This is a pre print version of the following article:

Optimization Methods for the Same-Day Delivery Problem / Côté, J., De Queiroz, T.A., Gallesi, F., Iori, M. (AIRO SPRINGER SERIES). - In: Advances in Optimization and Decision Science for Society, Services and Enterprises / [a cura di] Paolucci M., Sciomachen A., Uberti P.. - [s.l] : Springer, Cham, 2020. - ISBN 978-3-030-34959-2. - pp. 335-349 [10.1007/978-3-030-34960-8_30]

Springer, Cham
Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

13/10/2026 19:07

(Article begins on next page)

Optimization Methods for the Same-Day Delivery Problem

Jean-François Côté, Thiago Alves de Queiroz, Francesco Gallesi, and Manuel Iori

Abstract In the same-day delivery problem, requests with restricted time windows arrive during a given time horizon and it is necessary to decide which requests to serve and how to plan routes accordingly. We solve the problem with a dynamic stochastic method that invokes a generalized route generation function combined with an adaptive large neighborhood search heuristic. The heuristic is composed of destroying and repairing operators, and the generalized route generation function, taking advantage of sampled-scenarios that are solved with the heuristic, determines which decisions should be taken at any instant. Results on different instances have shown the effectiveness of the proposed method in comparison with a consensus function from the literature, with an average decrease of 10.7%, in terms of solution cost, and 24.5%, in terms of runtime.

Key words: Same-day delivery problem, pickup and delivery problem, dynamic stochastic, adaptive large neighborhood search, route generation function.

Invited session: SP-OU-1

1 Introduction

Same-day delivery is appearing constantly in online retail. Other similar, related applications of this problem emerge in the delivery of groceries and transportation of patients between their homes and a hospital. This problem claims attention, especially due to the complicated, expensive logistic decisions that arise, since it is related to the classical NP-hard vehicle routing problem. Assuming a horizon of time

Thiago Alves de Queiroz, Francesco Gallesi, and Manuel Iori
Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Reggio Emilia, Italy, e-mail: taq@ufg.br, gallesi.francesco@gmail.com, manuel.iori@unimore.it.

Jean-François Côté
Department of Operations and Decision Systems, Laval University, Quebec City, Canada, e-mail: jean-francois.cote@fsa.ulaval.ca.

over which a fleet of (identical) vehicles should operate, it is necessary to determine routes for these vehicles, aiming at maximizing the number of requests that can be delivered on time and minimizing traveled distance. As requests are arriving dynamically during the time horizon, each one associated with a time window that needs to be respected, vehicles may return to the depot after performing a delivery in order to pickup products and continue serving the following requests [10].

The Same-Day Delivery Problem (SDDP) is a dynamic problem that was introduced in [10]. The authors assumed that vehicles could return to the depot in order to pickup only after finishing their current routes. Based on sampled-scenarios, the authors used a consensus function as a way to take dynamic decisions and consequently generate the vehicle routes. They also considered that a vehicle can wait at the depot while new requests arrive (i.e., they studied a waiting strategy). The idea behind the latter was to anticipate decisions based on the already known requests and potential future ones that were sampled from known probability distributions. From the experimental results, the authors concluded that considering the uncertainty of future requests had an important impact on the solution quality.

The SDDP can be viewed as the dynamic version of a one-to-one Pickup and Delivery Problem (PDP) with time windows (TW), with a single pickup location, which is the depot, to where vehicles need to return for new pickups during the time horizon. Comprehensive surveys in the PDPTW can be found in [2], related to the transportation of goods, and in [6], related to the transportation of people. Moreover, some related works are: [3], in the delivery of groceries, where time windows must be strictly respected and requests are generally known one-day in advance; [1], aiming at maximizing the total expected profit and vehicles can depart from the depot as soon as requests are available (i.e., there is no waiting strategy); [4], in which release dates are associated with requests and a genetic algorithm with local searches is used to solve the problem; [8], solving the PDPTW in which the pickup and delivery nodes are known in advance but not the time at which requests are available (i.e., requests are arriving dynamically during the time horizon); [9], where a multi-period problem is solved in which requests are dynamically integrated into existing decisions and some requests can be served on the next day.

In this work, we tackle the SDDP but, differently from [10], the objective is to minimize the total cost that is incurred from performing all routes and rejecting requests. Moreover, when a vehicle starts performing its route, we allow it to return to the depot after serving a customer and before completing its route in order to pickup more requests. The latter assumption generalizes the proposal in [10], which allows vehicles to return to the depot only after finishing their routes, and enlarges substantially the space of decisions. This is expected to come at the expenses of a larger computational effort. Considering these assumptions, we propose a generalized route generation function in order to improve the way in which routes are built in [10] and develop an Adaptive Large Neighborhood Search (ALNS) [7] to iteratively solve sub-instances on the sampled-scenarios.

