Vehicle Routing with Transportable Resources: Using Carpooling and Walking for On-Site Services

Marc-Antoine Coindreau[a], Olivier Gallay[a,*], Nicolas Zufferey[b]

[a] *Department of Operations, HEC, University of Lausanne, CH-1015 Lausanne, Switzerland*

[b] *Geneva School of Economics and Management, GSEM – University of Geneva, 1211 Geneva 4, Switzerland*

Abstract

In the classical *Vehicle Routing Problem* (VRP), it is assumed that each worker moves using an individually assigned vehicle. Removing this core hypothesis opens the door for a brand new set of solutions, where workers are seen as transportable resources that can also move without the help of a vehicle. In this context, motivated by a major European energy provider, we consider a situation where workers can either walk or drive to reach a job and where carpooling is enabled. In order to quantify the potential benefits offered by this new framework, a dedicated *Variable Neighborhood Search* is proposed to efficiently tackle the underlying synchronization and precedence constraints that arise in this extension of the VRP. Considering a set of instances in an urban context, extensive computational experiments show that, despite conservative scenarios favoring car mobility, significant savings are achieved when compared to the solutions currently obtained by the involved company. This innovative formulation allows managers to reduce the size of the vehicle fleet while keeping the number of workers stable and, surprisingly, decreasing the overall driving distance simultaneously.

Keywords: Routing, On-Site Services, Synchronization, Carpooling, Variable Neighborhood Search.

*Corresponding author

Email addresses: marc-antoine.coindreau@unil.ch (Marc-Antoine Coindreau), olivier.gallay@unil.ch (Olivier Gallay), n.zufferey@unige.ch (Nicolas Zufferey)

1. Introduction

1.1. Industrial context

Transportation in urban areas is increasingly facing new challenges. On the one hand, the systematic use of cars produces hazardous impacts on the environment, such as noise, toxic emissions, and the effects induced by greenhouse gases (Knörr 2008). On the other hand, as highlighted by Jabali et al. (2012), city centers suffer from congestion and limited parking space. These phenomena, which are magnified by low vehicle occupancy rates, decrease the intrinsic efficiency of car-based transportation. Consequently, current legislation tends to constrain the use of cars within city centers either by limiting the number of authorized vehicles or completely banning vehicles in specific areas, such as pedestrian zones, as highlighted by Parragh and Cordeau (2017). For all these reasons, reducing the systematic use of cars in urban areas is becoming increasingly important. Firms that provide on-site services or parcel deliveries are directly concerned by these issues, as a substantial part of their activities takes place in metropolitan areas.

We focus on the case of a large European energy provider, denoted by EEP (it cannot be named because of a non-disclosure agreement), that routes technicians to provide on-site services (e.g., small maintenance work, consumption evaluations, and consumer-setting upgrades). Every day, technicians who are not assigned to clients are employed for heavy works on the electricity network. However, once assigned to on-site services, the workers cannot be re-assigned thereafter to heavy works, even if they terminate their working day earlier. Indeed, for the heavy works, teams of technicians are selected for the full day's work, and the jobs are frequently located outside of the cities. As a result, idle time arises in the workers' planning, either at the depot or on their route, due to the presence of time windows to serve the jobs. As each worker assigned to on-site services must be employed for the whole working day, EEP's current practice is to first minimize the number of technicians necessary to serve all jobs. In a second phase, EEP minimizes the remaining costs implied by the technicians' routes (i.e., vehicle fixed costs and total driving distance).

EEP manages thousands of workers in urban areas, who drive more than a million kilometers every year. In that respect, EEP aims to evaluate the savings potential generated by the use of walking to reduce the total costs of its routes while also meeting the workers' expectations. EEP observed that its technicians often leave their vehicles to perform clustered jobs on foot, even if their planning

would indicate driving to the next job. EEP also wants to go one step further by evaluating the savings potential of carpooling (i.e., using the same car to transport multiple workers), to scale down the size of its fleet and to possibly further reduce the overall driving costs.

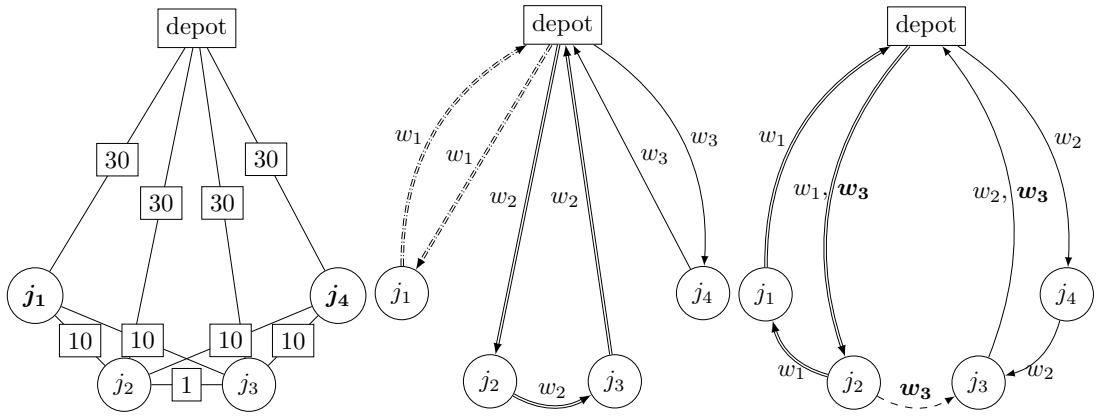
Introducing these alternative transportation options obviously presents significant challenges. It is necessary to build and manage routes that are highly synchronized. Possible waiting times must be efficiently managed, as drivers might have to wait for workers to be picked up, and non-motorized workers might have to wait for drivers to be transported. Competitive solutions must also ensure that the workers' productivity remains stable, which could be decreased by the slower walking speed and the detours imposed by carpooling to drop off and pick up non-motorized technicians.

1.2. Problem description

We consider the problem of routing a set of workers through different client locations in order to provide on-site jobs. Each job has a given duration and must be performed in a specific time window that is agreed upon with the involved client. This problem has garnered considerable interest in the research community in recent decades and is referred to as the *Vehicle Routing Problem* (VRP), or more specifically, as the *Vehicle Routing Problem with Time Windows* (VRPTW). In the VRPTW, each worker moves from one job to another by driving an individually assigned car. We propose a modeling framework that relaxes this assumption, and we consider an extension of the VRPTW in which workers are allowed to share a vehicle and to choose between walking or driving to reach their next job. The technicians can be separated from their vehicles and are seen as transportable resources that can move autonomously. We refer to this extension as the *Vehicle Routing Problem with Transportable Resources* (VRPTR), for which a full description of the considered assumptions is given in Section 3.1. While keeping the number of workers stable compared with EEP's current practice (i.e., VRP solutions), we aim at reducing the size of the vehicle fleet and/or the total driving distance. We allow for the modeling of every situation in which workers have to visit clients without any delivery or transportation of heavy equipment, making walking a viable option. This particularly occurs with various types of home services, such as health and elder care, IT support, household appliance repairs, and security checks.

A toy example is given in Figure 1, which illustrates how a VRPTR solution works. The characteristics of the instance are given in the left part of the figure. Compared with the VRP solution (middle part of the figure), the VRPTR solution (right part of the figure) provides improved effi-

ciency: the same number of workers, one car saved, and the total driving distance is reduced by 22.6%.



(a) Instance characteristics (b) VRP optimal solution (c) A VRPTR solution

(a) Values on the arcs denote the driving time (in minutes); walking is 10 times slower than driving; the planing horizon is 130 minutes; job durations are 60 minutes for j_1 and j_4, and 30 minutes for j_2 and j_3.
(b) and (c) The vehicle path is drawn with a specific line style; walking is represented with a dashed line; the label of an arc specifies which workers are using it.
(c) Worker w_3 is dropped off at j_2 by w_1 and then walks to j_3, where s/he is picked up by w_2. w_1 (resp w_2) works on j_1 (resp. j_4) after (resp. before) dropping off (resp. picking up) w_3.

Figure 1: Comparison between a VRPTR solution and the corresponding VRP optimal solution.

1.3. Contributions and outline

We develop both a mixed integer linear program (MILP) and a metaheuristic to solve the VRPTR. The latter uses a dedicated neighborhood structure and a fast insertion mechanism to tackle the increased complexity resulting from the introduction of walking and carpooling. Whereas the MILP is able to tackle instances up to 18 customers, the metaheuristic can solve all other instances, which involve up to 50 jobs. Compared with EEP's current practice (i.e., one vehicle assigned to each worker, no walking) on a representative set of instances capturing urban characteristics, the computational experiments yield an average improvement of 6.5% for the driving distance and 18.4% for the reduction of the vehicle fleet. We show that the introduction of carpooling and walking is able to generate a simultaneous gain, both in terms of fleet size and total driving distance, in 25% of the considered instances. We highlight and quantify the trade-off that might arise between removing cars and the resulting total driving distance. Finally, we study the existing relationship between the achieved gain and the specific instance characteristics.

