Arc routing with precedence constraints: An application to snow plowing operations

Anders H. Gundersen[1], Magnus Johansen[1], Benjamin S. Kjær[1], Henrik Andersson[1], and Magnus Stålhane[1]

Norwegian University of Science and Technology,
Department of Industrial Economics and Technology Management
Alfred Getz veg 3, Trondheim, Norway

Abstract. In this paper we present an arc routing problem with precedence constraints, with a focus on its application to snow plowing operations in Norway. The problem studied considers the clearing of snow from a network of roads, where there exists precedence relations between the driving lanes and the sidewalks. The goal is to minimize the total time it takes for a heterogeneous fleet of vehicles to clear all the snow from the road network. We describe a mathematical model of the problem and present symmetry breaking constraints to improve the computational performance. We present a computational study where the performance of the model is tested. Further, we study the effect of forbidding or penalizing U-turns along the route, something the snow plowing vehicles struggle to do. The computational experiments show that it is possible to generate solutions without U-turns with only a marginal increase in the objective value.

Keywords: Arc routing; Snow plowing; vehicle routing;

1 Introduction

In this paper we study an extension of the capacitated arc routing problem ([3]), where there exist precedence constraints on the traversal of given pairs of arcs, and where U-turns are undesirable. The problem is inspired by the planning of snow plowing operations in urban areas of Norway, during, or immediately after, a snowfall.

The problem under consideration can be described as clearing all the snow from a network of roads. This network consists of a set of road segments, where each segment consists of one or more lanes, in one or two directions, and/or one or more sidewalks. Whenever the characteristics of the road changes, such as at intersections, places where two lanes merge into one, or one splits into two, ramps on or off a highway, and so on, we assume that one road segment ends and another begins. We define the term *lane* for each driving lane on a road segment, and *sidewalk* for a sidewalk associated with the road segment. A map showing a small area of downtown Trondheim together with the corresponding road network is shown in Figure 1.

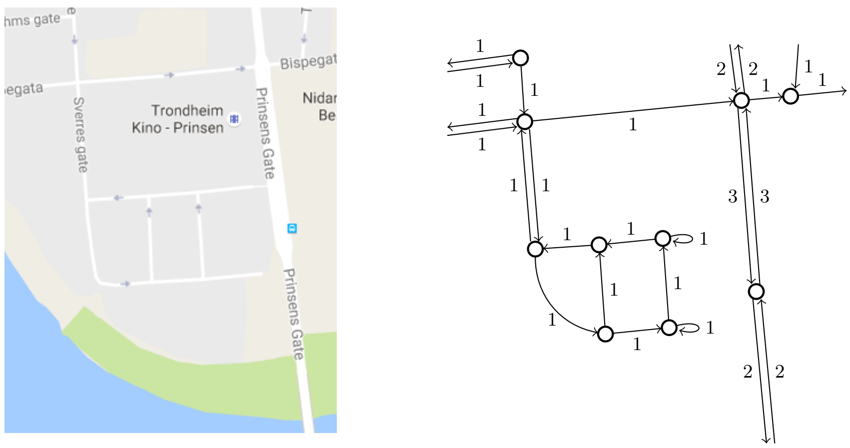

Fig. 1: Map and the corresponding road network from a small part of downtown Trondheim. Arrows and numbers indicate the driving direction and the number of lanes in each direction, respectively. Sidewalks and pedestrian pathways are not included.

To service the road network, two types of vehicles are available: heavy trucks for plowing the lanes, and smaller vehicles for clearing the sidewalks. All vehicles are associated with one depot where they begin and end their route. Each heavy truck has a large plow attached to the front, capable of clearing one lane at a time. However, when clearing a lane, the snow is pushed to the right hand side of the vehicle, pushing it from the middle toward the side of the road segment. Therefore, if there are multiple lanes in the same direction, a general requirement is the need to service the innermost lane first. Where there is a sidewalk beside a lane, some, or all, of the snow may be shoved on to it, and therefore the sidewalk must be serviced after the corresponding lane(s). This creates a precedence relation, all lanes must be cleared before the sidewalk can be cleared. The vehicles plowing the sidewalks are smaller and cannot be used to plow the lanes. However, the smaller vehicles can traverse the lanes without plowing. This is often a necessity, as the network of sidewalks may not be connected. Sometimes the only way to make sure that the sidewalks are serviced after the lanes, is for the vehicles to wait for the truck to plow the specific lane. It is therefore allowed for vehicles to wait at intersections.

Due to the large size of the heavy trucks plowing the lanes, U-turns is a rather problematic, time consuming, and often impossible maneuver for them in urban areas. For the smaller vehicles clearings the sidewalk, U-turns are possible, but time consuming since it has to cross all the driving lanes to reach the sidewalk on the opposite side of the road. Information regarding extra time spent at an intersection when taking a U-turn, and which intersections where U-turns are prohibited, is assumed to be known to the planner.

