A New Variant of Dynamic Pickup and Delivery Problem with Time Windows

Petr Valenta and **Hana Rudová** and **Soumen Atta**
Faculty of Informatics, Masaryk University, Brno, Czech Republic
xvalent4@mail.muni.cz, {hanka, atta}@fi.muni.cz

Abstract

Motivated by the challenges faced by a logistics company, we present a new variant of the dynamic capacitated pickup and delivery problem with time windows (PDPTW) where excessive changes of unaffected routes are undesirable. In real-life scenarios, different dynamism sources such as canceled requests, change of demands, change of pickup, or delivery time windows often disrupt the existing planning of routes. The static PDPTW is solved with the current information about the problem well before executing the routes, such as the previous night. We present an algorithmic idea of a dynamic solver quickly addressing changes that occur due to the dynamism while avoiding excessive modifications to the previous solution. Since the company has not yet the dynamic data, new dynamic instances are generated from the existing static PDPTW instances in the literature. Preliminary results demonstrate that we can quickly incorporate the required changes. Future perspectives of this ongoing work are discussed in the end.

Introduction

The pickup and delivery problem with time windows (PDPTW) is a generalization of the classical vehicle routing problem and the vehicle routing problem with time windows (Savelsbergh and Sol 1995; Braekers, Ramaekers, and Nieuwenhuyse 2016). PDPTW is a combinatorial optimization problem with applications in transportation and logistics having significant economic importance.

In PDPTW, a given homogeneous or heterogeneous fleet of vehicles is used to satisfy a set of transportation requests. Each request is a pair of a pickup location and a delivery location, and a fixed amount of commodity has to be transported between them. This article considers a single depot from where the homogeneous fleet of vehicles starts their routes and again travels back to the depot. Each location is associated with a time window within which the vehicle must arrive at the location. If a vehicle arrives at a location before starting its time window, the vehicle must wait for loading or unloading. The time required for loading or unloading is known as the service time. Each vehicle has its loading capacity limit beyond which it cannot carry a commodity. The solution to an instance of PDPTW is a set of routes that obeys some constraints. In each route, only one vehicle is used. A single vehicle must do the pickup at first and then the corresponding delivery of a transportation request. The total load of a vehicle must be at most its loading capacity at any time. Additionally, the time window of each location must be maintained. The primary objective of PDPTW is to minimize the number of vehicles used (i.e., the number of routes), and the secondary objective aims to minimize the total distance traveled by all the vehi:les (Curtois et al. 2018).

Our present work considers the dynamic PDPTW, which can be regarded as a series of some static PDPTWs (Gendreau et al. 1999). In the literature, a problem is said to be dynamic when input data on the problem known to the decision maker is updated concurrently with the determination of the routes (Psaraftis, Wen, and Kontovas 2016). The sources of dynamism may be the cancellation of requests, change of demands, change of pickup, or delivery time windows. Although any changes to the unprocessed routes are considered in the literature on dynamic problems, we do not allow arbitrary modifications to all the routes as some routes may already be fixed due to external transport providers. Surprisingly, we are not aware of any approaches where some prohibition or minimizing changes exist to unprocessed requests and routings.

This problem is directly motivated by the dynamic problem existing in the logistics company Wereldo. This paper formulates a base version of their dynamic problem and discusses an algorithmic idea for our approach. Preliminary results show that our dynamic solver can quickly address required changes for generated problem instances. Furthermore, we will discuss perspectives for future extension of this problem and approach to consider complex real-life scenarios.

Dynamic Problem with Limited Changes

Based on discussions with the company, we propose a new variant of the dynamic PDPTW. Here, a feasible solution (i.e., the set of routes) is computed overnight for available transportation requests. The most recent changes must be incorporated by the dispatcher, who starts his work with the existing solution, having possibly removed requests due to the changes. Such an initial solution is incomplete as new requests may have arrived, and modified requests were removed from existing routes. On top of that, we have information about vehicles which routes were affected by the changes. For this paper, we will work with a homogeneous fleet of vehicles and a single depot.

The primary objective is to minimize the number of routes, while the secondary objective aims to reduce the vehicles' total traveled distance. All the constraints for a static PDPTW are maintained here. Moreover, an additional constraint is added, which prohibits the removal of requests from the unaffected routes. It corresponds to the real-world scenarios where excessive changes to the unaffected routes are undesirable. However, the insertion of new requests to the unaffected routes and reordering existing requests within its routes are allowed. Any changes to the affected routes are allowed.