The remainder of the paper is organized as follows. A literature review is presented in Section 2.

We formally describe the VRPTR and develop the associated MILP formulation in Section 3. In Section 4, we describe the proposed metaheuristic. Section 5 presents the computational experiments and the results. Section 6 proposes managerial insights (e.g., quantifying the gains compared with current practice and understanding the promising configurations for carpooling). Finally, concluding remarks and future research opportunities are presented in Section 7.

2. Literature review

The literature review is structured as follows. We first position the VRPTR with respect to the existing VRP formulations that also synchronize different resources (a formal review of VRP with synchronization constraints can be found in (Drexl 2012)). Next, we describe the solution methodologies that have proven to be efficient for such related problems.

Several studies consider the situation in which drivers and vehicles are allowed to disassemble along their route. Domínguez-Martín et al. (2018) examine a case where vehicles must start and terminate their route at different depots, whereas drivers have to come back to their starting depot. As a consequence, drivers must change vehicles during their route. Similarly, in the *Vehicle and Crew Routing Problem* (Lam et al. 2015), a vehicle is driven by different drivers to maximize its use. Although these contributions explicitly consider the synchronization between vehicles and workers, they do not address aspects related to carpooling and walking.

Levy and Bodin (1989) originally introduced the combined used of walking and driving for mail delivery purposes. It is referred to as *Park-and-Loop* and was generalized by Ghiani and Laporte (2001) as the *Location-Arc Routing Problem*. The postman parks her/his car, visits a subset of jobs, comes back to the car, and drives to the next customers. In the related contributions, the modeling differs from ours, as an arc-oriented approach is considered (i.e., workers must visit arcs and not nodes). A node-oriented approach was later considered by Gussmagg-Pfliegl et al. (2011) for a similar mail delivery application. Whereas these works acknowledge the advantages of combining walking and driving to serve on-site jobs, they do not address a potential reduction of the fleet size through the use of carpooling.

Other extensions of the VRP share a similar structure as *Park-and-Loop*, in particular when trucks and trailers can uncouple at specific locations to serve clients that cannot receive a truck paired with a trailer (thus, a lone truck stands for an on-foot worker, and a truck paired with a trailer stands for

a worker equipped with a vehicle). Such problems have been introduced as the *Partially Accessible Constrained VRP* by Rochat and Semet (1994) and Semet (1995). More recently, this research axis has received substantial attention under the *Truck and Trailer Routing Problem* (TTRP) (Chao 2002, Lin et al. 2009) or the *Vehicle Routing Problem with Trailers and Transshipments* (VRPTT) (Drexl 2014) formulations. As our contribution is not limited to the introduction of *Park-and-Loop* sub-tours in vehicle routes but also involves carpooling, these works cannot be directly applied to the present situation. Additionally, these formulations differ from ours since the motivation for *Park-and-Loop* sub-tours differs. In our case, this is due to customer restrictions in the TTRP and cost reductions in our case. Whereas the locations where the uncoupling of trailers can take place are limited to specific areas, a car can be parked at any client location in the present case.

In addition to *Park-and-Loop* aspects, the VRPTR involves the transportation of on-foot workers. Among these, Lin (2008) considers the synchronization of on-foot couriers with vans to deliver mail. However, and contrary to our formulation, Lin (2008) does not consider a complete synchronization of the resources, as on-foot couriers can only walk from the depot to a van or from a van to the depot. Fikar and Hirsch (2015), in a problem referred to as *Home Health Care Staff Scheduling*, addressed the situation where nurses have to visit patients in their homes. Nurses are allowed to walk but cannot drive a car. When walking is not possible, drivers, who are not permitted to visit patients, are employed to transport nurses by cars. Hence, the total number of workers (namely, nurses and drivers) is strictly greater than in the situation where nurses would drive their own cars (n.b., in the VRPTR, the technicians are both able to drive and perform jobs; hence, a reduction in the size of the vehicle fleet is achieved without increasing the number of employed workers).

In the context of parcel delivery (more generally, when the on-site presence of technicians is not mandatory), unmanned vehicles, such as drones (Wohlsen 2014) or robots (Daimler 2017), can be synchronized with vans to decrease the routing costs. Whereas, in the present case, unmanned vehicles would not be eligible to perform on-site services, the associated formulations share some similarities with the VRPTR. Indeed, both situations yield a similar modeling framework, where autonomous and transportable resources are dropped off and retrieved at different locations along the van routes. Although several recent contributions (e.g., (Murray and Chu 2015, Ferrandez et al. 2016, Poikonen et al. 2017, Agatz et al. 2018, Boysen et al. 2018)) have considered such types of synchronization, various limitations and specific constraints prevent adapting the associated solution approaches to the present case. Murray and Chu (2015) introduced a formulation called *Flying*

Sidekick Traveling Salesman Problem, in which drones can be transported by vans to deliver parcels at client locations for some parts of their routes. In this situation and typically for contributions in this specific research domain, several major discrepancies with the present formulation can be underlined. First, only one location can be visited by the drone between its drop-off and its pick-up. Second, a single van is considered; hence, the global synchronization aspects that follow from the possibility of a drone being dropped off and picked up by different vans is not addressed. Third, the considered objective differs as its focus is on minimizing the completion time (i.e., time windows are not considered). Finally, Boysen et al. (2018) assume that robots can wait indefinitely at the depot or at client locations. Such an assumption precludes its application in the present context, since the number of workers is limited and their employment is costly.

In the *Active Passive Vehicle Routing Problem* (APVRP) (Meisel and Kopfer 2014, Tilk et al. 2017), a set of trailers has to be transported with trucks from loading to unloading locations (i.e., pick-up and delivery requests). The duration of these operations is long enough to allow the trucks (i.e., the active transportation resources) to move other trailers (i.e., the passive transportation resources) in the meantime. Moreover, trailers can be carried by different vehicles (see (Smilowitz 2006) for an example of such a practice in drayage operations in the Chicago region). Creating bridges with the present study, a trailer can be seen as a non-motorized worker that requires transportation between different locations. However, the complexity is increased in our problem because the passenger transportation requests are not fixed a priori and are part of the decision-making process.

Most of the above-cited papers propose an exact formulation for the problem under study, which is able to solve instances of limited size (e.g., the MILP developed in (Murray and Chu 2015) is able to tackle instances involving up to 10 customers in a 10-square-mile region). The exact approaches are often complemented with a two-stage heuristic to find solutions for larger instances, either in a *cluster-first-route-second* or in a *route-first-cluster-second* fashion. The first alternative is aimed at initially building job clusters that will be visited by the transportable resources alone and then creating routes for the carrying vehicles to connect the clusters together (Levy and Bodin 1989, Fikar and Hirsch 2015). The second alternative proposes to first build routes for the carrying vehicles and then to assign some clients to the transportable resources (Ghiani and Laporte 2001, Gussmagg-Pfliegl et al. 2011, Murray and Chu 2015). Even though these two approaches are able to efficiently improve the quality of the initially generated solutions, they suffer from being easily trapped in a local minimum since the decision at the first stage strongly impacts the quality of the

decisions in the second one.

General metaheuristics based on the *ruin and recreate* principle have proven to be successful for various related VRP formulations (Schrimpf et al. 2000). These solution approaches do not suffer from the drawbacks of a two-stage methodology, as the decisions on the job clusters and the routing are made simultaneously. In a routing context, the *ruin and recreate* principle aims to improve a solution by iteratively removing and reinserting some jobs (one of the numerous fruitful implemtations is presented in (Pisinger and Ropke 2007)). Known as *Large Neighborhood Search* (LNS) and introduced by Shaw (1997), this principle has been the basis of multiple successful contributions in various domains. In particular, two related metaheuristics are developed in (Derigs et al. 2013) for the TTRP. They both combine the strength of a descent algorithm for the intensification with the exploration ability of LNS for the diversification. The authors highlight the benefit of combining a local search and a collection of neighborhood structures of different amplitudes, as in LNS.

3. Problem formulation

As the VRPTW is a special case of the VRPTR (where walking is forbidden and the number of vehicles is equal to the number of workers), the VRPTR can be classified as an NP-Hard problem (see Cordeau et al. (2007) for overviews of the various VRP characteristics, their associated models, and their efficient solution approaches).

3.1. Definition and assumptions

A walking path between a set of jobs is called a walking route (WR). Idle time is the total time that a worker waits in a solution (either en route or at the depot). Returning to the depot earlier at the end of the day is considered idle time, as workers are employed for the whole day, and they cannot be assigned to other tasks once they are back at the depot. For the EEP context, the following features are taken into account:

- The planning horizon is a day (i.e., the daily working time is upper bounded), for which all the jobs and travel information are accurately known (static data).
- For each worker, the walking limitations are the maximum daily walking distance (d_M^f) and the maximum allowed walking time between two jobs (τ_M^f).

