Iterated Local Search for a Vehicle Routing Problem with Synchronization Constraints

Nacima Labadie[1], Christian Prins[1] and Yanyan Yang[2]
[1]*LOSI, University of Technology of Troyes, CS42060, 10004, Troyes Cedex, France*
[2]*CGS, Mines ParisTech, 60 Boulevard Saint Michel, 75272, Paris Cedex 06, France*

Keywords: Vehicle Routing, Synchronization Constraints, Iterated Local Search.

Abstract: This paper deals with vehicle routing problem (VRP) with synchronization constraints. This problem consists in determining a least-cost set of routes to serve customers who may require several synchronized visits. The main contribution of the paper is: 1) it presents a definition and a classification of different types of synchronization constraints considered in the VRP literature; 2) it describes a variant of vehicle routing problem with synchronization constraints which is formulated as a mixed integer programming model; 3) finally, it provides a constructive heuristics and an iterated local search metaheuristic to solve the considered problem. The performance of the proposed approaches is evaluated small and medium sized instances.

1 INTRODUCTION

The vehicle routing problem (VRP) was introduced by Dantzig and Ramser in 1959 (Dantzig and Ramser, 1959). It is a fundamental planning problem in the field of transportation, distribution and logistics. This combinatorial optimization problem consists in seeking routes for a set of vehicles, to perform a set of tasks in a network. During the past half century there has been a vast amount of literature and research on the VRP and its variants, complete surveys can be found in Desaulniers and Desrosiers (Desaulniers et al., 2002), Berbeglia and Cordeau (Berbeglia et al., 2007), and Parragh et al. (Parragh et al., 2008).

A recently arising and hot extension of VRP is the vehicle routing problem with synchronization constraints (VRPS). Drexl (Drexl, 2011) defined the VRPS as a VRP exhibiting additional synchronization requirements in spatial, temporal, and load aspects, he gave a recent review of VRPS and defined the VRPS as a vehicle routing problem where more than one vehicle may or must be used to fulfil a task. However, there are some problems which require more than one vehicle to fulfil a task but are not synchronization problems, like split delivery VRP. Thus we propose a simpler definition of the VRPS: "*A VRPS is a vehicle routing problem where there exists at least one vertex requiring simultaneous visits of vehicles, or successive visits resulting from precedence constraints*".

The fundamental difference between the VRP and VRPS is that in the VRPS the routes are interdependent. It means that a change in one route may have effects on other routes because of the synchronization constraints while a change in one route does not affect any other route in the standard VRP. In the worst case, a change in a VRPS route may lead all the other routes infeasible.

Due to the spatial, temporal, and load aspects, Drexl (Drexl, 2011) classifies the VRPSs into five categories: Task synchronization, operation synchronization, Movement synchronization, Load synchronization and Resource synchronization. However, we propose a classification according to two synchronization types based on our definition, simultaneous synchronization and precedence synchronization. In the simultaneous synchronization (SS), the vertex requires at least two visits, simultaneously or in a narrow time window while in the precedence synchronization (PS) the vertex requires several successive visits with or without time windows.

The literature on routing problems with synchronization constraints is relatively scattered. For the case of simultaneous synchronization we can find the paper of Ioachim and J. Desosiers (Ioachim and Desosiers, 1999) which describes an aircraft fleet assignment and routing problem with synchronization constraints. In this problem, the authors propose the synchronization constraints from so-called same-departure-time requirements: the same identifier flights have to depart at the same time every day

Labadie N., Prins C. and Yang Y..
Iterated Local Search for a Vehicle Routing Problem with Synchronization Constraints.
DOI: 10.5220/0004837502570263
In *Proceedings of the 3rd International Conference on Operations Research and Enterprise Systems* (ICORES-2014), pages 257-263
ISBN: 978-989-758-017-8