Therefore, we present results for three versions of the SDDP: *static*, in which the problem is solved with the ALNS for all the time horizon, assuming that all requests are known in advance; *dynamic*, in which the ALNS is applied several times during

the time horizon in order to update the current solution; and *dynamic-stochastic*, in which the generalized route generation function with the ALNS, considering sampled-scenarios of future information, is used to update the current solution. Results are compared with those of the consensus function in [10], specially in the dynamic-stochastic version, considering 120 instances that such authors have proposed.

This work is organized as follows: Section 2 has a formal description of the problem, with its objectives and constraints; Section 3 describes the ALNS and the generalized route generation function, highlighting the differences over that proposed in [10]; Section 4 contains the experimental results of the three versions of the problem and also considering the consensus function in [10]; finally, Section 5 brings concluding remarks and directions for future works.

2 Problem Definition

The SPPD under study considers a fleet M of identical vehicles and a set L of customers locations over a geographical area. A central depot, denoted as node 0, is associated with start and end times between which vehicles can depart and arrive (i.e., the depot working hours, or the time horizon over which the depot and vehicles are in operation). With each pair $i, j \in L$, it is associated a deterministic travel time t_{ij} and a cost c_{ij} (e.g., distance) that are known in advance. During the depot working hours, requests arrive at a rate $\lambda_i \geq 0$ from each location $i \in L$. Let $\mathcal{R}$ be the set of requests that will occur during the time horizon. It is composed of requests that are known in advance and some others that unkown requests at the beginning but will become known as time unfold. Each request $k \in \mathcal{R}$ has a service time μ_k, a demand d_k, and a delivery time window $[s_k, e_k]$. Request k becomes only known at release time r_k and can only be treated later on. Requests that are found impossible to deliver on time can be assigned to third-party logistic operator at the expenses of the cost. It is assumed that the delivery costs incurred by the fleet are always lower than the cost of the third-party logistic.

Vehicles start and end at the depot according to its working hours and may serve one or more requests according to the requests that are currently available, respecting vehicles capacity Q. The design of the route associated with each vehicle may involve waiting at the depot for new requests or picking up some requests to perform the deliveries. Also, no diversion is allowed when a vehicle on the way to a customer. However, as soon as a delivery is done, the vehicle can return to the depot to pickup new requests. This means the vehicle doesn't need to finish serving all its on board requests before going back to the depot. The objective of the SDDP is to plan routes for vehicles, aiming at first maximizing the number of requests served by the fleet and secondly, minimizing the total cost of performing the routes.

The description above corresponds to the dynamic version of the problem for which we consider an ALNS to solve partial instances of the problem at any given time of the time horizon. The ALNS is also used to solve the static version for which

all requests are known at the start time of the day and then this resulting solution serves as an estimation for the other versions. Aiming at improving solutions of the dynamic version, we consider the dynamic-stochastic version in which sampled-scenarios are used to help with decisions regarding possible future requests.

3 Solution Methods

This section first describes how the SDDP is modeled, then presents the different events that can occur in real-time. Therefore, it describes the ALNS and the two different approaches for tackling the problem: dynamic version and the dynamic-stochastic version.

3.1 Modeling

The problem is modeled as a classical pickup and delivery problem with time windows with the inclusion of release dates for the arrival of new requests. At any instant, the set of known requests is built where each request is composed of a pickup node at the depot and a drop node at the customer location, besides a restricted time window. Modification of any element that was performed is forbidden, so only choices concerning new requests or nodes that were not visited can be changed. Scenarios containing future requests are generated to help on minimizing costs. Futures requests are dealt like regular requests with the exception that a vehicle cannot take any action before the release date (i.e., the vehicle has to stay idle until the release of the request).

It is important to note that our method allows all types of sequences of nodes to happen. This is not the case in [10] where the problem is modeled as a team orienteering problem with time windows and multi-trips [5]. In this problem, every request is composed of a single delivery node. Future requests are also generated and vehicles must return to the depot to do the pickup every time one of their nodes are encountered in the routes. The drawback is that only a subset of possible routes can be produced. For example, it is impossible, in a single node per request model, a route where a vehicle is waiting at the depot for future requests, then it goes to deliver real requests and finally future requests.

