Van Oudheusden D. (2003) District design for arc-routing applications. *Journal of the Operational Research Society* 54, 1209–21.

Perrier N., Amaya A., Langevin A., Cormier G. (2006a) Improving snow removal operations using operations research: a case study. *Proceedings of the International Conference on Information Systems, Logistics and Supply Chain, ILS2006.* Lyon, France, May 15–17.

Perrier N., Langevin A., Campbell J.F. (2005) A survey of models and algorithms for winter road maintenance Part IV: vehicle routing and fleet sizing for plowing and snow disposal. *Computers & Operations Research* doi:10.1016/j.cor.2005.05.008.

Perrier N., Langevin A., Campbell J.F. (2006b) A survey of models and algorithms for winter road maintenance Part I: system design for spreading and plowing. *Computers & Operations Research* 33, 209–238.

Perrier N., Langevin A., Campbell J.F. (2006c) A survey of models and algorithms for winter road maintenance Part II: system design for snow disposal. *Computers & Operations Research* 33, 239–262.

Perrier N., Langevin A., Campbell J.F. (2006d) The sector design and assignment problem for snow disposal operations. *Les Cahiers du GERAD* G–2006–34 (forthcoming).

Robinson J.D., Ogawa L.S., Frickenstein S.G. (1990) The two-snowplow routing problem. *The UMAP Journal* 11, 251–9.

Salim M.D., Strauss T., Emch M. (2002a) A GIS-integrated intelligent system for optimization of asset management for maintenance of roads and bridges. *Lecture Notes in Computer Science* 2358, 628–37.

Salim M.D., Timmerman M.A., Strauss T., Emch M.E. (2002b) *Artificial-intelligence-based optimization of the management of snow removal assets and resources.* Report No. Y00-162, Iowa State University, (http://www.ctre.iastate.edu/reports/Snowplow Report.pdf).

Stricker R. (1970) *Public sector vehicle routing: the Chinese postman problem.* M.Sc. dissertation, Massachusetts Institute of Technology.

Wang J.Y. (1992) *Computer-aided system for planning efficient routes.* Ph.D. dissertation, Purdue University, West Lafayette, Indiana.

Wang J.Y., Kandula P., Wright J.R.(1995) Evaluation of computer-generated routes for improved snow and ice control. *Transportation Research Record* 1509, 15–21.

Wang J.Y., Wright J.R. (1994) Interactive design of service routes. *Journal of Transportation Engineering* 120, 897–913.