during a week which induces a simultaneous synchronization problem. To solve the problem, the authors proposed a branch-and-price using a multi-commodity flow formulation. Common real applications of routing with synchronization are home health care and hospital systems. Bertels and Fahle (S. Bertels, 2006) considered a problem with hard rostering constraints like qualification requirements or work time limitations, and soft constraints like preference of serving time, preference of patients and preference of nurses. Some tasks may need cooperation of nurses which leads to simultaneous synchronization. The objective is to minimize the total cost and maximize patients/staff satisfaction. The authors develop a combined constraint programming and tabu search method which produces good solutions within short time. Bredstrom and Ronnqvist (Bredstrom and Ronnqvist, 2008) also consider home care staff scheduling where may be two or more staff members would be required to accomplish a task (such as two nurses for bathing elderly person). The objective function is a weighted sum of preferences, traveling time and balancing variables. In this work, a MIP formulation is proposed and a heuristic that iteratively solves restricted MIP problems is used to improve the best known feasible solutions. Inc (Braysy et al., 2009) three communal routing problems with synchronization are described: the organization of home care, transportation of the elderly and home meal delivery in Finland. In the home care nurse problem, personnel has a maximum number of working hours per day and different levels of education and different skills qualified to perform different services. Each customer in a given service area are allocated to a given team and workers typically work exclusively in their own team. Each customer requires several visits within a given interval time. The objective is to maximize the working hours and the workers' preferences while taking into account the regulation for breaks.

Amaya et al. (Amaya et al., 2007), (Amaya et al., 2010) study the capacitated arc routing problem with refill points (CARP-RP) of road marking problem in Canada. In this problem, there are two vehicles, a painting vehicle serving the road and a tank vehicle to refill the painting vehicle. The painting vehicle can only be refilled by the tank vehicle but at any road junction and both vehicles are originally located and end their routes at the depot. In the first paper, after each refill the tank vehicle must return to the depot which makes the synchronization constraint much simpler. The objective is to determine the vehicle routes simultaneously both for the painting vehicle and the tank vehicle to minimize the total traveling cost. To solve the problem, the authors propose an integer programming model and develop a cutting-plane algorithm applied to solve instances involving 20 to 70 nodes and 50 to 595 arcs.

The second publication extends the problem by since there is no need for the tank vehicle to return to the depot each time after the replenishment. The authors upgrade their previous IP model and develop a heuristic method based on the route-first-cluster-second principle. However, due to the construction of the network, the heuristic does not need to consider the synchronization constraints. Salazar-Aguilar et al. (Salazar-Aguilar et al., 2013) consider a similar road marking problem where several capacitated vehicles are used to paint lines on roads instead of only one in the two previous studies, and a tank vehicle is also used to replenish the painting vehicles. The objective is to determine routes and schedules for the painting and replenishment vehicles to minimize the makespan: the duration of the longest route. The authors developed an adaptive large neighbourhood search (ALNS) metaheuristic to solve this problem. In the constructive procedure, routes of painting vehicles have been generated, the potential refill nodes on each of them are identified and the route of the replenishment vehicle is then constructed by means of a GRASP ensuring the synchronization constraints. Moreover, the authors improve the ALNS by combining seven destroy/repair operators.

Applications with similar nature are developed in (Salazar-Aguilar et al., 2012). These authors proposed a synchronized arc routing for snow plowing operations in Canada. In this problem, streets require plowing operations in one or two direction and each direction has one to three lanes. All lanes belonging to the same segment in the same direction must be plowed simultaneously and each lane can only be served by one vehicle at a time. There is a fleet of snow plowing vehicles originally located at a depot. The objective is to determine a set of routes to minimize the duration of the longest route (makespan) while all street must be plowed.

Few studies deal with precedence synchronization: in Kim et al. (Kim et al., 2010), a combined manpower-vehicle routing problem with multi-staged services is studied. In this problem, the workers can only be moved to the customers by vehicles and customers demand several visits of different workers in a predefined sequence. The objective is to minimize the total cost of the vehicle routing. To solve the problem, the authors develop a simple constructive heuristic and a particle swarm optimization (PSO). A home health care staff scheduling problem is studied in (Rasmussen et al., 2012), the authors consider staffs with different qualifications where both patients and staffs have time windows, all visits are associated with priorities and each patient has his own preferred carer. A branch-and-price algorithm using Dantzig-Wolfe decomposition is used to solve the problem where the aim is to minimize a weighted sum of three criteria: the number of uncovered visits, the total traveling cost and the sum of preferences. A similar problem was studied in a previous work from Dohn et al. (Dohn et al., 2009). In this study, the authors propose a vehicle routing problem with time windows and precedence constraints. Each customer demands a sequence of visits in a given time window. The objective is to minimize the total travel distance. The authors developed a column generation approach to solve the problem.