3.2 Event Management

In [10] is defined two types of events: (1) arrival of a new request when there is at least one vehicle that is waiting at the depot; and, (2) a vehicle has just arrived at the depot or completed its waiting period. Every time a new event happens, instances of

the PDPTW are generated and are solved using the ALNS. When allowing vehicles to not complete their routes, we need to consider a delivery completion as new event. Namely, when a vehicle completed a delivery, it can be diverted to the depot to pickup requests and perform the deliveries later. Finally, it is worth nothing that this additional event will possibly increase computational time.

3.3 Adaptive large neighborhood search

The proposed ALNS is based on [7], which uses the acceptance probability function of the simulated annealing to accept worse solutions. Then, it works as follows, given an input instance of the problem: (i) it obtains a feasible solution x by a constructive heuristic; (ii) it applies a destroy operator on x to obtain x'; (iii) it applies a repair operator on x' to obtain x''; (iv) it replaces x with x'' if x'' has lower cost or else by applying the acceptance probability function; (v) it goes back to step (ii) if the maximum number of iterations is not reached, or otherwise it returns x.

In step (i), the initial solution is constructed by observing the release date of requests in a greedy way. With relation to the destroy operator, we consider the *removal* and *random* operators that disregard requests of the solution. In the first one, requests that are closely related (i.e., in terms of cost, time, and capacity) are removed. In the second one, requests are randomly selected and removed. Thus, the removed requests are reinserted in step (iii) by one of two repair operators. The first one is a greedy operator that reinserts each of the removed requests into the best route overall. The other one is based on a regret operator, which is a generalization of the greedy one in the sense that not only the best but also k routes are analyzed since a given request cannot be reinserted into the best route.

In steps (ii) and (iii), an operator is chosen according to the roulette wheel selection principle in which a given weight is associated with each operator. These weights are dynamically updated by using statics of previous iterations in which a reaction factor is used to control the influence of weights. Moreover, at the end of step (iii), a local search is applied in x'', consisting of determining the best moment to serve each request that has not been served yet. Regarding the acceptance probability function, a given initial temperature is decreased over the ALNS iterations and thus the probability of accepting worse solutions in comparison with the current one is decreased as well.

3.4 Dynamic problem

In this version, a PDPTW instance and its solution are maintained over time. On each new event, the instance is updated with new information (e.g., delivery completion, new requests, etc.) and elements that were performed in the past are fixed inside their routes. The ALNS is run to obtain a new solution and it updates the maintained so-

lution. New pickup and departure commands to the vehicles are generated. Requests that remained outside the solution are given to the third-party logistic operator when they become impossible to serve.

3.5 Dynamic-stochastic problem

In order to improve routes that are planned in the dynamic version for any event, sample-scenarios of future requests are used. These scenarios are generated from a probability function taking into consideration the known requests until the current time. Hence, each sampled-scenario is solved with the ALNS similarly to what is performed in the dynamic problem but now also considering future requests that contemplate a time horizon.

After solving all scenarios, a generalized route generation function is used to identify the best solution among all them. Then, such best solution is used to update the current solution. This function works on the following way: (i) for each solution of a each scenario, remove the sampled requests and every real requests that lie after at least one sampled request from all routes, since they indicate that a vehicle must wait or return to the depot to pickup some future requests; (ii) assign a score to each solution based on the number of times each of its routes are in other solutions, where the solution with the highest score is chosen and implemented. As commented before, requests outside the solution are assigned to the third-party logistic operator when they become impossible to serve.

4 Experimental Results

All the methods were coded in the C++ programming language and ran on an Intel 2.667 GHz Westmere EP X5650 processor. The experiments were carried out over a subset of instances from [10]. The instances under consideration are of two types with relation to the customer location geographies, namely, clustered (C) and randomly dispersed (R). For each geography, we consider data sets that contains 100 (C_1 and R_1) and 200 (C_2, C_6, R_2, and R_6,) customers, as well as five types of time windows that are TW.d1, TW.f, TW.h, and TW.r, with one-hour deadlines, and TW.d2, with two-hours deadline. Moreover, the requests arrival rate is homogeneous and there are four different rates that are 1, 2, 3, and 4 (i.e, the overall arrival rate is of 0.1 requests per minute and so on). Therefore, we have a total of 120 instances in such a way the first instance is named as TW.d1_C_1_hom_1 (and so on). The number of vehicles is fixed to 10 for any instance.

