Computers and Operations Research 92 (2018) 87–97

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Large neighborhood search with constraint programming for a vehicle routing problem with synchronization constraints

Hossein Hojabri[a,c], Michel Gendreau[b,c], Jean-Yves Potvin[a,c,*], Louis-Martin Rousseau[b,c]

[a] Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succ. Centre-Ville, Montréal, Québec, H3C 3J7, Canada
[b] Département de mathématiques et de génie industriel, École Polytechnique de Montréal, C.P. 6079, succ. Centre-Ville, Montréal, Québec, H3C 3A7, Canada
[c] Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Université de Montréal, C.P. 6128, succ. Centre-Ville, Montréal, Québec, H3C 3J7, Canada

A R T I C L E I N F O

Article history:
Received 8 June 2016
Revised 20 November 2017
Accepted 21 November 2017
Available online 8 December 2017

Keywords:
Vehicle routing
Synchronization
Large neighborhood search
Constraint programming

A B S T R A C T

This paper considers an extension of the vehicle routing problem with time windows, where the arrival of two vehicles at different customer locations must be synchronized. That is, one vehicle has to deliver some product to a customer, like a home theater system, while the crew on another vehicle must install it. This type of problem is often encountered in practice and is very challenging due to the interdependency among the vehicle routes, but has received little attention in the literature. A constraint programming-based adaptive large neighborhood search is proposed to solve this problem. The search abilities of the large neighborhood search and the constraint propagation abilities of constraint programming are combined to determine the feasibility of any proposed modification to the current solution. Numerical results are reported on instances derived from benchmark instances for the vehicle routing problem with time windows with up to 200 customers.

1. Introduction

The Vehicle Routing Problem with Multiple Synchronization constraints (VRPMS) can be used to model many real-world applications, like home care delivery, aircraft fleet assignment, vehicle routing and scheduling, and forest operations, as reported in Rousseau et al. (2013). The problem considered in this paper comes from a software development company that produces vehicle routing solutions for large retailers involved in the delivery of large items to their customers, like furniture, appliances, home theater systems, etc. In the case of electronics equipment, the crew in the delivery vehicle simply unload the goods, while another crew with the required capabilities takes charge of the installation. In this case, the routes of two different types of vehicles must be synchronized at a certain number of customer locations. A dynamic version of a similar problem for pickup vehicles (i.e., goods are collected at customer locations rather than being delivered) is addressed in Rousseau et al. (2013). The authors consider a problem where new customer requests are inserted one by one into the current routes, as they occur. Constraint programming (CP), a computational paradigm based on constraint propagation, is used to determine if an insertion is feasible or not. Since only one customer is considered at a time, this approach proved to be viable. Surprisingly and to the best of our knowledge, no solution approach, neither exact nor heuristic, has been proposed for the static version of the problem where all customers are known in advance. This a priori knowledge is required in the case of delivery applications, because the vehicles are loaded before they depart from the depot. The goal of this paper is thus to provide a contribution in this regard. In particular, we exploit the constraint propagation capabilities of CP in a context where many customers must be handled at the same time, through a proper integration within the Adaptive Large Neighborhood Search (ALNS) framework.

Apart from the paper of Rousseau et al. (2013), a problem quite similar to ours is addressed in Bredström and Rönnqvist (2008) in the domain of home care crew scheduling. The authors formally state their problem as a mixed integer programming (MIP) model that accounts for temporal precedence and synchronization constraints. The problem is then solved through an optimization-based heuristic. Rasmussen et al. (2012) solve the same problem exactly with a branch-and-price algorithm. Due to the application context, there is no capacity constraint and specific issues about home care crews are taken into account, in addition to the synchronization requirements, like care giver preferences, customer priority and ability of a particular care giver to serve a given customer. Also, there is no need to serve all customers (i.e., visits can be rescheduled or canceled).

* Corresponding author.
E-mail address: potvin@iro.umontreal.ca (J.-Y. Potvin).

https://doi.org/10.1016/j.cor.2017.11.011
0305-0548/

Schmid et al. (2010) report a MIP for the delivery of concrete to construction sites from several plants by a fleet of heterogeneous vehicles. Two hybrid approaches are introduced to solve the problem by combining an exact algorithm with a variable neighborhood search (VNS). El Hachemi et al. (2013) consider an application in the forest industry where trucks and log-loaders must be synchronized. A decomposition approach is presented to solve a weekly problem in two phases. The first phase determines the assignment of forest areas to wood mills. Then, the daily routing and scheduling for the transportation of logs is done with either a constraint-based local search alone or a hybrid method consisting of CP coupled with the constraint-based local search. Other types of synchronization requirements are also found in urban mass transit systems where drivers must change buses at relief points (Freling et al., 2003; Haase et al., 2001) or in public transit rail systems where passengers must transfer several times to reach their destination (Wong et al. 2008).