The remainder of this paper is organized as follows. Section 2 describes the studied vehicle routing problem with simultaneous synchronization constraints, and proposes a mixed integer programming formulation of the problem. The solution approaches that we developed are described in Section 3. Computational results are presented in Section 4, followed by conclusions in Section 5.

2 PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

This work is dedicated a special case of vehicle routing problem with simultaneous synchronization constraints and is motivated by real applications occurring in home health care systems or services sector. In these fields, some customers may require a service which must be accomplished by more than one person because the personnel constituting a staff has not the same competencies.

Formally, the studied problem can be defined in an undirected graph $G = (V,E)$ with a node-set $V = \{0,1,2,,n,n+1\}$ and an edge-set E. Nodes 0 and $n+1$ are special nodes called initial resp. final depots; while the other vertices correspond to customers. Each edge $e = [i, j]$ is associated with a travel cost c_e and a traversal duration T_e. It is assumed that these times satisfy the triangle inequality. The initial depot 0 offers a set of possible services $P = \{1,2,,r\}$ to customers. Each customer demands a subset of services specified by a vector $U_i = (v_{i1}, v_{i2},, v_{ir})$ where v_{ip} equals to 1 if customer i demands the service $p \in P$ ($P = \{1,...,r\}$, 0 otherwise. Services required by the same customer should all start simultaneously and have a common time duration D_i. Each customer has its available time window $[\alpha_i, \beta_i]$ where α_i is the earliest start time of service at the customer i and β_i is the latest start time at the customer i.

A limited number V of vehicles in $H = H_1 \bigcup H_2 \bigcup ... \bigcup H_r$ offering services are originally located at the depot 0. Each vehicle $k \in H_p$ is qualified to perform a unique service p in the set P. $[\alpha_0, \beta_0]$ is the available time for all the vehicles which means that all vehicles can only leave the initial depot at α_0 and must return to the final depot before β_0.

The problem consists in building a set of routes starting at the initial depot 0 and ending at the final depot $n+1$, such that each vehicle route doesn't exceed the maximal imposed duration, each customer is provided the required services and each of these services must start at the same time (when there is at least two services needed) within the customer's time window. Each kind of service must be accomplished by the corresponding qualified vehicle (team). This study considers the minimization of overall cost of edges crossed by the vehicles.

The problem can be modeled as a linear mixed integer program using the following decision variables: binary variables x_{ijk} equal to 1 if and only if the vehicle k visits the node i then leaves to node j, real variables a_{ik}, t_{ik} and w_{ik} indicating respectively the arrival time of vehicle k at customer i, the starting time of service at customer i if visited by vehicle k, and the waiting time at customer i if visited by vehicle k.

The objective is to minimize the total cost (1). All customers' demands must be satisfied (2). Constraint (3) requires each vehicle to leave and return at the depot. (4) ensure the continuity of the routes. If a vehicle is set to travel between two customers, there has to be enough time between the two visits (5). All time windows must be respected (6). (7) imply the synchronization constraints. (8) define the service start time equal to the sum of the arrival time and the waiting time. The remaining constraints of the model fix the nature of the decision variables.

3 RESOLUTIONS APPROACH

To solve the problem, an iterative local search (*IlS*) framework is proposed. The general procedure of the implemented *ILS* (see Algorithm 1) starts by generating an initial solution (with a constructive heuristic H_1) which is then improved by a local search procedure (LS). The current best solution is considered as the starting one in the *ILS*. Each iteration of the algorithm considers the best current solution which is then perturbed to generated neighbor solutions later improved by Local Search (LS). If the new solution is better than the current best solution, this later is updated for the next steps. More details about the method can be found in (Loureno et al., 2003).