Regarding the parameters of the methods, we carried out preliminary experiments in which the sampling horizon was defined over the entire horizon, and the ALNS had 50 and 250 iterations, assuming 30 scenario samples. These experiments indicated, in terms of solution quality and runtime, that performing 250 iterations for the

ALNS are preferable. Thus, such values were adopted when solving all the 120 instances. The results that we obtained are presented in Tables 1 and 2. Each line of these tables has the name of the instance, the solution of the static, dynamic, and dynamic-stochastic versions as explained in Section 3, as well as the solution of the dynamic-stochastic but by using the consensus function in [10] with the ALNS. For each problem, it is presented the total solution cost, number of not served requests, and total computational time in seconds.

Observing Table 1, the average solution cost and runtime (in seconds) are, respectively: 2203.5 and 25.4, for the static problem; 3319.6 and 26.0, for the dynamic problem; and, 2592.7 and 14806.1, for the dynamic-stochastic problem that was solved with the generalized route generation function. We notice that the dynamic-stochastic that was solved with the consensus function in [10], where these values are 2913.2 and 19430.6, respectively, is outperformed by the proposed method, where there is a decrease of 11.0% and 23.8%, respectively. In terms of the number of not served requests, the proposed method performed the best with 0.3 more requests on average over [10].

The results of Table 2 are very similar to those of Table 1. In summary, from Table 2, the average solution cost, number of not served requests, and runtime (in seconds) are: 2321.4, 4.8, and 27.1, for the static problem; 3253.0, 14.4, and 27.3, for the dynamic problem; 2633.1, 8.7, and 14622.6, for the dynamic-stochastic problem that was solved with the generalized route generation function; and, 2937.2, 8.9, and 19552.5, the dynamic-stochastic that was solved with the consensus function in [10]. Once again, the proposed method is able to overcome the dynamic (i.e., in terms of solution cost and number of not served requests, there is a decrease of 19.1% and 35.7%, respectively) and dynamic-stochastic of the literature (i.e., in terms of solution cost and runtime, there is a decrease of 10.3% and 25.2%, respectively), and better approximate the results of the static problem (i.e., in terms of solution cost and number of not served requests, they have the smallest percentage deviation).

Finally, with relation to the instances characteristics, comparing the dynamic-stochastic problem with the respective version that was solved with the consensus function in [10], from Tables 1 and 2, we can highlighted that the latter performed worse in all geographies (R and C), time windows (TW.d1, TW.d2, TW.f, Tw.h, and TW.h), and requests arrival rates (1, 2, 3, and 4) in terms of average solution cost and runtime. Thus, we can conclude that the generalized route generation function, which allows vehicles to stop their current routes and return to depot to pickup requests, performs well in practice.

5 Concluding Remarks

The same-day delivery problem is tackled for which a generalized route generation function combined with an adaptive large neighborhood search is proposed, where sampled-scenarios are used to anticipate future requests and improve decisions. The ALNS has destroy and repair operators whose respective weights are dynamically

Table 1 Results of the C instances.