The goal is to determine one route for each vehicle so that the total time it takes to clear the snow from the entire road network is minimized, i.e. minimizing the makespan, while adhering to the precedence constraints between the sidewalks and the lanes, and trying to avoid U-turns. A route for a given vehicle is a sequence of road segments, and information of whether the road segment is plowed or deadheaded (driving a road segment without plowing). Note that since the vehicles clearing the lanes cannot clear the sidewalks and vice versa, the route only needs to keep track of which road segments are traversed for each vehicle. In addition, we need to know the time each lane and sidewalk is plowed, to ensure that the precedence constrains are respected at each road segment.

A similar problem to the one studied in this paper was introduced in [11] which studies an arc routing problem for snow plowing operations, where multiple lanes going in the same direction have to be serviced at the same time. The problem consists set of homogeneous vehicles, where each vehicle has a maximum number of arcs it can traverse, and the objective is to minimize the makespan. The problem is solved by a two-phase Adaptive Large Neighborhood Search heuristic (ALNS). Another similar snow plowing problem is studied in [2], where the time it takes to deadhead an arc before plowing it is longer than after it has been serviced. The problem is defined for a single vehicle, and the objective is to minimize costs associated with the route. To solve the problem they introduce a local search heuristic. In contrast to the problem studied in this paper, [11] and [2] consider a homogeneous fleet of vehicles, and a single vehicle, respectively.

In [4] a vehicle routing problem for snow plowing operations that considers heterogeneous vehicle fleets is introduced. The presented model is designed to consider both plowing and salt-spreading operations. The problem is defined on a mixed multigraph representing unidirectional and bidirectional plow jobs. Unlike the problem studied in this paper they consider replenishment of consumed resources such as fuel and salt along a route. However, they do not consider any type of temporal dependencies, such as precedence, between the different vehicles. They compare a MIP model, a constraint programming model, and a two-phase heuristic procedure for solving the problem.

Another paper that considers a heterogeneous fleet for snow plowing operations is [6], which studies a problem where the set of arcs is divided into non-overlapping subsets called priority classes, and each class can only be serviced by a subset of the available vehicles. The vehicles can vary both with respect to size, and service- and deadheading speed. The problem includes penalties on U-turns, and synchronization of plowing operations. A mathematical model, and two constructive methods are presented to solve the problems. A difference between [6] and this paper is that the model in [6] assumes that all arcs in one priority set is serviced before the first arc in a lowered priority set (though the model allows arcs to be upgraded to a higher priority class), while the problem studied in this paper has precedence relations between pairs of arcs. Thus, the mathematical formulation presented below explicitly models the service time of each arc, and allows waiting times before an arc is serviced, while in [6] they compare the completion time of the last arc in one priority set to the first arc in another set.

For a complete overview of earlier papers studying optimization of the routing of snow plowing vehicles, we refer to the survey presented in [10], while for a thorough review of other winter road maintenance operations we refer to [7, 8, 9]. For a comprehensive discussion of the capacitated arc routing problem and its variants, we refer to [1].

The purpose of this paper is to study a new variant of the capacitated arc routing problem, where there is precedence between when pairs of arcs may be serviced. We present a new mathematical formulation of the problem, and several ways in which the computational efficiency of the model may be improved. Further, we conduct a computational study to inspect the effect of the suggested improvements to the model, and to study the effect of U-turns on the solution quality of the problem.

The remainder of the paper is organized as follows: In Section 2 we present a mathematical model of the problem, before conducting a computational study of this model in Section 3. Finally, we give some concluding remarks in Section 4.

2 Mathematical Model

The most intuitive way of formulating the arc routing problem with precedence relations is to extend the capacitated arc routing problem presented by [3], with the necessary sets and constraints needed for plowing operations. The formulation is a mixed integer program (MIP).

Let $G = (\mathcal{V}, \mathcal{A})$ be a directed multigraph where the vertex set $\mathcal{V}$ represents the nodes in the road network (geographic locations with changes in service criteria), and the arc set $\mathcal{A}$ represents the lanes and sidewalks. If there is a lane and a sidewalk between the same two nodes, this is represented with two separate arcs. If there are two lanes in one direction, this is represented by just one arc. Figure 2 illustrates an example of such a directed multigraph.