$$\min z = \sum_{i\in V}\sum_{j\in V}\sum_{k\in H} C_{ij}\cdot x_{ijk} \tag{1}$$

subject to

$$\sum_{k\in H_p}\sum_{j\in V\setminus\{0\}} x_{ijk} = v_{ip} \qquad \forall i\in V\setminus\{0,n+1\}, \forall p\in P \tag{2}$$

$$\sum_{j\in V\setminus\{0,n+1\}} x_{0jk} = \sum_{j\in V\setminus\{0,n+1\}} x_{jn+1k} = 1 \qquad \forall k\in H \tag{3}$$

$$\sum_{j\in V\setminus\{0\}} x_{ijk} = \sum_{j\in V\setminus\{n+1\}} x_{jik} \qquad \forall i\in V\setminus\{0,n+1\}, \forall k\in H \tag{4}$$

$$t_{ik} + (T_{ij}+D_i)\cdot x_{ijk} \le a_{jk} + \beta_i\cdot(1-x_{ijk}) \qquad \forall i,j\in V, \forall k\in H \tag{5}$$

$$\alpha_i\cdot\sum_{j\in V} x_{ijk} \le t_{ik} \le \beta_i\cdot\sum_{j\in V} x_{ijk} \qquad \forall i\in V\setminus\{0,n+1\}, \forall k\in H \tag{6}$$

$$v_{ip_1}\cdot v_{ip_2}\sum_{k\in H_{p_1}} t_{ik} = v_{ip_1}\cdot v_{ip_2}\sum_{k\in H_{p_2}} t_{ik} \qquad \forall i\in V, \forall p_1,p_2\in P, p_1\neq p_2 \tag{7}$$

$$t_{ik} = a_{ik} + w_{ik} \qquad \forall i\in V, \forall k\in H \tag{8}$$

$$x_{ijk}\in\{0,1\} \qquad \forall i,j\in V, \forall k\in H$$

$$a_{ik}, t_{ik}, w_{ik}\in\mathbb{R}^+ \qquad \forall i\in V, \forall k\in H \tag{9}$$

Algorithm 1: *General structure of the* IlS.

GenerateInitialSolution (S_0)
$S^* := LS(S_0)$
$BestSol := S^*$
for $k := 1$ **to** k_{max} **do**
  $SChild := Perturb(S^*)$;
  $NewSol := LS(SChild)$
  if $z(NewSol) < z(BestSol)$ **then**
    $BestSol := NewSol$
  end if
end for

3.1 Constructive Heuristic

A constructive heuristic called *Simple-Append* is proposed to generate an initial solution. In this algorithm, a permutation of all customers is transformed to a feasible schedule. Here the initial permutation is a list of the customers sorted by their earliest completion time $\alpha_i + D_i, i\in V\setminus\{0,n+1\}$ in a non-descending order. at each iteration j of the algorithm, the customer in the position j from the permutation is selected, then the set of vehicles which can service the corresponding customer before the end of its time window is computed. If there are more than one possible vehicle, the vehicle that have the least travel cost is selected. If there are insufficient vehicles to service the customer, the solution is unfeasible. Services at each customer are scheduled to start when all required vehicles have arrived. If this occurs earlier than the earliest starting time of service, a waiting time must be considered for each needed vehicle and starting time of service is set to the beginning of customer time window.

3.2 Perturbation Procedure

The perturbation procedure has a role of the diversification in the *ILS*. To move from one feasible solution to neighboring solutions, neighbors of the current permutation are found by using a *Relocate* operator. First, all the routes are concatenated then randomly chosen customers are selected and their position position is changed randomly in the permutation. The neighbor detailed solutions are then constructed by the *Simple-Append* heuristic.

3.3 Local Search

The Local Search procedure used in the *ILS* contains two main procedures: the first one is based on 2-opt move executed on a single route. The second one is the *k-exchange* moves including three types of movements that involve two routes. In local search procedure, the service start time at the customers requiring synchronization visits are fixed like in the input solution. The *2-opt* procedure removes and replaces two arcs in a tour and reorders vertices. The *k-exchange* moves offer the possibility of improving the assignment decisions of customers to routes. Three basic *k-exchange* neighborhoods for the VRP (see (Savelsbergh, 1992)) relocating vertices between two routes are tested, the first one is the relocate operator that removes one customer from a route and insert it in a different tour, the second one is the *exchange* operator that swaps the positions of two customers from two different routes. The last one is the *2-opt** operator applied to different routes. The neighborhoods exploration requires $O(n^2)$ by reducing the computational effort of time windows violation to $O(1)$ thanks to the feasibility checks of (Savelsbergh, 1985).