	Static			Dynamic			Dyn-Stoc			Dyn-Stoc in [10]		
Instance	Cost	#Not	Time(s)	Cost	#Not	Time(s)	Cost	#Not	Time(s)	Cost	#Not	Time(s)
TW.d1_C_1_hom_1	2264.0	1	2	2976.0	1	2	2878.4	1	704	2980.8	1	1667
TW.d1_C_1_hom_2	3275.2	2	5	4281.6	6	3	3980.8	4	2393	4192.0	6	3198
TW.d1_C_1_hom_3	4211.2	9	15	4913.6	22	6	4684.8	20	4578	4772.8	20	4016
TW.d1_C_1_hom_4	4270.4	13	21	4878.4	30	9	4601.6	26	9815	4620.8	25	9895
TW.d2_C_1_hom_1	1329.6	1	3	2633.6	1	2	1846.4	1	1382	1961.6	1	1938
TW.d2_C_1_hom_2	1820.8	1	12	3820.8	1	5	2185.6	1	5754	2609.6	1	8430
TW.d2_C_1_hom_3	2456.0	5	38	4497.6	10	11	2734.4	8	17992	3305.6	6	22907
TW.d2_C_1_hom_4	2515.2	4	61	4942.4	9	14	3009.6	4	35838	3587.2	5	54882
TW.f_C_1_hom_1	1398.4	1	3	2851.2	1	2	1891.2	2	1221	2312.0	2	2678
TW.f_C_1_hom_2	1886.4	1	10	3558.4	1	6	2257.6	1	4895	3113.6	1	6416
TW.f_C_1_hom_3	2531.2	5	30	3993.6	7	14	2937.6	8	14820	3776.0	8	13258
TW.f_C_1_hom_4	2609.6	4	47	4323.2	8	16	2910.4	5	27199	3601.6	7	25843
TW.h_C_1_hom_1	1134.4	1	3	1940.8	5	4	1512.0	2	1863	1676.8	3	1933
TW.h_C_1_hom_2	1624.0	1	12	2160.0	10	16	1910.4	1	7055	2275.2	2	7364
TW.h_C_1_hom_3	2163.2	5	55	2561.6	28	72	2636.8	14	28556	2622.4	13	25866
TW.h_C_1_hom_4	2246.4	4	71	2620.8	33	126	2547.2	11	46525	2611.2	17	56207
TW.r_C_1_hom_1	1168.0	1	4	1969.6	6	4	1544.0	3	1712	1801.6	3	1814
TW.r_C_1_hom_2	1401.6	1	12	2297.6	12	16	1649.6	1	6703	1972.8	4	7064
TW.r_C_1_hom_3	2438.4	5	39	2684.8	31	50	2683.2	13	21983	2859.2	12	31046
TW.r_C_1_hom_4	2401.6	5	68	2590.4	39	97	2681.6	11	43907	2987.2	13	46370
TW.d1_C_2_hom_1	2696.7	2	1	3253.0	2	2	3302.3	2	608	3265.1	2	1553
TW.d1_C_2_hom_2	3210.2	3	3	4122.8	4	2	3690.2	4	1972	3917.2	3	3431
TW.d1_C_2_hom_3	4254.0	17	13	5032.6	30	6	4812.1	33	3714	4801.9	30	2933
TW.d1_C_2_hom_4	4174.4	26	17	4982.3	38	7	4447.3	40	8061	4568.2	39	9112
TW.d2_C_2_hom_1	1578.6	2	2	3041.9	2	2	1827.9	2	1122	1947.0	2	2129
TW.d2_C_2_hom_2	1591.6	3	7	3542.3	3	4	1813.0	3	4661	2323.7	3	7272
TW.d2_C_2_hom_3	2251.2	7	36	4815.8	8	11	2527.4	7	17309	3330.2	7	26622
TW.d2_C_2_hom_4	2574.0	5	54	4832.1	7	16	2922.1	5	35374	3570.7	6	49661
TW.f_C_2_hom_1	1582.3	2	2	3046.5	2	2	1839.1	2	1019	2480.0	2	1868
TW.f_C_2_hom_2	1607.4	3	6	3321.9	3	4	2052.1	3	4108	2454.9	3	7197
TW.f_C_2_hom_3	2280.0	7	29	4353.5	12	14	2686.5	7	14388	3739.5	9	13423
TW.f_C_2_hom_4	2674.8	5	43	4281.5	11	23	3182.3	6	27112	3732.8	7	29382
TW.h_C_2_hom_1	1375.8	2	3	2266.0	6	3	1586.0	3	1442	1922.8	5	2998
TW.h_C_2_hom_2	1162.8	3	7	2016.7	9	11	1501.4	3	5933	1778.6	5	5184
TW.h_C_2_hom_3	2044.7	7	39	2881.9	17	68	2439.1	11	22743	2668.8	13	39875
TW.h_C_2_hom_4	2418.3	6	63	2635.4	33	95	2821.4	15	43067	2953.3	11	53510
TW.r_C_2_hom_1	1312.6	2	3	2094.9	6	3	1734.0	2	1429	1848.4	2	2253
TW.r_C_2_hom_2	1385.1	3	7	2160.0	15	8	1709.8	4	5674	2037.2	3	9349
TW.r_C_2_hom_3	2011.2	7	39	2888.4	22	56	2887.4	7	22960	2935.8	8	37789
TW.r_C_2_hom_4	2337.7	5	61	2372.5	41	112	2795.7	6	41491	2876.3	8	69096
TW.d1_C_6_hom_1	1054.7	0	3	1666.9	1	4	1332.2	0	1779	1468.1	0	3870
TW.d1_C_6_hom_2	1713.8	2	18	2374.7	12	29	2047.5	4	10909	2139.4	5	11673
TW.d1_C_6_hom_3	1721.3	3	37	2556.6	16	77	2164.7	5	22547	2408.4	3	33890
TW.d1_C_6_hom_4	2223.8	6	83	2705.6	32	131	2796.6	10	52112	2686.9	11	71392
TW.d2_C_6_hom_1	1845.9	0	2	2377.5	0	2	2142.2	0	785	2353.1	0	2589
TW.d2_C_6_hom_2	3384.4	3	6	4400.6	6	4	4322.8	6	2878	4215.0	6	3847
TW.d2_C_6_hom_3	3802.5	3	13	4769.1	15	6	4283.4	12	6093	4488.8	14	6891
TW.d2_C_6_hom_4	4254.4	13	27	5012.8	28	9	4596.6	25	11753	4710.9	21	8514
TW.f_C_6_hom_1	1247.8	0	3	2356.9	0	2	1393.1	0	1462	1589.1	0	3069
TW.f_C_6_hom_2	1928.4	2	13	4281.6	4	5	2294.1	2	7143	2794.7	2	13094
TW.f_C_6_hom_3	2079.4	3	32	4267.5	7	12	2452.5	4	18950	3011.3	4	33698
TW.f_C_6_hom_4	2462.8	6	75	4988.4	10	17	2891.3	8	42447	3343.1	8	62566
TW.h_C_6_hom_1	1286.3	0	3	2175.0	0	2	1393.1	0	1270	1815.0	0	3592
TW.h_C_6_hom_2	1981.9	2	11	3826.9	5	7	2441.3	3	5948	3285.9	2	8083
TW.h_C_6_hom_3	2167.5	3	26	3829.7	7	15	2541.6	4	15367	3310.3	4	23639
TW.h_C_6_hom_4	2595.0	6	60	4156.9	9	30	2879.1	8	33223	3699.4	8	36124
TW.r_C_6_hom_1	1080.0	0	3	1455.9	4	3	1319.1	0	1753	1459.7	0	3689
TW.r_C_6_hom_2	1760.6	2	14	2391.6	19	16	2061.6	4	8192	2261.3	5	17287
TW.r_C_6_hom_3	1772.8	3	38	2472.2	21	56	2046.6	7	23503	2354.1	8	41579
TW.r_C_6_hom_4	2180.6	6	108	2672.8	24	207	2502.2	13	71170	2602.5	19	69321
Average	2203.5	4.2	25.4	3319.6	12.5	26.0	2592.7	7.0	14806.1	2913.2	7.3	19430.6