We have a set of vehicles $\mathcal{K}$, which is separated into two fleets. Let $\mathcal{K}^L \subset \mathcal{K}$ be the set of plowing trucks for the lanes, and $\mathcal{K}^S \subset \mathcal{K}$ be the set of vehicles that service the sidewalks. The trucks can only drive on and service lanes, while the vehicles for sidewalks can drive on both lanes and sidewalks, but only service the sidewalks. Let $\mathcal{A}^S \subseteq \mathcal{A}$ represent the arcs that the vehicles for sidewalks can traverse, and $\hat{\mathcal{A}}^S \subseteq \mathcal{A}^S$ be the set of arcs that have sidewalks with service needs. Similarly, let $\mathcal{A}^L \subseteq \mathcal{A}^S$ represent the arcs that can be traversed by the plowing trucks, and $\hat{\mathcal{A}}^L \subseteq \mathcal{A}^L$ be the set of lanes which have to be serviced.

To fulfill the service demands on different arcs, each vehicle $k \in \mathcal{K}$ has to drive a defined route. Each route starts and ends at the depot, and each traversal of an arc corresponds to a leg, numbered by n, in a route. Since a route can pass through the depot, D, several times, we define the vertices $o(k)$ and $d(k)$ as the artificial origin and destination of vehicle k, which are only connected to the original depot D. An upper bound on the number of legs included in a route

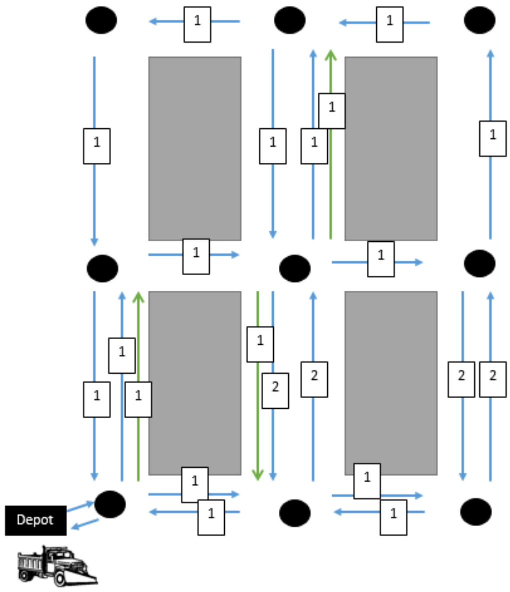

Fig. 2: An illustration of a directed multigraph. The black dots represent the nodes, while the blue and green arrows represent the lanes and sidewalks, respectively. The number on each arrow indicates how many lanes and sidewalks there are on that arc.

is given by $\overline{n}$, and we define the set of possible legs, $\mathcal{N} = \{1, ..., \overline{n}\}$. As seen in Figure 2 an arc can have a number of lanes or sidewalks in the same direction. Let R^L_{ij} and R^S_{ij} be the number of lanes and sidewalks on arc (i, j), respectively. Further, let T_{kij} be the time vehicle k uses to service arc (i, j), and T^D_{kij} be the time vehicle k uses to deadhead arc (i, j). In general, the plowing trucks use shorter time to service an arc (i, j), compared with the smaller vehicles for service the corresponding sidewalk, therefore $T_{kij} \leq T_{\hat{k}ij}$, given $k \in \mathcal{K}^L$ and $\hat{k} \in \mathcal{K}^S$. This means that we only need to consider the start of service a lane and the corresponding sidewalk in the precedence relation. Let T^{Max} be an upper bound on the maximum time a vehicle can use on its route.

To penalize U-turns, T^U_{kij} is the time it takes to do a U-turn from arc (j, i) to arc (i, j) for vehicle k, and u_{kijn} is a binary variable that states whether vehicle k made a U-turn before it traversed arc (i, j) as leg n, or not. Let the binary variable x_{kijn} be 1 if vehicle k service arc (i, j) as the n^{th} of its route, and 0 otherwise. Similarly, let y_{kijn} be 1 if arc (i, j) is traversed by vehicle k and appears as the n^{th} leg of the route while deadheading, and 0 otherwise. The variable τ_{kn} tracks the end time of service or traversal of leg n in the route of vehicle k, while t^L_{ij} and t^S_{ij} tracks the end time of service of the lanes and sidewalks on arc (i, j), respectively. Finally, the variable t^{MS} defines the total makespan of the solution. For shorthand notation, we denote $\mathcal{A}_k$ as the set of arcs vehicle k can traverse, and for a given vehicle k the sets $\delta_k^+(i) = \{j|(i,j) \in \mathcal{A}_k\}$, and $\delta_k^-(i) = \{j|(j,i) \in \mathcal{A}_k\}$. Using this notation the mathematical model of the problem can be described as follows:

$$\min t^{MS} \tag{1}$$

s.t.