- Vehicles and workers can disassemble and reassemble at any job location (the duration of this operation is assumed to be null).
- Each vehicle has a single assigned worker, meaning that the workers are separated into two categories: the drivers and the passengers (drivers have to perform their assigned jobs and to fulfill the transportation requests of passengers).
- Workers and vehicles start and end their routes at the depot.
- Both driver and passenger workers can walk to reach the next job on their routes. In the driver case, the return path to her/his car is mandatory (i.e., departure and arrival points of a WR must coincide), whereas a in the passenger case, departure and arrival points of a WR can be different.
- Idling is allowed for both the drivers and the passengers at job locations.

3.2. Graph modeling and variables

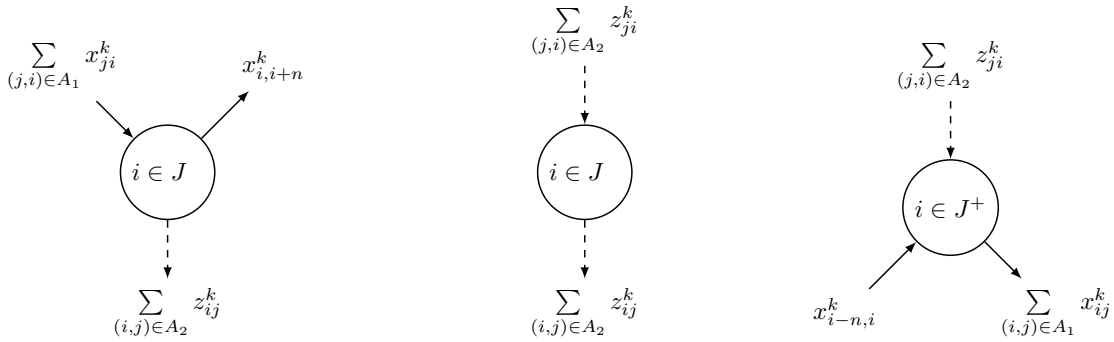
Let $J = \{1, \ldots, n\}$ be the set of jobs, K the set of motorized workers (i.e., drivers), and L the set of non-motorized workers (i.e., passengers). $W = K \cup L$ denotes the set of all workers. For job $j \in J$, $p_j \in \mathbb{R}^+$ is its processing time, and $(e_j, l_j) \in \mathbb{R}^{+2}$ is its time window, consisting of the earliest and latest possible service times. Between two jobs $(i, j) \in J^2$, the distance (in km) is given by $d_{ij} \in \mathbb{R}^+$, and the driving (resp. walking) time (in minutes) is denoted by $\tau_{ij} \in \mathbb{R}^+$ (resp. $\tilde{\tau}_{ij} \in \mathbb{R}^+$). $c \in \mathbb{N}$ indicates the maximum number of non-motorized workers allowed in a car (in addition to the motorized worker). Finally, $M_1 = \max_{j \in J} l_j + \max_{i \in J, j \in J} \tau_{ij}$ and $M_2 = \max_{j \in J} l_j + \max_{j \in J} p_j + \max_{i \in J, j \in J} \tilde{\tau_{ij}}$ are sufficiently large numbers, which are required for the MILP.

The node set J is duplicated using $J^+ = \{n+1, \ldots, 2n\}$, where $i \in J$ and $i + n \in J^+$ represent the same physical location, with $i \in \{1, \cdots, n\}$. These two sets allow distinguishing the different operations taking place at the same physical node (e.g., a motorized worker parks her/his car, starts a WR, and retrieves her/his car). J (resp. J^+) stands for the set of starting and intermediate points (resp. terminating points) of a WR. As a result, in the optimization model, a motorized worker starts a WR at $j \in J$ and finishes it at $j + n \in J^+$ (three different times are thus managed: arrival time, service time, and departure time). $V = J \cup J^+ \cup \{0, 2n+1\}$ is the set of all nodes, where 0 represents the starting depot and $2n + 1$ the ending depot. Based on this notation, $A_1 = \{(i, j) \in V \backslash \{2n+1\} \times V \backslash \{0\}$, such that $i \neq j$ and $j \neq i - n$ for $i \in J^+\}$ is the driving arc set and $A_2 = \left\{(i, j) \in J \times (J \cup J^+)$, such that $i \neq j$ and $\tilde{\tau}_{ij} < \tilde{\tau}^f_M\right\}$ is the walking arc set.

We define the variables:

- $x_{ij}^k = 1$ if motorized worker $k \in K$ uses arc $(i,j) \in A_1$; $x_{ij}^k = 0$, otherwise,
- $y_{ij}^{kl} = 1$ if non-motorized worker $l \in L$ is transported by the vehicle associated to motorized worker $k \in K$ on arc $(i,j) \in A_1$; $y_{ij}^{kl} = 0$, otherwise,
- $z_{ij}^w = 1$ if a worker $w \in W$ walks on arc $(i,j) \in A_2$; $z_{ij}^w = 0$, otherwise,
- t_i^w denotes the time at which worker $w \in W$ leaves node $i \in V$,
- s_i stands for the time at which the service starts at node $i \in J$.

Figure 2 illustrates the flow of a motorized worker when walking is involved. It is helpful to understand the coordination between a motorized worker and her/his vehicle as detailed in the constraints below.



(a) Start WR (b) Intermediate (c) End WR

Dashed (resp. plain) lines denote an edge traveled on foot (resp. with a car).
(a) Shows the arcs activated if motorized worker $k \in K$ starts a WR in $i \in J$.
(b) Shows the arcs activated if $i \in J$ is an intermediary point of a WR performed by a motorized worker $k \in K$.
(c) Shows the arcs activated if motorized worker $k \in K$ terminates a WR in $i \in J^+$.

Figure 2: Different flow configurations for a WR performed by a motorized worker.

Due to carpooling, and contrary to the standard VRP formulations, a worker can visit a node without performing the associated job. Indeed, when a non-motorized worker is dropped off at a node $i \in J$, the motorized worker (and potentially other non-motorized workers) stops at node i, but only the dropped worker performs the associated job. Consequently, with the introduced notation, a worker $w \in W$ completes the job at $i \in J$ if and only if w exits the node on foot (i.e., $\sum_{(i,j)\in A_2} z_{ij}^w = 1$). Figure 3 illustrates the different node sets and variables for the VRPTR

solution displayed in Figure 1. In this example, $J = \{1, 2, 3, 4\}$ and $J^+ = \{5, 6, 7, 8\}$ (e.g., nodes 1 and 5 represent the same physical node j_1), and node 0 (resp. node 9) represents the starting (resp. ending) depot.

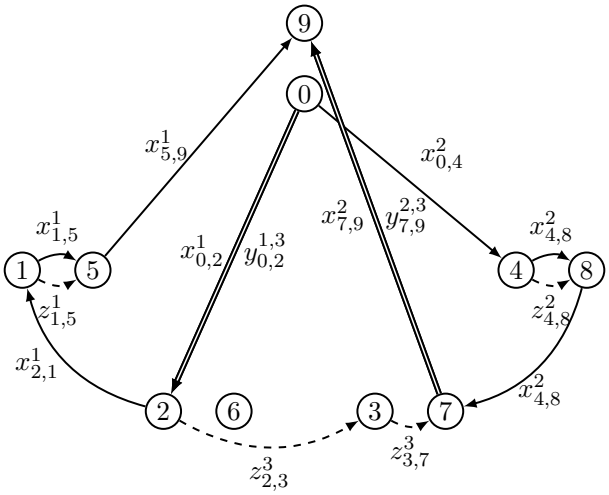


Figure 3: Modeling of the VRPTR solution displayed in Figure 1 using the introduced sets and variables

.

3.3. Mathematical formulation

We propose a MILP model for the VRPTR, where the numbers of both the involved workers and vehicles are given as inputs. More precisely, we fix the number of workers to the optimal value found when all workers are motorized. The costs associated with worker daily wages hence remain the same in the VRPTR solutions and in the corresponding optimal VRP solutions. To reduce the number of vehicles, the MILP is then launched sequentially, every time with one vehicle less, until no feasible solution can be found (i.e., a sequence of instances is thus generated in this way). The trade-off that appears between the size of the employed vehicle fleet and the total driving distance is discussed in Section 6.1.2.