Table 2 Results of the R instances.

	Static			Dynamic			Dyn-Stoc			Dyn-Stoc in [10]		
Instance	Cost	#Not	Time(s)	Cost	#Not	Time(s)	Cost	#Not	Time(s)	Cost	#Not	Time(s)
TW.d1_R_1_hom_1	2078.7	1	2	2798.7	1	2	2562.6	1	706	2674.8	1	2222
TW.d1_R_1_hom_2	1756.4	0	1	2379.5	0	2	2231.9	0	696	2461.0	0	2399
TW.d1_R_1_hom_3	4300.6	12	14	4999.4	24	6	4614.2	27	4717	4800.0	23	4338
TW.d1_R_1_hom_4	4453.9	8	24	5047.0	22	9	4558.3	24	11475	4714.8	22	11696
TW.d2_R_1_hom_1	1215.5	1	4	2471.6	1	3	1538.7	1	1374	1740.0	1	3177
TW.d2_R_1_hom_2	1117.9	0	3	2371.1	0	2	1368.1	0	1391	1592.1	0	3790
TW.d2_R_1_hom_3	2640.0	4	38	4654.8	8	14	2926.5	6	18259	3400.6	7	26058
TW.d2_R_1_hom_4	2598.3	1	67	4773.9	6	19	2970.4	5	40201	3407.0	1	61951
TW.f_R_1_hom_1	1267.7	1	3	2429.0	1	2	1577.4	1	1217	2034.2	1	2902
TW.f_R_1_hom_2	1145.5	0	2	2090.7	0	2	1384.8	0	1246	1799.4	0	2519
TW.f_R_1_hom_3	2707.7	4	31	4277.4	9	17	3031.0	7	14971	3774.2	6	15039
TW.f_R_1_hom_4	2688.7	1	52	4053.9	12	26	3165.2	4	30960	3676.5	7	32787
TW.h_R_1_hom_1	1105.2	1	4	1807.7	3	7	1306.5	2	1889	1639.4	1	2652
TW.h_R_1_hom_2	1553.0	0	12	2130.4	9	15	1735.7	1	7285	2151.3	0	17666
TW.h_R_1_hom_3	2266.5	7	46	2990.3	22	66	2537.4	13	22523	2829.7	16	27810
TW.h_R_1_hom_4	2488.7	2	93	2768.7	24	173	2770.4	11	62857	2913.0	11	92204
TW.r_R_1_hom_1	1101.3	1	4	1906.5	2	6	1445.8	1	1854	1604.5	1	2786
TW.r_R_1_hom_2	1560.0	0	13	2260.9	6	13	1836.5	1	7325	2073.0	1	11856
TW.r_R_1_hom_3	2498.7	4	55	2549.0	40	79	2707.7	11	28977	2893.5	10	40499
TW.r_R_1_hom_4	2375.7	3	71	2702.6	32	106	2733.9	11	46710	2593.0	19	67312
TW.d1_R_2_hom_1	2291.4	2	1	2769.5	2	2	2534.3	2	657	2707.6	2	1676
TW.d1_R_2_hom_2	3411.6	5	4	3792.4	15	3	3492.9	13	2035	3763.6	12	3803
TW.d1_R_2_hom_3	4467.8	22	12	4956.7	31	5	4823.8	32	3594	4900.2	34	2742
TW.d1_R_2_hom_4	4410.5	37	22	4969.5	60	9	4710.5	54	8440	4741.0	55	8042
TW.d2_R_2_hom_1	1331.4	2	3	2608.6	2	2	1567.6	2	1209	1727.6	2	1971
TW.d2_R_2_hom_2	1738.5	4	8	3521.7	4	6	2112.4	4	5206	2495.2	4	11402
TW.d2_R_2_hom_3	2881.0	4	33	4787.1	8	11	3333.2	6	14383	3692.2	5	23146
TW.d2_R_2_hom_4	3513.3	9	60	4999.0	20	20	3427.6	17	35536	3831.4	18	38436
TW.f_R_2_hom_1	1339.0	2	2	2501.9	2	2	1572.4	2	1081	2197.1	2	2930