$$x_{ko(k)D1} + y_{ko(k)D1} = 1 \qquad k \in \mathcal{K} \tag{2}$$

$$\sum_{n\in\mathcal{N}} \Big(x_{kDd(k)n} + y_{kDd(k)n}\Big) = 1 \qquad k \in \mathcal{K} \tag{3}$$

$$\sum_{k\in\mathcal{K}^L} \sum_{n\in\mathcal{N}} x_{kijn} = R_{ij}^L \qquad (i,j) \in \hat{\mathcal{A}}^L \tag{4}$$

$$\sum_{k\in\mathcal{K}^S} \sum_{n\in\mathcal{N}} x_{kijn} = R_{ij}^S \qquad (i,j) \in \hat{\mathcal{A}}^S \tag{5}$$

$$\sum_{i\in\delta_k^-(j)} \Big(x_{kijn} + y_{kijn}\Big) - \sum_{i\in\delta_k^+(j)} \Big(x_{kji(n+1)} + y_{kji(n+1)}\Big) = 0 \qquad k \in \mathcal{K}, j \in \mathcal{V}\backslash\{o(k), d(k)\}, n \in \mathcal{N}\big|n < \overline{n} \tag{6}$$

$$\sum_{(i,j)\in\mathcal{A}_k} \Big(x_{kijn} + y_{kijn}\Big) \leq 1 \qquad k \in \mathcal{K}, n \in \mathcal{N} \tag{7}$$

$$x_{kij(n-1)} + y_{kij(n-1)} + x_{kjin} + y_{kjin} \leq u_{kjin} + 1 \qquad k \in \mathcal{K}, (i,j) \in \mathcal{A}_k, n \in \mathcal{N}\big|(j,i) \in \mathcal{A}_k, n > 1 \tag{8}$$

$$\tau_{kn} - \tau_{k(n-1)} \geq \sum_{(i,j)\in\mathcal{A}_k} \Big(T_{kij}x_{kijn} + T_{kij}^D y_{kijn} + T_{kij}^U u_{kijn}\Big) \qquad k \in \mathcal{K}, n \in \mathcal{N}|n > 1 \tag{9}$$

$$\tau_{kn} - T^{Max}\Big(1 - x_{kijn}\Big) \leq t_{ij}^L \qquad k \in \mathcal{K}, (i,j) \in \hat{\mathcal{A}}^L, n \in \mathcal{N} \tag{10}$$

$$\tau_{kn} - T^{Max}\Big(1 - x_{kijn}\Big) \leq t_{ij}^S \qquad k \in \mathcal{K}, (i,j) \in \hat{\mathcal{A}}^S, n \in \mathcal{N} \tag{11}$$

$$\tau_{kn} \leq T^{Max} \qquad k \in \mathcal{K}, n \in \mathcal{N} \tag{12}$$

$$t_{ij}^L - T_{kij} \leq t_{ij}^S - T_{\hat{k}ij} \qquad k \in \mathcal{K}^L, \hat{k} \in \mathcal{K}^S, (i,j) \in \hat{\mathcal{A}}^L|(i,j) \in \hat{\mathcal{A}}^S \tag{13}$$

$$\tau_{kn} \leq t^{MS} \qquad k \in \mathcal{K}, n \in \mathcal{N} \tag{14}$$

$$x_{kijn} \in \{0,1\} \qquad k \in \mathcal{K}, (i,j) \in \mathcal{A}_k, n \in \mathcal{N} \tag{15}$$

$$y_{kijn} \in \{0,1\} \qquad k \in \mathcal{K}, (i,j) \in \mathcal{A}_k, n \in \mathcal{N} \tag{16}$$

$$u_{kijn} \in \{0,1\} \qquad k \in \mathcal{K}, (i,j) \in \mathcal{A}_k, n \in \mathcal{N} \quad (17)$$

$$\tau_{kn} \geq 0 \qquad k \in \mathcal{K}, n \in \mathcal{N} \quad (18)$$

$$t^L_{ij} \geq 0 \qquad (i,j) \in \hat{\mathcal{A}}^L \quad (19)$$

$$t^S_{ij} \geq 0 \qquad (i,j) \in \hat{\mathcal{A}}^S \quad (20)$$

The objective function (1) is to minimize the total makespan of the solution. Constraints (2) and (3) state that each route starts and ends in each vehicle's depot, while constraints (4) and (5) ensure that all arcs with demands are serviced. Further, constraints (6) make sure that the plowing routes are connected. Each vehicle can only traverse one arc in each leg of its route; this is taken care of by constraints (7). Constraints (8) ensure the U-turn variable to be 1 if a vehicle take a U-turn, while constraints (9) provide that the vehicles behave consistent according to time. Constraints (10) and (11) connect the time variables, and constraints (12) ensure that the time of a route does not exceed the upper bound. Constraints (13) assure that the precedence requirements between the corresponding lanes and sidewalks hold. Finally, constraints (14) ensure that no traversal time of a given leg for a given vehicle can be larger than the makespan of the solution, while constraints (15)–(20) define the domain of the variables in the model.