For a fixed number of workers and a given vehicle fleet, Objective (1) minimizes the total driving distance (which constitutes the remaining transportation costs):

$$\text{minimize} \sum_{(i,j)\in A_1} \sum_{k\in K} d_{ij} \cdot x^k_{ij} \tag{1}$$

Constraints for workers and vehicles flows.

$$\sum_{(i,j)\in A_2} \sum_{w\in W} z_{ij}^{w} = 1, \qquad i \in J \tag{2}$$

$$\sum_{(i,j)\in A_2} d_{ij} \cdot z_{ij}^{w} \leq d_M^f, \qquad w \in W \tag{3}$$

$$\sum_{(j,i)\in A_1} x_{ji}^{k} = \sum_{(i,j)\in A_1} x_{ij}^{k}, \qquad i \in J \cup J^+,\, k \in K \tag{4}$$

$$\sum_{(i,j)\in A_2} z_{ij}^{k} \leq \sum_{(j,i)\in A_2} z_{ji}^{k} + \frac{1}{2}\left(\sum_{(j,i)\in A_1} x_{ji}^{k} + x_{i,i+n}^{k}\right), \qquad i \in J,\, k \in K \tag{5}$$

$$\sum_{(j,i)\in A_2} z_{ji}^{k} + \sum_{(j,i)\in A_1} x_{ji}^{k} \leq 1, \qquad i \in J,\, k \in K \tag{6}$$

$$\sum_{(j,i)\in A_2} z_{ji}^{k} = x_{i-n,i}^{k}, \qquad i \in J^+,\, k \in K \tag{7}$$

$$\sum_{(i,j)\in A_1} x_{ij}^{k} \leq 1, \qquad i \in V,\, k \in K \tag{8}$$

$$\sum_{(i,j)\in A_2} z_{ij}^{w} \leq 1, \qquad i \in J,\, w \in W \tag{9}$$

$$\sum_{i\in V} x_{0i}^{k} = \sum_{i\in V} x_{i,2n+1}^{k}, \qquad k \in K \tag{10}$$

Constraints (2) ensure that all the jobs are processed. Note that some nodes in J^+ might not be visited, as all jobs do not terminate a WR (e.g., see node 6 in Figure 3). Constraints (3) define an upper bound on the total daily walking distance of a worker (valid for both motorized and non-motorized workers). Constraints (4) ensure that vehicles arriving at a node ultimately exit the node. Constraints (5) impose that a motorized worker leaving $i \in J$ by walking must formerly arrive at this node either by walking or by car (see Figure 2 (a, b)). Constraints (6) forbid a motorized worker from arriving both by car and by walking at $i \in J$. Constraints (7) state that a motorized worker walking to $i \in J^+$ has her/his car waiting for her/him at i (see Figure 2 (c)). Constraints (8) and (9) forbid a worker (motorized or non-motorized) from using two different arcs simultaneously. Constraints (10) state that every motorized worker who leaves the depot has to come back to it in a single trip.

Specific constraints for non-motorized workers:

$$\sum_{k\in K}\sum_{(j,i)\in A_1} y_{ji}^{kl} + \sum_{(j,i)\in A_2} z_{ji}^{l} = \sum_{k\in K}\sum_{(i,j)\in A_1} y_{ij}^{kl} + \sum_{(i,j)\in A_2} z_{ij}^{l}, \qquad i \in J,\, l \in L \tag{11}$$

$$\sum_{k\in K}\sum_{(i,j)\in A_1} y_{ij}^{kl} + \sum_{(i,j)\in A_2} z_{ij}^{l} \leq 1, \qquad i \in J,\, l \in L \tag{12}$$

$$\sum_{k\in K}\sum_{(j,i)\in A_1} y_{ji}^{kl} + \sum_{(j,i)\in A_2} z_{ji}^{l} = \sum_{k\in J}\sum_{(i,j)\in A_1} y_{ij}^{kl}, \qquad i \in J^{+},\, l \in L \tag{13}$$

$$\sum_{k\in K}\sum_{i\in V} y_{0i}^{kl} = \sum_{k\in K}\sum_{i\in V} y_{i,2n+1}^{kl}, \qquad l \in L \tag{14}$$

$$\sum_{(j,i)\in A_2} z_{ji}^{w} \leq \sum_{(i,j)\in A_2} z_{ij}^{w}, \qquad i \in J,\, w \in W \tag{15}$$

$$\sum_{l\in L} y_{ij}^{kl} \leq c \cdot x_{ij}^{k}, \qquad (i,j) \in A_1,\, k \in K \tag{16}$$

$$\sum_{(j,i)\in A_1} y_{ji}^{kl} \leq 1, \qquad i \in V,\, l \in L,\, k \in K \tag{17}$$

Constraints (11) ensure that a non-motorized worker arriving at node $i \in J$ (either by walking or by car) ultimately exits the node. Constraints (12) state that a non-motorized worker cannot use the two different transportation modes (i.e., walking and driving) to leave a node. Constraints (13) force any non-motorized worker arriving at a node $i \in J^{+}$ to exit the node by car. Constraints (14) state that every non-motorized worker who leaves the depot has to come back to it in a single trip. Constraints (15) ensure that a worker arriving by walking at $i \in J$ exits by walking. Constraints (16) couple non-motorized worker transportation and motorized worker routes. Such constraints are also considered in the APVRP (see (Meisel and Kopfer 2014)). Constraints (17) forbid a worker from using two different arcs arriving at the same node.

Time constraints:

$$t_j^{l} \geq t_i^{k} + \tau_{ij} - M_1 \cdot (1 - y_{ij}^{kl}), \qquad (i,j) \in A_1,\, k \in K,\, l \in L \tag{18}$$

$$t_j^{k} \geq t_i^{k} + \tau_{ij} - M_1 \cdot (1 - x_{ij}^{k}), \qquad (i,j) \in A_1,\, k \in K \tag{19}$$

$$t_j^{w} \geq s_i + p_i + \tilde{\tau_{ij}} - M_2 \cdot (1 - z_{ij}^{w}), \qquad (i,j) \in A_2,\, w \in W \tag{20}$$

$$s_i \geq t_i^{w} - M_3 \cdot (1 - \sum_{(i,j)\in A_2} z_{ij}^{w}), \qquad i \in J,\, w \in W \tag{21}$$

$$t_i^{k} \geq t_i^{l} - M_3 \cdot (1 - \sum_{(i,j)\in A_1} y_{ij}^{kl}), \qquad i \in J^{+},\, k \in K,\, l \in L \tag{22}$$

$$l_i \geq s_i \geq e_i, \qquad i \in J \tag{23}$$

$$l_0 \geq t_0^{w} \geq e_0, \qquad w \in W \tag{24}$$

For each non-motorized worker $l \in L$, constraints (18) set the arrival time at $j \in V$ after being transported by motorized worker $k \in K$ using arc $(i, j) \in A_1$. Constraints (19) define the arrival time at $j \in V$ of motorized worker $k \in K$ using arc $(i, j) \in A_1$. Constraints (20) set the arrival time at $j \in J \cup J^+$ for worker $w \in W$ after processing job $i \in J$ and walking thereafter. Constraints (21) impose that the service time of job $i \in J$ takes place after the arrival time of the worker at that node. Constraints (22) impose that a motorized worker can only leave node $i \in J^+$ if all the non-motorized workers to be transported by her/him have arrived at the node. Constraints (23) impose that the service time of each job must belong to the associated time window. Constraints (24) impose that all workers leave and come back to the depot within the regulatory hours.

4. Methodology

This section starts by describing the general principles of the proposed *Variable Neighborhood Search* (VNS) and the reasons why it is expected to provide good results for the VRPTR. Next, we present a dedicated insertion heuristic that helps to specifically manage walking and carpooling. Finally, after highlighting the complexity associated with searching for the best insertion position for a job in a VRPTR solution, we introduce an algorithm to speed up this procedure, which is then employed as a key procedure of our VNS.

4.1. VNS: motivation and general principles

To tackle the considered problem, we propose a VNS (Mladenović and Hansen 1997) that combines a large neighborhood structure $\mathcal{N}_q$ (it first removes $q > 1$ jobs from the solution and then reinserts them sequentially) and a local search (LS). On the one hand, the role of the LNS component is to diversify the search (i.e., explore new parts of the solution space). For this purpose, it is mandatory to consider large neighborhoods to tackle the VRPTR, as the presence of WRs is responsible for trapping the search in local minima. More precisely, a WR synchronizes multiple workers (i.e., the one that is dropped, the driver that brings her/him at the beginning of the WR, and the driver that picks her/him up at the end of it). Unless all the jobs composing such a WR are removed from the solution, the removed jobs tend to be reinserted at the same position in the same WR. The exploration capability of $\mathcal{N}_q$ would thus be poor for small values of q. On the other hand, and in contrast with the LNS component of VNS, the role of LS is to intensify the search in promising regions of the solution space. For this purpose, small (in terms of the modification of the solution

structure) but efficient (i.e., it should be able to favorably modify the solution value) moves should be performed iteratively on the incumbent solution. Unsurprisingly, Derigs et al. (2013) (for the TTRP) and Meisel and Kopfer (2014) (for the APVRP) have shown that combining a LNS and a LS outperforms the use of a LNS only. We have confirmed this observation by performing preliminary experiments on our instances.

A generic version of the VNS is given in Algorithm 1. It starts from an initial solution s and considers a collection of neighborhood structures $\mathcal{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_{q_{max}}\}$ that are ranked according to their strength for modifying a solution (i.e., $\mathcal{N}_q$ modifies the structure of the involved solution more than $\mathcal{N}_{q-1}$). Section 4.2 details the dedicated algorithm that is used to explore the neighborhood $\mathcal{N}_q$.

The implemented LS is a descent algorithm based on *relocate* moves. Formally, at each step, a job is removed from the solution and reinserted at the best possible location, with or without involving additional walking. As long as the generated neighbor solution $s^{neighbor}$ outperforms the current solution s, the new current solution immediately becomes $s^{neighbor}$. The process stops when the current solution cannot be improved further (i.e., all jobs of s have been tested). More refined LS algorithms (e.g., a tabu search for which it is forbidden to insert a job in some positions) were tested, but they did not yield better results.