4 NUMERICAL RESULTS

A set of test instances has been extended from the data set proposed by Bredstrom and Ronnqvist (Bredstrom and Ronnqvist, 2008). In these new instances, the customers' demands are randomly generated but the information including the distance matrix, time windows, and the time duration of routes remains the same as in (Bredstrom and Ronnqvist, 2008). Table 1 describes the used instances, column 2 gives the number of customers $|N|$, the total number of vehicles $|K|$ and the number of synchronized visits $|P^{syn}|$ are provided in the two last columns.

Table 1: The used benchmark.

Instances	$\|N\|$	$\|K\|$	$\|P^{syn}\|$
1	8	3	1
2	8	3	2
3	8	4	2
4	12	5	1
5	12	5	2
6	12	5	3
7	18	5	2
8	18	5	4
9	45	14	8
10	45	16	5

The mathematical models have been implemented in the software GUSEK, and the heuristic and meta-heuristic algorithms haven been coded in C^{++} language. The results are shown in Tables 2, 3, 4. All experiments have been carried out on a 3.00 GHz, Intel(R) Core(TM) 2Duo processor PC, running under Windows XP with 1.96 GB of RAM.

Note that in the *Simple-Append* heuristic, a list of customers sorted by earliest finish time is used to get the initial solution in the *ILS*. *LS-2OPT* indicates the local search with *2-opt* procedure, *LS-REL*, *LS-EXCH*, *LS-2OPT** stand for the local search moves when two routes are involved and respectively the *Relocate*, *Exchange*, *2-opt** operator is used. The maximum number of iterations K_{max} is set to 1000. Column *Imp.* in Table 4 computes the improvement achieved by the *ILS* compared to the initial input solution obtained with *Simple-Append*.

Table 2: Computational results-Part I.

	GUSEC		*Simple-Append*	
File	Obj	CPU(s)	Obj	CPU(s)
1	217	6.2	222	0.031
2	224	24.4	224	0.047
3	207	1.7	208	0.031
4	390	5427	402	0.031
5			446	0.047
6			462	0.031
7			340	0.031
8			388	0.047
9			856	0.047
10			959	0.031
Av.				0.0374

Table 3: Computational results-Part II.

	LS-2OPT		*LS-REL*		LS-EXCH	
File	Obj	CPU(s)	Obj	CPU(s)	Obj	CPU(s)
1	222	0.047	222	0.047	222	0.047
2	224	0.031	224	0.031	224	0.047
3	207	0.047	207	0.047	207	0.047
4	402	0.063	393	0.047	393	0.047
5	446	0.063	446	0.047	442	0.047
6	462	0.047	462	0.047	458	0.047
7	340	0.047	333	0.047	334	0.047
8	387	0.063	377	0.047	387	0.047
9	823	0.047	778	0.047	781	0.047
10	941	0.094	929	0.094	916	0.094
Av.		0,0549		0,0501		0,0517

Table 2 reports that GUSEK only performs well with small instances and cannot solve problems from 12 customers and 2 synchronization visits. Furthermore, we can see that *Simple-Append* with a list of customers sorted by earliest finish time used as the initial permutation has a really good performance with a convergence error less than 5% and a much smaller computation time.

Table 3 shows that the *2-opt* local search move involving a single route has not big improvement on the results compared to the initial input solution obtained by *Simple-Append*. It is due to the fact that in the local search method, we fix to the same value the service start time of the synchronization visits as in the input solution. In this case, the improvement of assignment decisions is much more effective than the improvement of routing decisions as shown in the table. Moreover, the same operator involving two different routes turns out to be the most powerful movement among the three moves involving more than one route.

Finally, Table 4 shows that *ILS* can reach the optimum on small instances and has a much smaller computation time. For the larger instances, *ILS* has reachs a maximum improvement of 16% when compared to the initial solutions.

Table 4: Computational results-Part III.