2.1 Improvements to the model

As the MIP model described has two homogeneous vehicle fleets, the model can produce several mathematically different solutions which are practically equivalent by altering which vehicle drives which route in a solution. E.g. given a fleet of three homogeneous vehicles, and a solution of three vehicle routes, there exist six ways to assign routes to vehicles which are all practically equivalent (since the vehicles are identical). To reduce the number of symmetric solutions we introduce two sets of symmetry breaking constraints based on lexicographic ordering of the vehicle routes based on the consumption of some resource accumulated along the route. Given that the resource consumption along each route is unique, this will remove all permutations except for one, while in the case where the resource consumption along two or more routes are equal, the lexicographic order is arbitrary, and some (or all) symmetry may remain in the problem. However, ffor both sets of constraints there exist (at least one) lexicographic ordering of the routes, and thus we are ensured that all practically different routing solutions are still present in the model. For more details on lexicographic symmetry breaking constraints we refer to [5].

The first set of symmetry breaking constraints proposed are based on the number of arcs traversed by each vehicle along its route. We here formulate constraints to force the vehicle with the lowest index number to service at least as many arcs as the vehicle with the second lowest index and so on. The constraints are given in constraints (21) and (22).

$$\sum_{n\in\mathcal{N}}\sum_{(i,j)\in\hat{\mathcal{A}}^L} x_{kijn} \geq \sum_{n\in\mathcal{N}}\sum_{(i,j)\in\hat{\mathcal{A}}^L} x_{(k+1)ijn} \qquad k\in\mathcal{K}^L \big| k<|\mathcal{K}^L| \tag{21}$$

$$\sum_{n\in\mathcal{N}}\sum_{(i,j)\in\hat{\mathcal{A}}^S} x_{kijn} \geq \sum_{n\in\mathcal{N}}\sum_{(i,j)\in\hat{\mathcal{A}}^S} x_{(k+1)ijn} \qquad k\in\mathcal{K}^S \big| k<|\mathcal{K}^S| \tag{22}$$

The second set of symmetry breaking constraints proposed are based on the total duration of the route of each vehicle. We here formulate constraints to force the vehicle with the lowest index number to drive a route with at least the same duration as the vehicle with the second lowest index and so on. The constraints are given in constraints (23) and (24).

$$\sum_{n\in\mathcal{N}}\sum_{(i,j)\in\hat{\mathcal{A}}^L} \big(T_{kij}x_{kijn} - T_{(k+1)ij}x_{(k+1)ijn}\big) \geq 0 \qquad k\in\mathcal{K}^L \big| k<|\mathcal{K}^L| \tag{23}$$

$$\sum_{n\in\mathcal{N}}\sum_{(i,j)\in\hat{\mathcal{A}}^S} \big(T_{kij}x_{kijn} - T_{(k+1)ij}x_{(k+1)ijn}\big) \geq 0 \qquad k\in\mathcal{K}^S \big| k<|\mathcal{K}^S| \tag{24}$$

Note that these two sets of lexicographic ordered constraints cannot be implemented at the same time in the model, and still guarantee optimality. It is hard to say which symmetry breaking constraints are the best and how well they will perform. This is further studied in Section 3.

To reduce the solution time when the MIP model is solved by a commercial software, we may tighten the range of the variables, thus improving the lower bound - and thereby decrease the solution space for the relaxed formulation. We know that the earliest time an arc is serviced is the shortest time it takes to travel from the depot to the start of the arc, plus the service time of the arc. We therefore introduce the parameters α_j^L and α_j^S, which state the shortest travel time from the depot to node j for vehicles in $\mathcal{K}^L$ and $\mathcal{K}^S$, and obtain the following constraints:

$$t_{ij}^L \geq min_{k\in\mathcal{K}^L}\{\alpha_i^L + T_{kij}\} \qquad (i,j)\in\mathcal{A}^L \tag{25}$$

$$t_{ij}^S \geq min_{k\in\mathcal{K}^S}\{\alpha_i^S + T_{kij}\} \qquad (i,j)\in\mathcal{A}^S \tag{26}$$

Constraints (19) and (20) in the initial formulation can now be replaced with constraints (25) and (26), which improves the lower bound of the time variables, and likely reduce the solution time.

3 Computational Study

In this section we present a computational study of the mathematical model described in Section 2. We first present the set of test instances used, before testing the computational effect of adding the different improvements of the model suggested in Section 2.1. Finally, we study the impact of adding U-turn penalties or forbidding U-turns in the model, both when it comes to computational efficiency and solution quality.

The model has been implemented in the commercial optimization software Xpress Optimization Suite and run on a computer with a 3.4 GHz Intel Core i7 processor and 32 GB of RAM, running Windows 10 Education. Version 1.24.08 of Xpress IVE was used, with version 3.10.0 of Xpress Mosel, and version 28.01.04 of Xpress Optimizer.