In order to facilitate the exploration of the solution space associated with the VRPTR, constraints (2), which guarantee that all jobs are visited, are relaxed (i.e., removed from the constraints and penalized in the objective function with a penalty parameter ψ). Following this, for each solution $s = \{x, y, z\}$, $d(s) = \sum_{k \in K} \sum_{(i,j) \in A_1} d_{ij} \cdot x_{ij}^k$ is the overall driving distance. $J^{in}(s) = \{j \in J \mid \sum_{(j,i) \in A_2} z_{ji} = 1\}$ is the set of jobs that are served in s, and $J^{out}(s) = J \backslash J^{in}(s)$ is the set of unserved jobs in s. Objective (1) thus becomes as shown below, where ψ is chosen sufficiently large to ensure that for the two solutions s and s', if $|J^{out}(s)| < |J^{out}(s')|$, then $c(s) < c(s')$, with:

$$c(s) = d(s) + \psi \cdot \mid J^{out}(s) \mid \tag{25}$$

4.2. VNS: shaking phase

Neighborhood $\mathcal{N}_q$ is explored by means of an LNS-type procedure, based on the sequential use of a *removal* heuristic (Section 4.2.1) and an *insertion* heuristic (Section 4.2.2).

Algorithm 1 Variable Neighborhood Search (VNS)

Input: n_L, q_{max}, K, W.

Generate an initial solution s with $|K|$ vehicles and $|W|$ workers.
Set $q = 1$.

While no stopping condition is met, **do**

(1) *Shaking:* randomly generate n_L (parameter) solutions in $\mathcal{N}_q(s)$, and let s' be the best of these solutions.
(2) *Local search (LS):* apply the local search on s', and let s'' be the resulting solution.
(3) *Move or not:* if s'' is better than s, move there (i.e., set $s = s''$), and continue the search with $\mathcal{N}_1$ (i.e., set $q = 1$); otherwise set $q = q + 1$, but if $q > q_{max}$, set $q = 1$ (i.e., start a new research cycle).

4.2.1. Removal heuristic

The removal heuristic aims at dropping jobs that currently block the search process in a local minimum. We consider here the *related removal heuristic* (RRH) proposed by Shaw (1998) and adapt it for the VRPTR. The general idea is that it is likely to be easier to reinsert removed jobs that share some similarities. The relatedness function $R(i, j)$ indicates how two jobs i and j are similar. To that aim, some parameters are introduced. $(\alpha, \beta, \gamma, \delta, \epsilon)$ are positive weights, and $\text{WR}(i, j) = 1$ if i and j are served in the same WR; $\text{WR}(i, j) = 0$ otherwise. $\mathbb{1}_{k_i=k_j} = 1$ if i and j are served in the same route; $\mathbb{1}_{k_i=k_j} = 0$ otherwise. This relatedness function takes into account the geographical proximity $(\alpha \cdot d_{ij})$, the similarity in service time $(\beta \cdot |h_i - h_j|)$, the similarity in time windows $(\gamma \cdot |l_i - l_j|)$, and the presence in common WRs $(\delta \cdot (1 - \text{WR}(i, j)))$ and common routes $(\epsilon \cdot (1 - \mathbb{1}_{k_i=k_j}))$. The smaller $R(i, j)$ is, the greater i and j are related:

$$R(i, j) = \alpha \cdot d_{ij} + \beta \cdot |h_i - h_j| + \gamma \cdot |l_i - l_j| + \delta \cdot (1 - \text{WR}(i, j)) + \epsilon \cdot (1 - \mathbb{1}_{k_i=k_j}) \tag{26}$$

The first removed job is randomly selected in $J^{in}(s)$. Then, L_{NR} designates the ranked list of non-removed jobs. The relatedness of a non-removed job i is computed according to one of the removed jobs j (that is randomly selected, as proposed in other studies considering related removal, e.g., (Ropke and Pisinger 2006)). Next, as long as q removals have not been performed, the job $L_{NR}[\lfloor y^\rho \cdot |J^{in}(s)| \rfloor]$ is removed from the solution. y is randomly generated in $[0, 1]$, and $\rho \in [0, 1]$ is a parameter that calibrates the degree of randomness of the removal heuristic ($\rho = 1$, jobs are randomly removed; $\rho = 0$, the most related job is removed at each step).

4.2.2. Insertion heuristic

Before introducing the proposed insertion heuristic, we start by describing the drawbacks that best-insertion heuristics (BIHs), which is one of the most frequently used insertion components of the LNS (e.g., (Ropke and Pisinger 2006, Masson et al. 2014, Grangier et al. 2016)), have in the present situation. The BIH inserts first the job that minimizes the insertion cost (i.e., the additional driving distance in the present case). The BIH is inefficient in the VRPTR context because it either favors (a) insertions involving walking only or (b) insertions in a driver's planning. For (a), it follows from the fact that walking does not increase the driving distance. For (b), moving a driver to a job only requires one detour with her/his assigned car, whereas assigning this job to a passenger requires two detours: one for the drop-off and one for the pick-up. First, these drawbacks limit the diversification ability of the insertion heuristic. Second, unbalanced schedules are created for the workers because more jobs are assigned to drivers than to passengers, which finally results in assigning the latest considered jobs to the passengers (which are the most difficult resources to move). For all these reasons, we propose below an insertion heuristic capable of removing these two drawbacks due to carpooling and walking.

We propose a *Random Worker Best-Insertion* (RWBI) heuristic, the pseudo-code of which is given in Algorithm 2. At each step of the RWBI, a worker is first randomly chosen, and then a non-dominated insertion is performed. An insertion is said to be non-dominated if no other insertion has a better performance, according to both the walking distance and the insertion cost. Randomly selecting the worker that will serve the next job helps in overcoming the problem of over-insertions in drivers' planning. Furthermore, choosing a non-dominated insertion position allows for efficiently managing the amount of walking time in the solution. On the one hand, walking seems favorable, as it does not contribute cost-wise to the objective function. Additionally, the more the passengers are walking, the less the drivers are used for transporting the passengers, and the more time they can allocate to perform jobs themselves. However, on the other hand, walking directly reduces the workers' availability and thus augments the likelihood of having unserved jobs that will remain at the end of the shaking phase. During the RWBI, the jobs are inserted without walking in the drivers' routes. This maximizes the likelihood of getting a feasible and non-saturated solution at the end of the diversification step. Walking is added to the drivers' planning during the intensification step (i.e., LS).

Algorithm 2 Random Worker Best Insertion heuristic (RWBI)

Input: s

Set $J^{out}(s)$ as the set of unserved jobs in s.

While $J^{out}(s) \neq \emptyset$, **do**

(1) Compute the set W^{av} of available workers.
(2) Select a worker $w \in W^{av}$ randomly.
(3) Randomly choose a non-dominated insertion from J^{out} in w.

If no feasible insertion is found, *change* w **or** stop RWBI if all workers have been tested.
Else: proceed the insertion **and** update $J^{out}(s)$.

4.3. Complexity of an insertion

The best-insertion move is the core component of both RWBI and LS. This section shows that finding the best insertion position for a job in a solution involving carpooling requires $\mathcal{O}(n^5)$ feasibility tests. For the VRP, it only requires $\mathcal{O}(n)$ of such tests.

All the insertion positions are greedily tested to find the cheapest one. We consider the insertion of job $j \in J^{out}(s)$ to a non-motorized worker route (n jobs are inserted in the solution). In this case, we therefore need to transport this worker from its previous WR to j, and from j to its next WR (after the job has been processed). This leads to the creation of two new transportation requests (each one composed of a pick-up and a delivery) that have to be inserted in the driver routes. As a result, four nodes must be inserted into the solution, and accordingly, the number of feasibility tests is in $\mathcal{O}(n^4)$. The number of insertion positions (between two WRs) for this non-motorized worker is proportional to the number of jobs in the solution, and therefore the total number of required tests grows to $\mathcal{O}(n^5)$. Note that removing a job from the route of a non-motorized worker is also a complex task, as a transportation request from her/his previous WR to her/his next WR has to be created.

4.4. Fastening the insertion phase

In addition to the significantly larger number of tests to be performed, proving the feasibility of an insertion is a more complex task for the VRPTR than for the VRP. First, we need to check that the induced delay after the insertion does not violate future time windows. As all routes can be interconnected, it is not sufficient to only recompute the concerned route; rather the whole solution may have to be updated. Second, when assigning a job to a non-motorized worker, four nodes must

be inserted into the solution (see Section 4.3). Therefore, the solution must be correctly updated four times (once after each node insertion) when checking the overall feasibility of the insertion.