Ioachim et al. (1999) propose a column generation approach for an aircraft fleet assignment and routing problem where a number of flight subsets during the weekdays must depart at the same time. The master problem is a set partitioning problem with synchronization constraints, while the subproblem is a shortest path problem with time windows and linear costs on the time variables. An extension of this work for a periodic airline fleet assignment with time windows is also reported in Bélanger et al. (2006).

Li et al. (2005) report an integer programming model for a manpower allocation problem with time windows and job-teaming constraints. To perform a job, the required workers must show up together at the job location before it can be executed. Furthermore, nobody can leave before the job is completed. The authors propose two construction heuristics embedded within a simulated annealing framework. They provide comparisons with optimal solutions produced by a commercial solver and lower bounds obtained with a network flow model on their own set of test instances. Dohn et al. (2009) also solve the problem with a branch-and-price algorithm where branching on the time variables automatically imposes synchronization.

The interested reader will find in Drexl (2012) a comprehensive survey on synchronization, with a particular emphasis on vehicle routing applications. In this survey, different types of synchronization requirements are presented (spatial, temporal, etc.), as well as different applications (dial-a-ride, school bus routing, truck-and-trailer, etc.), but nothing quite similar to our problem.

The remainder of this paper is organized as follows. Our problem is first introduced in Section 2. A description of the solution methodology is provided in Sections 3 and 4. A few refinements to this methodology are proposed in Section 5. Then, numerical results are reported in Section 6. Finally, concluding remarks are made in Section 7.

2. Problem definition and formulation

Our problem can be seen as an extension of the Vehicle Routing Problem with Time Windows (VRPTW) with the addition of synchronization requirements. In a standard VRPTW, the routes are independent, in the sense that a modification to a given route has no impact on the other routes. This is not the case for the VRPMS because the routes are interdependent due to the synchronization requirements at certain customer locations.

Let us assume that two different types of vehicles are available, called regular (delivery) and special vehicles. The regular vehicles deliver goods to customers and have limited capacity. The special vehicles do not deliver goods but rather provide a service related to the goods delivered by the regular vehicles. The set of customers is divided into two subsets: regular customers visited only by regular vehicles, and special customers visited by both a regular and a special vehicle. A regular customer is represented by a single vertex in the transportation network, while a special customer is represented by two vertices, one for the delivery by the regular vehicle and another for the service provided by the special vehicle.

A regular vertex has a demand and a time window, while a special vertex has only a time window, which is defined relative to the delivery time of the regular vehicle. More precisely, if t is the delivery time at the regular vertex of a special customer, then the service of the special vertex must begin within the time window $[t-\delta,\ t+\gamma]$, where δ and γ are parameters. Different synchronization requirements can be represented with different values of δ and γ. For example, if the service of the special vehicle cannot start before the delivery, then the time window's lower bound of the special vertex is set to t (i.e., $\delta = 0$); if the service of the special vehicle cannot start before the end of the delivery, then the time window's lower bound of the special vertex is set to t plus the service time of the regular vehicle (i.e., δ is set to minus the service time of the regular vehicle), etc.

The problem then consists of constructing routes for the two types of vehicles such that:

1. Each route begins and ends at a single depot;
2. Each regular customer is served in the route of exactly one regular vehicle;
3. Each special customer is served in the route of exactly one regular vehicle and one special vehicle;
4. The total demand on the route of a regular vehicle cannot exceed its capacity;
5. A regular vehicle must start its delivery at a regular or special customer within the time window of the corresponding regular vertex;
6. A special vehicle must start its service at a special customer within the time window of the corresponding special vertex, which is defined relative to the delivery time of the regular vehicle at the regular vertex;
7. A regular (special) vehicle is allowed to wait up to the lower bound of the time window of the regular (special) vertex if it arrives earlier.

The objective is to minimize the total distance traveled by all vehicles (regular and special).

In the following, a small example is given before providing a more formal CP-based formulation of our problem.

2.1. Example

We consider a small example where the load and time window constraints of the regular vehicles are relaxed in order to focus on the synchronization issue. We have two regular vehicles $\{v_1^r, v_2^r\}$, one special vehicle $\{v_1^s\}$, a single depot 0 and 5 customers {1, 2, 3, 4, 5}, two of which are special customers, namely {1, 2}. In the underlying network, vertex 0 is the depot, vertices 1–5 correspond to regular vertices, while vertices 6 and 7 correspond to the special vertices of customers 1 and 2. Therefore, the vertex pairs (1,6) and (2,7) require synchronization. In Fig. 1, each pair is aggregated into a single gray circle. Without loss of generality, we assume that the service or dwell time is equal to 1 and $\delta = \gamma = 0$ for every special vertex.

In the solution shown, solid arcs are used to represent the routes of regular vehicles and dashed arcs for the route of the special vehicle. Each arc is labeled with its travel time and each vertex i is labeled with the vehicle's arrival time t_i. The return time of each vehicle to the depot, denoted t_0, is also indicated on the last arc of the corresponding route. The routing plan is the following:

$v_1^r : 0, 5, 1, 0$

$v_2^r : 0, 2, 3, 4, 0$