3.1 Test instances

The test instances are based on fictitious road networks that are set to mimic those found in urban areas. These generally involve road segments with one lane in each direction, and 4-way intersections where two perpendicular roads meet. Additionally, there often exists a sidewalk on one or both sides of the traffic lanes. The numerical values of the traversing times are based on proportionality, such that there is a difference between road segments, while they all lie in the same order of magnitude. All instances have an average traversal time of $5 - 6$ time units per arc. Equally, the service time for a sidewalk is longer than that of the associated lane, if such a lane exist.

A set of 25 test instances have been generated to test the model. These are grouped into test instances $1 - 10$, presented in Table 1, and $11 - 25$, presented in Table 2. For each instance, Table 1 and 2 presents the number of trucks ($|\mathcal{K}^L|$) and smaller vehicles ($|\mathcal{K}^S|$) in each instance, as well as the number of nodes (# Nodes), lanes (# Lanes) and sidewalks (# SW) in the graph representing the road network. All arcs need to be serviced only once. Further, the number of arcs with precedence constraints (# Prec), the upper bound on the number of legs used (# Legs) and the maximum time a vehicle can use on its route (T^{Max}) is given. Test set 1 is smaller and only used to test the symmetry breaking constraints and improved bounds, while test set 2 is larger, and used to test the capabilities of the model.

3.2 Testing the effect of the suggested improvements to the model

To compare the different variations of the models, we have run each of the test instances in Table 1, without considering U-turn penalties, for a maximum of $1,000$ seconds. The results can be found in Table 3. The column *Original model* is the mathematical model with none of the suggested improvements from Section 2.1. In *Symmetry Breaking 1 (SB1)* we have included constraints (21) and (22) to the model, while *Symmetry Breaking 2 (SB2)* includes constraints (23) and (24). In *Increased Bound (IB)*, the lower bound of the time variables have been increased. That is, we have replaced constraints (19) – (20) with (25) – (26). In addition we have tested combining each of the symmetry breaking constraints with the increased bound. For each variant of the model we report the computing time (Time) in seconds and the optimality gap (Gap) in percent.

Table 1: Characteristics of the test instances in *Test set 1*.

Instance	$\vert\mathcal{K}^L\vert$	$\vert\mathcal{K}^S\vert$	# Nodes	# Lanes	# SW	# Prec	# Legs	T^{Max}
1	1	1	6	10	3	3	17	50
2	2	1	6	10	3	3	17	50
3	2	1	8	18	5	5	17	50
4	3	2	13	32	16	14	17	90
5	2	2	17	42	24	18	28	150
6	3	2	17	42	24	18	28	150
7	3	2	20	48	29	21	35	150
8	4	2	20	48	29	21	35	150
9	3	3	30	78	42	30	45	250
10	4	3	30	78	42	30	45	250

Table 2: Characteristics of the test instances in *Test set 2*.

Instance	$\vert\mathcal{K}^L\vert$	$\vert\mathcal{K}^S\vert$	# Nodes	# Lanes	# SW	# Prec	# Legs	T^{Max}
11	1	1	7	14	3	3	17	50
12	2	1	7	14	3	3	17	50
13	2	1	11	24	8	6	17	90
14	2	2	11	24	8	6	17	90
15	2	2	15	36	24	18	22	90
16	3	2	15	36	24	18	22	90
17	2	2	20	48	29	21	35	150
18	3	3	20	48	29	21	35	150
19	2	2	30	78	42	30	45	250
20	3	2	30	78	42	30	45	250
21	4	2	30	78	42	30	45	250
22	3	2	40	112	60	46	65	350
23	3	3	40	112	60	46	65	350
24	4	2	40	112	60	46	65	350
25	4	3	40	112	60	46	65	350

Table 3: Results from running test instance 1–10 with different symmetry breaking constraints and valid inequalities added to the model formulation.

	Original model		SB1		SB2		Inc. Bound		SB1 + IB		SB2 + IB	
Instance	Time(s)	Gap(%)	Time(s)	Gap(%)	Time(s)	Gap(%)	Time(s)	Gap(%)	Time(s)	Gap(%)	Time(s)	Gap(%)
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0.2	0	0	0	0.2	0	0.2	0	0.2	0	0.1	0
3	1.1	0	0.8	0	0.4	0	0.8	0	0.1	0	0.8	0
4	12.5	0	18.3	0	13.3	0	10.8	0	11.8	0	9.5	0
5	938.3	0	120.4	0	59.1	0	181	0	79.7	0	118.7	0
6	526.1	0	304.7	0	157.9	0	761	0	215.8	0	262.6	0
7	1000	0.56	694.1	0	1000	0.56	1000	0.56	1000	0.56	1000	0.56
8	332.6	0	235.5	0	1000	2.38	1000	2.03	759.5	0	200.3	0
9	480.6	0	1000	7.26	1000	3.36	1000	3.36	1000	6.5	1000	14.18
10	1000	38.51	1000	27.78	1000	33.09	1000	27.78	1000	32.59	1000	29.46
Average	429.14	3.91	337.38	3.50	423.09	3.94	495.38	3.37	406.71	3.97	359.2	4.42