To avoid having to recompute the whole solution after each of the four insertions, we propose a fast insertion algorithm based on a precedence graph that captures all the temporal constraints. Masson et al. (2014) introduced the same type of graph structure for a vehicle routing problem in which transfers are allowed to transport persons. Each physical node in the real network has its associated node in the precedence graph. For each node v in this graph, it is possible to compute the earliest arrival time h_v and the latest departure time λ_v (i.e., leaving v after this time would lead to a violation of a future time window) and the waiting time matrix Φ between all pair of nodes. For a driver, waiting at a node is either due to an early arrival before the start of a time window or to the later arrival of a non-motorized worker that needs to be transported further. The computation of these values is done in $\mathcal{O}(n^2)$. Appendix A shows the construction of the precedence graph and provides details on the computation of h_v, λ_v and Φ (Cherkassky et al. 2009).

$i_1 + 1$ denotes the successor node of i_1 in route k_1. e_{D_1} (resp. l_{D_1}) is the earliest start time of the WR starting at node D_1 (resp. latest arrival time at the WR starting at D_1 to serve all jobs of the WR on time). p_{D_1} is the processing time of the WR starting at D_1. After proceeding with an insertion in the graph, for node v in the precedence graph, $\bar{h}_v$ is the new arrival time and $\delta_v = \max\{\bar{h}_v - h_v, 0\}$ denotes the delay induced in v.

Based on these notations, Algorithm 3 tests in constant time whether assigning a job to a non-motorized worker at a given position is feasible. More precisely, the proposed algorithm contains a specific feasibility check that corresponds to the precedence constraints arising between the WRs performed by a same worker. Algorithm 3 details the most complex situation when four nodes (namely (P_1, D_1) and (P_2, D_2), corresponding to the two new transportation requests created for the non-motorized worker) are inserted into positions (i_1, j_1) (resp. (i_2, j_2)) in route k_1 (resp. k_2). After inserting any of the four nodes, the induced delay at any other node is computed in constant time. For this purpose, the delay after each insertion is reduced by the smallest waiting time between the predecessor and successor nodes. If the delay does not exceed the latest departure time of the successor node, all other jobs will still be served within their time window, and the four nodes remaining to be inserted are tested. If this is not so, the insertion position is determined to be unfeasible. Experiments have shown that using this fast feasibility check procedure can reduce the computation time of the proposed VNS by 95%.

Algorithm 3 Algorithm for testing the feasibility of the insertion of (P_1, D_1) and (P_2, D_2) after i_1, j_1, i_2, j_2, respectively.

Evaluate the insertion of P_1:

- **set** $\bar{h}_{P_1} = \max\{e_{P_1}, h_{i_1} + \tau_{i_1,P_1}\}$ **, if** $\bar{h}_{P_1} > l_{P_1}$ **return FALSE**.
- **set** $\bar{h}_{i_1+1} = h_{P_1} + \tau^d_{P_1,i_1+1}$ **, if** $\bar{h}_{i_1+1} > \lambda_{i_1+1}$ **return FALSE**.

Evaluate the insertion of D_1:

- **set** $\bar{h}_{j_1} = h_{j_1} + \max\{\delta_{i_1+1} - ST_{i_1+1,j_1}, 0\}$, **set** $\bar{h}_{D_1} = \max\{h_{j_1} + \tau_{j_1,D_1}, e_{D_1}\}$, **, if** $\bar{h}_{D_1} > l_{D_1}$ **return FALSE**.
- **set** $\bar{h}_{j_1+1} = \bar{h}_{D_1} + \tau_{D_1,j_1+1}$ **, if** $\bar{h}_{j_1+1} > \lambda_{j_1+1}$ **return FALSE**.

Evaluate the insertion of P_2:

- **set** $\bar{h}_{i_2} = h_{i_2} + \max\{\delta_{i_1+1} - ST_{i_1+1,i_2}, \delta_{j_1+1} - ST_{j_1+1,i_2}, 0\}$ and **set** $\bar{h}_{P_2} = \max\{h_{i_2} + \tau_{i2,P_2}, h_{D_1} + p_{D_1}\}$.
- **set** $\bar{h}_{\sigma(i_2)} = h_{P_2} + \tau_{P_1,i_2+1}$ **, if** $\bar{h}_{\sigma(i_2)} > \lambda_{\sigma(i_2)}$ **return FALSE**.

Evaluate the insertion of D_2:

- **set** $\bar{h}_{j_2} = h_{j_2} + \max\{\delta_{i_1+1} - ST_{i_1+1,j_2}, \delta_{j_1+1} - ST_{j_1+1,j_2}, \delta_{i_2+1} - ST_{i_2+1,j_2}, 0\}$, **set** $\bar{h}_{D_2} = \bar{h}_{j_2} + \tau^d_{j_2,D_2}$, **if** $\bar{h}_{D_2} > l_{D_2}$ **return FALSE**.
- **set** P_3 as the pick-up at the end of WR D_2, **if** $\max\{\bar{h}_{D_2}, e_{D_2}\} + p_{D_2} > \lambda_{P_3}$ **return FALSE**.

return TRUE.

5. Computational experiments

We start by describing the considered set of benchmark instances in Section 5.1. Section 5.2 introduces some notation needed to present the numerical experiments as well as the considered routing configurations. Section 5.3 presents the results of the MILP, whereas Section 5.4 analyzes the performance of the proposed VNS. Finally, Section 5.5 gives the results of the VNS for the introduced set of instances.

The MILP and the VNS have been coded in C++. The MILP is solved with CPLEX 12.4 (called with the Concert Technology). Computations were performed on a 2.2 GHz Intel Core i7 with 16 Go 1600 MHz DDR3 RAM. In Algorithm 1, the parameters q_{max} and n_L were tuned to 30% and 10 respectively. In preliminary experiments, values were tested in $[10\%, 50\%]$ for q_{max} and in $[1, 15]$ for n_L.

5.1. Instances

The VRPTR is a new problem proposed by company EEP for which no benchmark instance exists in the literature. Focusing on urban contexts, a set of instances has been generated according to the real parameter distributions provided by EEP. The job locations are uniformly distributed in a square grid of 10 km by 10 km. The Euclidean metric is used to compute the distance between the jobs. As highlighted by Boysen et al. (2018), using Euclidean distances ensures that the triangle inequality is satisfied for both walking and driving. The driving speed is 30 km/h and the walking speed is 4 km/h. The job duration ranges between 20 and 35 minutes (uniformly distributed). The maximum walking time τ_M^f to reach a job on foot is 15 minutes (i.e., 1 km), and the maximum walking distance d_M^f per day and per worker is 8 km (i.e., 2 hours). The duration of the working day is 7 hours, from 8 am to 3 pm. The depot is located at the center of the considered urban area.

Instances with $n \in \{20, 30, 40, 50\}$ are considered. Such instance sizes allow for comparing our results with VRP optimal solutions and are in line with the existing literature considering the en route synchronization of transportable resources (e.g., Boysen et al. (2018) solve real-world instances for up to 40 customers). In the present study, the same distance matrix is used for both walking and driving. This yields the obtained results to be a lower bound on the ones that would be obtained when walking implies shorter distances than driving (e.g., when vehicles would be constrained by one-way streets or in the presence of pedestrian walkways). Although other distance matrices could be alternatively considered, preliminary experiments have shown that similar results are found when the Manhattan metric is used, but for a slightly reduced grid size compared to the Euclidean case considered here. While the Manhattan distance could decrease the walking potential of the instances (as the distances between the jobs would increase by a factor lying in $[1, \sqrt{2}]$), the walking potential (as introduced in Section 6.2.1) of the considered instances is in line with EEP's field observations when using the Euclidean metric.

Three service levels are envisioned by EEP. The smaller the time window, the shorter the mandatory availability for the involved client and, hence, the better the service level. Three types of time window are considered: *all day* (i.e., each job can be served in the [8 am, 3 pm] time window), *half day* (i.e., each job is associated to either the [8 am, 11:30 am] or the [11:30 am, 3 pm] time windows), and *quarter day* (i.e., each job is associated with one of the following time windows: [8 am, 9:45 am], [9:45 am, 11:30 am], [11:30 am, 1:15 pm], [1:15 pm, 3 pm]). We consider three types of instances, each of them representing one single envisioned service level. The time window

assigned with each job is uniformly chosen among the possible alternatives, describing the clients' preferences.

An instance is referred to as "n_TW_i", where n stands for the number of jobs, TW represents the size of the used time window (A, H, and Q correspond, respectively, to *all day*, *half day*, and *quarter day*), i characterizes the instance identifier, and n_TW denotes the set of all instances of size n and time window size TW. 10 instances have been generated for each n and each TW, leading to a total of 120 instances.

5.2. Notation and considered configurations

All of the 120 instances have been solved to optimality for the VRP configuration. For this purpose, we have used the algorithm proposed by Desaulniers et al. (2008), which is acknowledged to be one of the most efficient algorithms for solving the VRP (Baldacci et al. 2012). By assigning appropriate weights to the number of workers used in the solution, Desaulniers et al. (2008) first minimize the number of employed workers and then minimize the total traveled distance. Accordingly, we know the associated smallest number of workers $|W^\star|$ required to serve all jobs. In the following, we consider different configurations $\left(P_a^{|K|}\right)$, where $|K|$ designates the number of used vehicles and $a \in \{walk, no\ walk\}$ indicates whether or not walking is allowed. $d\left(P_a^{|K|}\right)$ (resp. $d^\star\left(P_a^{|K|}\right)$) gives the total driving distance for configuration $(P_a^{|K|})$ found by VNS (resp. the total driving distance for the optimal solution). All configurations are solved with $|W^\star|$ workers (fixed by the VRP optimal solution).