Sketch of Solution Methodology

This section provides a sketch of the solution methodology for solving the dynamic PDPTW described in the previous section. It is based on the iterated greedy heuristic proposed in (Qu and Curtois 2017), which is combined with an intra-route neighborhood operator introduced in (Li and Lim 2003). The operator is a hill climbing algorithm that can improve the existing solution in a local neighborhood.

Since it is possible to process any changes on affected vehicles (with removed or modified requests), their requests are removed from the initial solution. Also, modified requests are added to unassigned requests. The algorithm starts with an incomplete solution and tries to insert all unassigned requests into existing routes. A pool of feasible insertions is computed for each unassigned request. An insertion is picked based on the increase of the solution cost caused by the insertion. If no insertions are possible, and the vehicle limit has not been reached, a new vehicle is added. This process is repeated until all requests are assigned or until no insertions are feasible, and the vehicle limit has been reached. If a new feasible solution was found, the intra-route neighborhood operator is applied to improve it by reordering requests inside particular routes. The overall procedure is repeated until the time limit is exceeded, e.g., 5 seconds. Additional optimizations are also tested to make the runtime faster and to improve the quality of solutions.

Benchmark Instances

To our knowledge, there is no standard reference benchmark for the dynamic PDPTW, which would allow us to compare the proposed methods objectively. Various works use the dynamic extension of Solomon's benchmarks (Solomon 1987), having up to 100 customers only. Still, time windows are available here. Larger problems with up to 385 customers derived by (Kilby, Prosser, and Shaw 1998) are also available. However, these do not consider even time windows.

Since the company currently does not have any dynamic data, the dynamic PDPTW is generated from the well-known benchmark dataset for the static PDPTW (Li and Lim 2003) available at SINTEF's website[1]. We will take the static problem together with its best existing solutions. We apply two types of modifications to these solutions. We either cancel a fixed number of requests or change the demands of a fixed number of requests by multiplying their demand with a random number from [0.75, 1.25]. Two heuristics are used for selecting these requests from the existing solutions. In the first heuristic, they are chosen randomly. In the second heuristic, a route is randomly selected, and several other routes are selected based on their proximity to the first route. From these routes, the required number of requests is removed/modified randomly.

We apply our solver on the generated problems with initial partial solutions and unassigned transportation requests given by modified requests. Random instances generated by the first heuristics are considered together with instances where requests from close routes were modified or removed. This specific modification is realized to make sure that there is some opportunity for improvement of the solution. It might not be the case for the first random instances since the best-known solutions, together with changes spread over time and space, might be tough to improve.

Preliminary Results and Discussions

To demonstrate our current approach, we will present some preliminary results on instances with 200 requests, i.e., 400 locations to be served. In Figure 1, we present experiments with 10, 20, and 30 canceled requests generated from three static instances[1] lc1_4_5, lr1_4_4, and lrc1_4_6 for clustered locations (LC), randomly distributed locations (LR), and a combination of both (LRC), respectively. Canceled requests are selected using the second heuristics taking into account their proximity. One hundred dynamic data instances are generated for each number of canceled requests and each static instance. Each run of the solver was limited by 5,000 iterations in total (one iteration corresponds to the insertion of one request), which corresponds up to 3 seconds on a standard computer (CPU Intel Core i7-5600U at 2.6 GHz).

We compare our results with the initial solution. Since only request cancellations are considered in this experiment, the valid and complete initial solution can be created from the best-known solutions available at the SINTEF website[1] by removing canceled requests. Figure 1 shows that our solver reduces the number of used vehicles compared to the initial solution on clustered (LC) and random-clustered instances (LRC). On random instances (LR), the number of vehicles was not decreased, but the total distance (secondary objective) was reduced in most cases.

Future Perspectives

This work introduces an initial step in the long-term project in cooperation with the company Wereldo, aiming to develop the solution approach while sharing our experiences and working on real-life data. There are various directions where we would like to extend our dynamic approach, and we will discuss the most important of them here.