On average, *Symmetry Breaking 1* yields the best results, with respect to computational time, and the second best with respect to the optimality gap. It is also the only improved model that solves 8 out of 10 instances to optimality within $1,000$ seconds. The *Increased Bound* model has the lowest gap on average and although not performing better than the original model in all test instances, we conclude that the initial bound on the time variables in general contribute positively to better bounds on the obtained solutions. It shall be noted, that when merging *Increased Bound* and *Symmetry Breaking 1*, this constitutes a model that, on average, perform worse than each of them applied separately. The original model, which, on average, perform among the worst models with respect to solution time, was the only model to prove optimality on instance 9 within the $1,000$ seconds limit.

We conclude that the *Symmetry Breaking 1* model is the best performing, and have chosen to continue with these lexicographically ordered symmetry breaking constraints. In the analysis that follows we refer to the *Symmetry Breaking 1* model as the *Basic model.* try to minimize the fleet size, it is not clever to say that we have an infinite fleet size in the initiate state. It is better to start with a realistic amount of vehicles, and increase iterative if it should not exist any feasible solution for the given size within the maximum time for a schedule.

3.3 Effect of penalizing or forbidding U-turns

We now study how forbidding or penalizing U-turns affect the computational performance and the solution quality of the test instances. When penalizing a U-turn the cost is given in time units, which in this case is set to 2, a bit less than half of the average service time for a lane. In the case where U-turns are forbidden, they are only forbidden for the plowing trucks. Since the vehicles plowing the sidewalks are smaller, they are allowed to make U-turns, which corresponds to crossing a lane after plowing a sidewalk, to reach the sidewalk on the other side. Although allowed, a penalty cost of 2 time units is given for this maneuver. The results, and a comparison with the *Basic model* are presented in Table 4 where all instances have been run for $1,000$ seconds. For each version of the model, the computational time (Time) in seconds as well as the optimality gap (Gap) in percent, is given. For the instances with an n/a in the Gap column, no feasible solution was found within the time limit.

The results show that the computational time is roughly the same for all three versions of the problem. However, while the *Basic model* is able to provide a feasible solution within the time limit on all but one instance (24), penalizing and forbidding U-turns do not provide a feasible solution within the time limit on 3 and 4 instances, respectively. Forbidding U-turns reduce the number of feasible solutions to the problem which may explain why Xpress struggles more to find feasible solutions in this case. In case of U-turn penalties the explanation may be related to increased fractionality in the solutions, since the u-variables now indirectly affect the objective value. It is also interesting to note that for instance 24, Xpress is able to find a feasible solution when adding U-turn penalties to the model, while no feasible solution is found for the two other versions.

Table 4: Comparison of the computational performance of forbidding U-turns and penalizing U-turns to the *Basic model*.

	Basic model		With U-turn penalty		U-turns forbidden	
Instance	Time(s)	Gap(%)	Time(s)	Gap(%)	Time(s)	Gap(%)
11	0	0.00	0	0.00	0	0.00
12	0	0.00	1	0.00	1	0.00
13	1	0.00	2	0.00	1	0.00
14	3	0.00	11	0.00	9	0.00
15	10	0.00	32	0.00	102	0.00
16	13	0.00	102	0.00	43	0.00
17	397	0.00	1000	3.00	219	0.00
18	1000	3.48	1000	11.42	1000	34.01
19	1000	0.29	1000	4.14	1000	10.62
20	1000	4.55	1000	19.36	1000	22.50
21	1000	1.09	1000	25.40	1000	1.76
22	1000	21.89	1000	n/a	1000	n/a
23	1000	49.12	1000	n/a	1000	n/a
24	1000	n/a	1000	42.08	1000	n/a
25	1000	55.13	1000	n/a	1000	n/a
Average	562	9.68	610	8.78	558	6.26

Table 5 compares the optimal solution of the *Basic model* on instance 11–21 to the optimal solutions of the model when forbidding and penalizing U-turns, respectively. For each instance and version of the model we give the change in the number of U-turns performed by trucks (Δ lanes), by small vehicles (Δ SW), and the change in the makespan (Δ Makespan). The optimal solutions to each instance was obtained by running the model for several days, however, for the instances marked with a * and **, we only managed to obtain results within 2 % and 6 % of optimum, respectively.