	*LS-2OPT**		*ILS*		
File	Obj	CPU(s)	Obj	CPU(s)	Imp.
1	222	0.031	217	0.106	0.0225
2	224	0.047	224	0.104	0
3	207	0.047	207	0.108	0.00480
4	393	0.047	390	0.125	0.0298
5	442	0.047	436	0.138	0.0224
6	458	0.047	452	0.135	0.0216
7	333	0.063	333	0.139	0.0205
8	387	0.047	361	0.163	0.069
9	741	0.047	718	0.203	0.161
10	884	0.047	844	0.25	0.119
Av.		0.047		0.1471	0.047255491

5 CONCLUSIONS

This paper provides a quick review of routing problem with synchronization constraints studied in the literature. It considers a particular case of vehicle routing problem with simultaneous synchronization constraints where a service centre (the depot) offers several types of services and customers may demand more than one service to be provided simultaneously. To solve this NP-hard problem, a mixed integer programming model minimizing the total travel cost has been proposed. Furthermore, a constructive heuristics and a metaheuristic based on local search have been developed. Computational experiments are carried out on 10 instances which are extended from the benchmark initially proposed by (Bredstrom and Ronnqvist, 2008). The experimental results obtained by solving the MIP model with GUSEK solver show that only instances with less than 12 customers and 2 synchronization visits can be solved optimally.

The constructive heuristic has shown quick solutions with all the testing instances with up to 45 customers and 15 synchronization visits. To improve the results, four local search operators have been tested and embedded within an Iterative Local Search (*ILS*) framework. The *ILS* has reached the optimum for the small instances solved by GUSEK very quickly. For the other instances, the *ILS* achieves a maximum improvement of 16% when compared to the initial input solutions. Future work would be dedicated to improving the local search especially by designing moves more suitable for handling the synchronization constraints. The problem studied in this paper uses the minimization of the total travel cost as objective function, another interesting direction for future research could consider other criteria like the overall waiting, work balancing.

REFERENCES

Amaya, A., Langevin, A., and Trpanier, M. (2007). The capacitated arc routing problem with refill points. *Operations Research Letters*, 35:45 –53.

Amaya, A., Langevin, A., and Trpanier, M. (2010). A heuristic method for the capacitated arc routing problem with refill points and multiple loads. *Journal of the Operational Research Society*, 61:1095–1103.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *TOP*, 15:1–31.

Braysy, O., Dullaert, W., and Nakari, P. (2009). The potential of optimization in communal routing problems: case studies from finland. *Journal of Transport Geography*, 17:484 – 490.

Bredstrom, D. and Ronnqvist, M. (2008). Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research*, 191:19–31.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 1:80–91.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M., and Soumis, F. (2002). Vrp with pickup and delivery. In Toth, P. and Vigo, D., editors, *The vehicle routing problem*, volume 9, pages 225–242. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA.

Dohn, A., Rasmussen, M. S., and Larsen, J. (2009). The vehicle routing problem with time windows and temporal dependencies. volume 1. DTU Management Engineering, Copenhagen.

Drexl, M. (2011). Synchronization in vehicle routing: A survey of vrps with multiple synchronization constraints. *Technical Report LM*.

Ioachim, I. and Desosiers, J. (1999). Fleet assignment and routing with schedule synchronization constraints. *European Journal of Operational Research*, 119:75 – 90.

Kim, B., Koo, J., and Park, J. (2010). The combined manpower-vehicle routing problem for multi-staged services. *Expert Systems with Applications*, 37:8424–8431.

Loureno, H., Martin, O., and Thomas, T. (2003). Iterated local search. In Hillier, F. S., Glover, F., and Kochenberger, G., editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 320–353. Springer New York.

Parragh, S., Doerner, K., and Hartl, R. (2008). A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, 58:21–51.

Rasmussen, M. S., Justesen, T., Dohn, A., and Larsen, J. (2012). The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219:598–610.

S. Bertels, T. F. (2006). A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. *Computers and Operations Research*, 33:2866 – 2890.

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

Salazar-Aguilar, M. A., Langevin, A., and Laporte, G. (2013). The synchronized arc and node routing problem: Application to road marking. *Computers and Operations Research*, 40:1708 – 1715.

Savelsbergh, M. W. P. (1985). Local search in routing problems with time windows. *Annals of Operations Research*, 4:285–305.

Savelsbergh, M. W. P. (1992). The vehicle routing problems with time windows: Minimizing route duration. *INFORMS Journal on Computing*, 4:146–154.