The following five $\left(P_a^{|K|}\right)$ are considered:

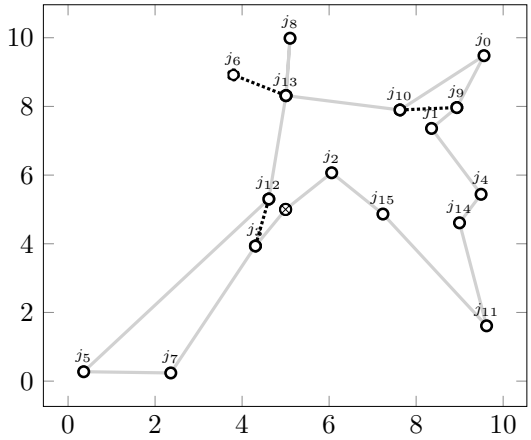
- $\left(P_{no\ walk}^{|W^\star|}\right)$: all workers are motorized, but they are not allowed to walk (i.e., VRP);
- $\left(P_{walk}^{|W^\star|}\right)$: all workers are motorized, and they are allowed to walk (i.e., *Park-and-Loop*);
- $\left(P_{no\ walk}^{|W^\star|-1}\right)$: carpooling is allowed, one worker is not motorized, but walking is forbidden;
- $\left(P_{walk}^{|W^\star|-1}\right)$: both carpooling and walking are allowed, and one worker is not motorized;
- $\left(P_{walk}^{|W^\star|-2}\right)$: both carpooling and walking are allowed, and two workers are not motorized.

For the considered instances and when walking was not permitted, it was never possible to remove more than one car with respect to the optimal VRP solution. When walking was allowed, it was

never possible to remove more than two cars.

5.3. MILP results for the VRPTR

As already mentioned, a time limit of 10 hours is used. When only walking is allowed (but no carpooling), the MILP can find solutions for instances involving 20 jobs. However, when both walking and carpooling are considered, the MILP can only find solutions for instances with up to $n = 18$ jobs. A solution obtained by the MILP is shown in Figure 4. It exhibits both carpooling and walking, and points out the efficient synchronization that arises between a driver and a passenger. Considering the instances for which the MILP can be used to find optimal solutions, the proposed VNS finds results with a percentage gap never exceeding 1%.



Two workers (the driver w_0 and the passenger w_1) and one vehicle are required to visit the $n = 16$ jobs. Dashed (resp. plain) lines represent walking (resp. driving); light gray (resp. black) lines represent the movement of w_0 (resp. w_1). w_0 and w_1 initially move from the central depot to job j_3, where w_1 is dropped off. After serving j_3, w_1 walks to j_{12} and serves it. Meanwhile, using the car, w_0 serves j_7 and j_5 before picking up w_1 at j_{12}. w_1 is then dropped off at j_{13}, serves it, and walks to j_6 before coming back to j_{13}. During this period of time, w_0 uses the car to serve j_8, and then comes back to pick up w_1. Both workers then move together to j_{10}, where w_1 is dropped off. Then, the tour continues with the same logic.

Figure 4: An optimal solution to the VRPTR, with both carpooling and walking.

5.4. Performance of VNS on the VRP configuration

Focusing on configuration $P^{|W^\star|}_{no\ walk}$ (i.e., VRP), for all generated instance types (number of jobs and time window sizes), Table 1 gives the average percentage of unserved jobs in the solutions found by the VNS (column "% unserved."), the average percentage gap of VNS with respect to the optimal values (column "% gap*") and the percentage of instances that could be solved to optimality

(column "% opt"). "% gap*" is computed as follows: $\frac{d\left(P^{|W^\star|}_{no\ walk}\right)-d^\star\left(P^{|W^\star|}_{no\ walk}\right)}{d^\star\left(P^{|W^\star|}_{no\ walk}\right)} \cdot 100$. Columns "All Day", "Half Day", and "Quarter Day" refer to the size of the considered time window, and each line considers the 10 instances for a given value of n. Table 1 shows that the VNS finds optimal solutions for 95% of the instances. For the remaining 5%, either the VNS did not find a feasible solution (i.e., some jobs remain unserved) with a number of vehicles fixed at its optimal value (see column "% unserved"), or a small gap is observed with respect to the optimal driving distance (see column "% gap*"). Although the proposed VNS has been specifically designed for the VRPTR, these results contribute to validating its efficiency and consistency.

Table 1: Performance of VNS on configuration $P^{|W^\star|}_{no\ walk}$.

Time Window Size	All Day			Half Day			Quarter Day		
n	% unserved	% gap*	% opt	% unserved	% gap*	% opt	% unserved	% gap*	% opt
20	0%	0%	100%	0%	0%	100%	0%	0%	100%
30	0%	0%	100%	0%	0%	100%	0.3%	0%	90%
40	0%	0%	100%	0%	0.04%	90%	0.2%	0.08%	90%
50	0%	0%	100%	0%	1.34%	90%	0.4%	0.82%	80%

5.5. VNS results for the VRPTR

5.5.1. Proportion of feasible instances for configurations involving less cars than workers.

Contrary to the *Park-and-Loop* configuration for which a VRP solution can be initially built and then improved through the introduction of walking sub-tours, the configurations involving carpooling (i.e., less cars than workers: $\left(P^{|W^\star|-1}_{walk}\right)$, $\left(P^{|W^\star|-1}_{no\ walk}\right)$, and $\left(P^{|W^\star|-2}_{walk}\right)$) are structurally more complex. Finding a feasible solution cannot be taken for granted. Indeed, both walking (slower than driving) and carpooling (need for detours to drop off and pick up non-motorized workers) involve potential inefficiencies, and it is therefore not surprising that some instances end up unfeasible when some workers are non-motorized. While Fikar and Hirsch (2015) generate solutions involving less cars than workers, it comes at the price of increasing the total number of employed workers (compared with the VRP optimal solution). Here, we keep this total number of workers stable.

Figure 5 qualitatively highlights the potential values associated with the feasible solutions of the configurations involving less cars than workers. More precisely, three situations might arise, ranging from an improvement with respect to the *Park-and-Loop* configuration, an amelioration of the

VRP solution, to not improving the one-man-one-car models (VRP and *Park-and-Loop*). Indeed, reducing the driving distance poses an additional challenge since detours to transport non-motorized workers must be efficiently compensated by merging the right paths with carpooling.

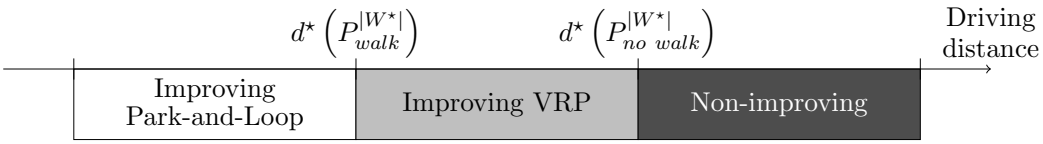


Figure 5: Potential driving distance of the solutions found feasible with less cars than workers.

For the three configurations involving carpooling, Table 2 gives the number of instances (and the associated percentage), over the 120 considered instances, that belong to each category identified in Figure 5. It indicates that for 55% of the instances, the VNS is able to reduce the fleet size when only carpooling is allowed (configuration $\left(P_{no\ walk}^{|W^\star|-1}\right)$). Moreover, the VNS finds smaller total driving distances than in the VRP solution for 6.7% of the instances, meaning that the need for detours generated by carpooling can be overcompensated by efficiently merging worker journeys. When walking is permitted in addition to the implementation of carpooling (configuration $\left(P_{walk}^{|W^\star|-1}\right)$), the number of feasible instances found by the VNS grows from 55% to 56.7%, the number of instances for which VRPTR dominates VRP increases from 6.7% to 19.2%, and the proportion of instances for which VRPTR is able to improve the *Park-and-Loop* configuration $\left(P_{walk}^{|W^\star|}\right)$ increases from 0% to 6.7%. When two cars are removed with respect to the VRP optimal solution, the VNS finds feasible solution for 7.5% of the instances. This relatively low number can be explained by the fact that the generated instances require a maximum of 5 workers (with an average of 3.3 workers) to be solved. Thus, removing two cars represents a drastic reduction of the vehicle fleet.

Table 2: Proportion of feasible instances for the different configurations involving less cars than workers.