As shown in Table 5, on average, the number of U-turns performed by the two vehicle fleets is reduced quite significantly both when penalizing and forbidding U-turns. This comes at an average increase in the total makespan of less than three time units. Even in the worst case, the increase in the makespan is only six time units (instance 18), equal to the maximum traversal time of a single arc. An interesting anomaly in the results in the case of penalizing U-turns is instance 16, where the number of U-turns performed by the larger trucks increase. This may be explained by the fact that any vehicle that drives a route that is significantly shorter than the route defining the makespan, may perform U-turns without it affecting the objective value. However, the total makespan of the instance is increased by 5, indicating that the route defining the makespan has changed from the *Basic model*.

The results presented in Table 5 indicate that it is possible to design vehicle routes for the plowing trucks that does not perform any U-turns, without significantly increasing the total time it takes to clear the road network of snow.

Table 5: Comparing how the U-turn constraints influence the makespan and the number and U-turns in the different models.

	Time penalty for U-turns			**U-turns forbidden**		
Instance	*Δ lanes*	*Δ SW*	*Δ Makespan*	*Δ lanes*	*Δ SW*	*Δ Makespan*
11	-2	0	0	-2	-2	0
12	-1	0	2	-4	0	2
13	-2	0	4	-6	0	4
14	-7	0	0	-7	2	0
15	3	-7	5	-8	-7	5
16	1	-4	5	-11	-4	5
17	-7	-1	0	-7	-3	0
18	-3	0	4**	-5	-1	6**
19	-16	-10	1	-16	-3	1
20	-14	0	3	-28	0	4*
21	-13	-2	3*	-27	-2	3*
Average	-5.55	-2.18	2.45	-11.00	-1.82	2.73

4 Concluding remarks

In this paper we have studied an arc routing problem inspired by a snow plowing problem faced by planners in Norway. The objective is to minimize the total time of clearing a road network of snow, but it is complicated by the fact that there is precedence between pairs of arcs in the network. To solve the problem we have introduced a mathematical model, and suggested symmetry breaking constraints to improve the computational performance when solving the model using commercial software. In addition, we have tested the effect of penalizing or forbidding U-turns in the model, something which is difficult for the snow plowing vehicles to do in many urban areas. The results of these tests show that we can eliminate the need for U-turns in the vehicle routes, with only a marginal increase in the total time it takes to clear the road network.

Since the mathematical model presented in this paper is unable to solve realistic instances of the problem, future research should look into heuristic solution methods for the problem. In addition, it would be interesting to test the model presented in this paper on graphs generated from real road networks to corroborate the findings regarding the influence of U-turns on the makespan of a solution.

Bibliography

[1] Corberán, Á. and Laporte, G. (2013). *Arc routing: problems, methods, and applications*. SIAM.

[2] Dussault, B., Golden, B., Groër, C., and Wasil, E. (2013). Plowing with precedence: A variant of the windy postman problem. *Computers & Operations Research*, 40(4):1047–1059.

[3] Golden, B. L. and Wong, R. T. (1981). Capacitated arc routing problems. *Networks*, 11(3):305–315.

[4] Kinable, J., van Hoeve, W.-J., and Smith, S. F. (2016). Optimization models for a real-world snow plow routing problem. In Quimper, C.-G., editor, *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*, Lecture Notes in Computer Science, pages 229–245. Springer.

[5] Margot, F. (2010). Symmetry in integer linear programming. In Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., Rinaldi, G., and Wolsey, L. A., editors, *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pages 647–686. Springer Berlin Heidelberg, Berlin, Heidelberg.

[6] Perrier, N., Langevin, A., and Amaya, C.-A. (2008). Vehicle routing for urban snow plowing operations. *Transportation Science*, 42(1):44–56.

[7] Perrier, N., Langevin, A., and Campbell, J. F. (2006a). A survey of models and algorithms for winter road maintenance. part I: system design for spreading and plowing. *Computers & Operations Research*, 33(1):209–238.

[8] Perrier, N., Langevin, A., and Campbell, J. F. (2006b). A survey of models and algorithms for winter road maintenance. part II: system design for snow disposal. *Computers & Operations Research*, 33(1):239–262.

[9] Perrier, N., Langevin, A., and Campbell, J. F. (2007a). A survey of models and algorithms for winter road maintenance. part III: Vehicle routing and depot location for spreading. *Computers & Operations Research*, 34(1):211–257.

[10] Perrier, N., Langevin, A., and Campbell, J. F. (2007b). 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*, 34(1):258–294.

[11] Salazar-Aguilar, M. A., Langevin, A., and Laporte, G. (2012). Synchronized arc routing for snow plowing operations. *Computers & Operations Research*, 39(7):1432–1440.