TW.f_R_2_hom_2	1811.9	4	7	3390.7	4	5	2148.1	4	4512	3087.3	4	8341
TW.f_R_2_hom_3	2954.4	4	29	4494.5	12	15	3373.9	6	12888	4087.9	5	13915
TW.f_R_2_hom_4	3265.7	10	50	4347.6	20	25	3592.4	18	27822	4166.7	17	19028
TW.h_R_2_hom_1	1268.6	2	2	1869.5	7	3	1466.7	2	1398	1663.8	5	2091
TW.h_R_2_hom_2	1601.7	4	10	2314.7	9	15	2030.1	4	6975	2279.0	5	7954
TW.h_R_2_hom_3	2267.1	6	42	2383.1	34	72	2622.1	11	19423	2629.1	18	29474
TW.h_R_2_hom_4	2727.6	16	81	2557.1	61	128	2985.7	25	48078	3139.0	29	57724
TW.r_R_2_hom_1	1132.4	2	3	1674.3	7	4	1417.1	2	1475	1541.9	2	1624
TW.r_R_2_hom_2	1503.5	4	10	2578.5	5	12	1842.6	4	6482	2111.4	4	11709
TW.r_R_2_hom_3	2481.3	5	37	2619.2	35	60	2868.1	14	17836	2881.0	12	30735
TW.r_R_2_hom_4	2973.3	15	68	3558.1	38	48	3505.7	25	42296	3434.3	26	39780
TW.d1_R_6_hom_1	2413.2	0	1	2932.8	0	2	2873.4	0	631	2916.3	0	1766
TW.d1_R_6_hom_2	2795.1	3	7	4054.2	3	5	3572.3	3	3562	3852.9	3	7034
TW.d1_R_6_hom_3	4496.3	17	17	5027.1	30	7	4804.6	27	4694	4992.0	27	2948
TW.d1_R_6_hom_4	4631.1	16	22	5026.2	35	9	4673.5	29	8408	4814.8	27	6503
TW.d2_R_6_hom_1	1404.8	0	2	2713.8	0	2	1702.0	0	1166	2060.9	0	2494
TW.d2_R_6_hom_2	1625.5	3	13	3352.6	3	6	1881.2	3	6529	2379.7	3	12211
TW.d2_R_6_hom_3	2743.4	2	50	4914.5	6	14	3071.1	4	20234	3632.3	4	25544
TW.d2_R_6_hom_4	2822.8	3	65	4851.7	7	17	3130.2	6	38330	3579.7	6	55518
TW.f_R_6_hom_1	1441.1	0	2	2693.9	0	2	1694.3	0	1067	2343.9	0	2393
TW.f_R_6_hom_2	1668.9	3	11	3100.6	3	7	2010.5	3	5474	2678.8	3	8215
TW.f_R_6_hom_3	2847.7	2	39	4391.1	7	20	3136.6	5	16332	4182.5	4	11064
TW.f_R_6_hom_4	2880.0	3	52	4292.3	12	24	3274.2	7	29221	3904.6	7	29138
TW.h_R_6_hom_1	1268.3	0	2	1761.5	4	4	1452.1	3	1519	1586.4	1	2006
TW.h_R_6_hom_2	1488.0	3	13	2332.6	7	15	1886.8	4	7777	2103.7	4	13214
TW.h_R_6_hom_3	2520.0	3	60	3202.2	29	75	2968.6	13	28453	2900.3	13	30533
TW.h_R_6_hom_4	2359.4	7	76	2254.2	49	160	2502.5	13	49839	2804.3	15	78774
TW.r_R_6_hom_1	1174.7	0	3	2448.4	1	3	1520.4	0	1436	1793.4	0	1907
TW.r_R_6_hom_2	1390.2	3	13	2099.1	10	16	1748.3	4	7623	1903.4	3	11141
TW.r_R_6_hom_3	2698.2	2	75	2882.8	27	106	2992.6	11	30495	3141.2	9	47257
TW.r_R_6_hom_4	2321.5	6	78	2691.7	32	126	2660.3	13	46607	2808.9	15	75305
Average	2321.4	4.8	27.1	3253.0	14.4	27.3	2633.1	8.7	14622.6	2937.2	8.9	19552.5