[1]http://www.sintef.no/projectweb/top/pdptw/li-lim-benchmark

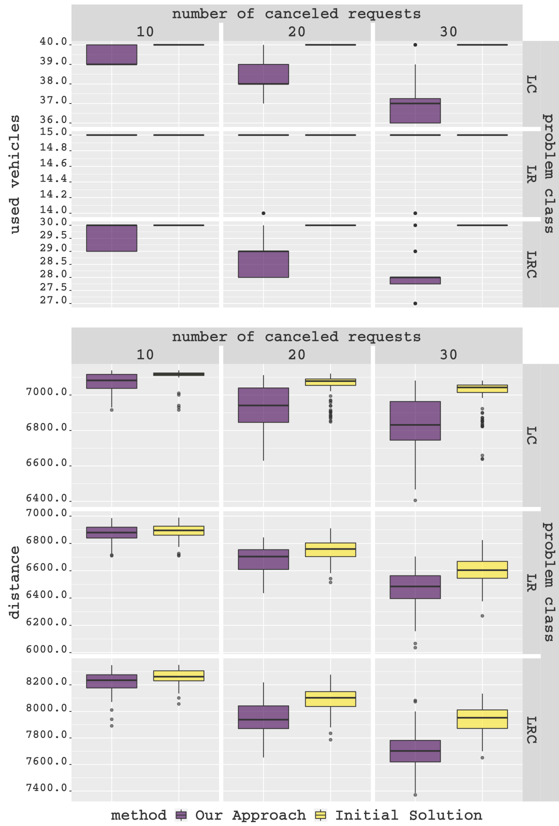

Figure 1: Comparison on instances with canceled requests

We plan to study the impact of changes on affected routes when they are still feasible (e.g., demand is decreased, or request is canceled). Based on the discussion with the company, it may still be desirable to keep some of these routes stable when the profit from rearrangement is negligible.

It is necessary to consider heterogeneous vehicles having different capacities (Baldacci, Battarra, and Vigo 2008; Brandao 2011). Variable cost related to the distance traveled is sometimes considered (Liu 2013; Brandao 2011), studies where the cost of transportation reflects the current load of vehicles or the minimal cost of vehicles are not so common. Based on the data model from the company, we plan to take into account the cost optimization based on market prices.

Our ultimate problem consists of the routing problems coming from various customers, having potentially own vehicle fleet. Particular customers may share their vehicle fleets as well as use external transport providers. Indeed, it introduces a large scale problem (Arnold, Gendreau, and Sörensen 2019). Even before working on scalability issues, we must work on additional problem complexities. A common problem is introduced by the inclusion of multiple depots (Montoya-Torres et al. 2015). Also, consideration of separate outbound requests (from depot to customers) or inbound requests (from customers to depot) is necessary.

Finally, we need to consider both the dynamic and static solution approaches, which, however, introduces a different part of our work.

Conclusion

Motivated by problems existing in the logistics company, a new variant of dynamic PDPTW was introduced where excessive changes of unaffected routes are avoided. A sketch of the solution methodology was presented, and benchmark problems were proposed to verify our ideas before having real-life data from the company. We have presented initial experiments and concluded our ongoing work by the discussion of long-term perspectives.

References

Arnold, F.; Gendreau, M.; and Sörensen, K. 2019. Efficiently solving very large-scale routing problems. *Computers & Operations Research* 107:32–42.

Baldacci, R.; Battarra, M.; and Vigo, D. 2008. Routing a heterogeneous fleet of vehicles. In Golden, B.; Raghavan, S.; and Wasil, E., eds., *The Vehicle Routing Problem: Latest Advances and New Challenges*, 3–27. Springer US.

Braekers, K.; Ramaekers, K.; and Nieuwenhuyse, I. V. 2016. The vehicle routing problem: State of the art classification and review. *Computers & industrial eng.* 99:300–313.

Brandao, J. 2011. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers & Operations Research* 38(1):140–151. Project Management and Scheduling.

Curtois, T.; Landa-Silva, D.; Qu, Y.; and Laesanklang, W. 2018. Large neighbourhood search with adaptive guided ejection search for the pickup and delivery problem with time windows. *EURO Journal on Transportation and Logistics* 7(2):151–192.

Gendreau, M.; Guertin, F.; Potvin, J.-Y.; and Taillard, E. 1999. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation science* 33(4):381–390.

Kilby, P.; Prosser, P.; and Shaw, P. 1998. Dynamic VRPs: A study of scenarios. Technical report, Univ. of Strathclyde.

Li, H., and Lim, A. 2003. A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools* 12(02):173–186.

Liu, S. 2013. A hybrid population heuristic for the heterogeneous vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review* 54:67–78.

Montoya-Torres, J. R.; López Franco, J.; Nieto Isaza, S.; Felizzola Jiménez, H.; and Herazo-Padilla, N. 2015. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering* 79:115–129.

Psaraftis, H. N.; Wen, M.; and Kontovas, C. A. 2016. Dynamic vehicle routing problems: Three decades and counting. *Networks* 67(1):3–31.

Qu, Y., and Curtois, T. 2017. Job insertion for the pickup and delivery problem with time windows. *Lecture Notes in Management Science* 9:26–32.

Savelsbergh, M. W., and Sol, M. 1995. The general pickup and delivery problem. *Transportation science* 29(1):17–29.

Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problems. *Operations research* 35:254–265.