updated during the search process. Aiming at improving results of a recent consensus function [10], our function allows vehicles to return to the depot in order to pickup requests even if they have not completed their routes, and requests can be rejected (i.e., reassigned to a third-party logistic operator by paying a cost).

The computational results of the static, dynamic, and dynamic-stochastic versions over different geographies, arrival rates, and time windows have indicated the proposed method is quite effective to solve the problem when sampled-scenarios are taken into consideration. In general, there is an overall average increase in the solution cost, considering the static problem, of 42.3%, compared with the dynamic, 15.5%, compared with the dynamic-stochastic that uses the generalized route generation function, and 29.3%, compared with the dynamic-stochastic that uses the consensus function in [10]. In terms of runtime, this increase is of 1.5%, 56061.8%, and 74295.4%, respectively.

Future works will focus on reducing the total runtime of the proposed method, including a study on the number of scenario samples, sampling horizon, vehicles, and iterations of the ALNS. One direction might also consider a parallel version of the proposed method.

Acknowledgements Thiago Alves de Queiroz would like to thank CNPq (grant number 308312/2016-3) and FAPEG for their financial support.

References

1. Azi, N., Gendreau, M., Potvin, J.Y.: A dynamic vehicle routing problem with multiple delivery routes. Annals of Operations Research **199**(1), 103–112 (2012)
2. Battarra, M., Cordeau, J.F., Iori, M.: Pickup-and-delivery problems for goods transportation. In: P. Toth, D. Vigo (eds.) Vehicle Routing: Problems, Methods, and Applications, 2nd edn., chap. Chapter 6, pp. 161–191. SIAM (2014)
3. Campbell, A.M., Savelsbergh, M.: Incentive schemes for attended home delivery services. Transportation Science **40**(3), 327–341 (2006)
4. Cattaruzza, D., Absi, N., Feillet, D.: The multi-trip vehicle routing problem with time windows and release dates. Transportation Science **50**(2), 676–693 (2016)
5. Gunawan, A., Lau, H.C., Vansteenwegen, P.: Orienteering problem: A survey of recent variants, solution approaches and applications. European Journal of Operational Research **255**(2), 315–332 (2016)
6. Ho, S.C., Szeto, W., Kuo, Y.H., Leung, J.M., Petering, M., Tou, T.W.: A survey of dial-a-ride problems: Literature review and recent developments. Transportation Research Part B: Methodological **111**, 395–421 (2018)
7. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science **40**(4), 455–472 (2006)
8. Srour, F.J., Agatz, N., Oppen, J.: Strategies for handling temporal uncertainty in pickup and delivery problems with time windows. Transportation Science **52**(1), 3–19 (2018)
9. Ulmer, M.W., Thomas, B.W., Mattfeld, D.C.: Preemptive depot returns for dynamic same-day delivery. EURO Journal on Transportation and Logistics (2018)
10. Voccia, S.A., Campbell, A.M., Thomas, B.W.: The same-day delivery problem for online purchases. Transportation Science **53**(1), 167–184 (2019)