Solution characteristics	$\left(P_{no\ walk}^{\lvert W^\star\rvert-1}\right)$		$\left(P_{walk}^{\lvert W^\star\rvert-1}\right)$		$\left(P_{walk}^{\lvert W^\star\rvert-2}\right)$	
	% Inst.	Nb. Inst.	% Inst.	Nb. Inst.	% Inst.	Nb. Inst.
Improving *Park-and-Loop*	0.0%	(0 / 120)	6.7%	(8 / 120)	0.0%	(0 / 120)
Improving VRP	6.7%	(8 / 120)	19.2%	(23 / 120)	0.0%	(0 / 120)
Non-improving	48.3%	(58 / 120)	30.8%	(37 / 120)	7.5%	(7 / 120)
Total feasible	55.0%	(66 / 120)	56.7%	(68 / 120)	7.5%	(7 / 120)

5.5.2. Detailed results

Appendix B details the results found by VNS for all instances and all configurations. An extract of three representative instances (corresponding to the three boxes displayed in Figure 5) is given in

Table 3. The "VRP" columns reflect the characteristics of the optimal VRP solutions: the number of workers required ($|W^\star|$), the total driving distance ($d^\star$), and the corresponding idle time (either en route or at the depot). The "Park-and-Loop" column gives the total driving distance found for configuration $\left(P_{walk}^{|W^\star|}\right)$. The "Carpooling" columns denote the associated driving distance (d) and the number of jobs that cannot be served in the solution ($|J^{out}|$) for all configurations involving carpooling (n.b., the driving distance is not displayed for unfeasible solutions). Focusing on configuration $\left(P_{walk}^{|W|^\star-1}\right)$, it shows that for instance 40_H_2, it improves the *Park-and-Loop* solution (driving distance reduced by 2.8%), which itself already improves the optimal VRP solution. The solution of instance 50_A_6 improves the VRP optimal solution (driving distance reduced by 1.8%) but exhibits a driving distance 7.2% greater than in the *Park-and-Loop* solution. Instance 50_Q_5 is found feasible, but its solution returns a driving distance larger than in the optimal VRP solution.

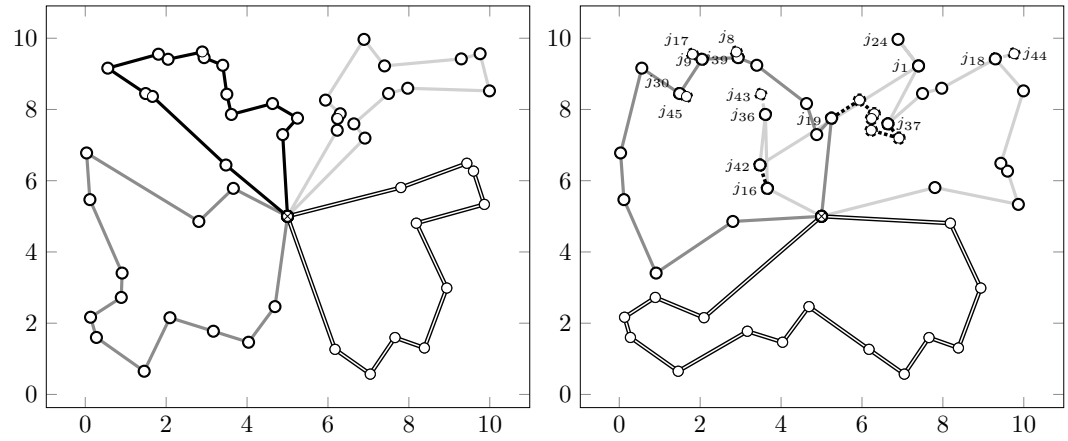
Table 3: Detailed results for the representative instances.

Instance	VRP $\left(P_{no\ walk}^{|W^\star|}\right)$				Park-and-Loop $\left(P_{walk}^{|W^\star|}\right)$	Carpooling					
			Idle Time			$\left(P_{no\ walk}^{|W^\star|-1}\right)$		$\left(P_{walk}^{|W^\star|-1}\right)$		$\left(P_{walk}^{|W^\star|-2}\right)$	
	$\|W^\star\|$	$d^\star$	Route	Depot	d	d	$\|J^{out}\|$	d	$\|J^{out}\|$	d	$\|J^{out}\|$
40_H_2	4	73.1	0.0%	25.8%	63.9	75.1	0	62.1	0	94.6	0
50_A_6	4	71.8	0.0%	11.7%	66.1	78.7	0	70.9	0	-	4
50_Q_5	5	113.2	17.9%	2.7%	104.4	130.7	0	117.0	0	-	2

Figure 6 exemplifies a VRPTR solution (right part of the figure) for configuration $\left(P_{walk}^{|W^\star|-1}\right)$ in instance 50_A_6 and compares it with the optimal VRP solution (left part of the figure). In this example, carpooling and walking allow for improving the optimal VRP solution, as for the same number of employed workers, the driving distance is reduced by 1.3% and one car is saved. In this VRPTR solution, the non-motorized worker walks for 71 minutes. Note that for all performed experiments, walking never exceeds 90 minutes per worker.

5.6. Execution time

The stopping criterion of the VNS has been set to 10 hours (i.e., one night of computation, from 8 pm to 6 am), which follows EEP's requirements in the present one-day-ahead optimization context. Table 4 gives the average execution times (for each n and TW values) to obtain the best found solutions. It highlights the fact that solving ($P_{walk}^{|W^\star|-1}$) is more complex than solving ($P_{walk}^{|W^\star|}$). Indeed, the additional complexity of finding the best insertion position for a job when carpooling



(a) VRP optimal solution of $\left(P^{|W^\star|}_{no\ walk}\right)$ (b) VRPTR solution of $\left(P^{|W^\star|-1}_{walk}\right)$

Plain (resp. dashed) lines represent the vehicle paths (resp. the walking routes). Each line type corresponds to a worker: double line for worker w_1, light gray for w_2, gray for w_3, and black for w_4 (non-motorized worker in (b)). The jobs explicitly labeled are those included in a WR. In (b), w_4 is dropped off at j_{16} and walks to j_{42}. w_2 fulfills j_{36} and j_{43} in a WR before picking up w_4 at j_{42}. Before being dropped off at j_{37}, w_4 works on j_1 while w_2 serves j_{24}. The tours continue as described, and w_4 returns to the depot with w_3.

Figure 6: Illustration of a VRPTR solution for which one car is saved and the total driving distance is reduced by 1.3%. The optimal VRP solution and a VRPTR solution are presented for the same instance.

is allowed (as shown in Section 4.3) is reflected in these execution times. Interestingly, Table 4 furthermore indicates that the time window size only has a marginal impact on the execution time.

Table 4: Average execution time of the VNS (in seconds) for each instance and time window size.

	$(P^{\|W^\star\|}_{walk})$			$(P^{\|W^\star\|-1}_{walk})$		
n	All day	Half day	Quarter day	All day	Half day	Quarter day
20	71	62	30	1,133	1,470	611
30	381	860	1,174	4,552	5,498	3,834
40	1,993	1,988	3,747	10,939	10,030	9,825
50	6,779	8,143	11,219	24,454	18,972	21,856

6. Managerial insights

In Section 6.1, we position the obtained VRPTR solutions with respect to the existing industrial practices. Next, in Section 6.2, we highlight the instance characteristics that influence the gain obtained with the VRPTR formulation. Finally, in Section 6.3, we discuss how the obtained static solutions would be expected to react when unforeseen events occur, and we propose some associated research avenues.

6.1. Comparison with existing practices

In Section 6.1.1, we discuss how classical VRP solutions can be improved with the introduction of walking only (i.e., results of the *Park-and-Loop*). Next, in Section 6.1.2, we show how also allowing carpooling competes with respect to the one-man-one-car models (i.e., results of the VRP and of the *Park-and-Loop*).

6.1.1. Benefits of the Park-and-Loop

Table 5 quantifies the aggregated gains provided by the *Park-and-Loop* formulation over the 120 considered instances. Each line corresponds to a specific time window and instance size (i.e., it covers 10 instances). The "VRP" columns characterizes the optimal VRP solutions (idle time, $|W^\star|$ and driving distance). The "Park-and-Loop" columns display the average total driving distance (in km) found by VNS in column "$d\left(P_{walk}^{|W^\star|}\right)$" and the corresponding percentage gap with respect to the optimal VRP solution values in column "% gap" (computed as: $\frac{f\left(P_{no\ walk}^{|W^\star|}\right)-f\left(P_{walk}^{|W^\star|}\right)}{f\left(P_{no\ walk}^{|W^\star|}\right)} \cdot 100$).

Table 5 shows that a significant reduction in the driving distance is achieved when walking is allowed (average gain of 6.4% over the 120 instances). The gain remains stable with the instance size, and no systematic effect can be observed from the time window size. The explanation of such an output will be discussed in Section 6.2.1. This driving distance gain is the consequence of transferring parts of the journeys traveled by car to walking, resulting in an efficient use of the available idle time present in the VRP optimal solutions. This aspect will be further discussed in Section 6.2.1. We also observe from Table 5 that a threshold effect on the achieved gains might appear depending on the number of involved jobs and the considered time window size. This is due to the fact that depending on the considered configuration (n, TW), the worker plannings can be more or less saturated in the optimal VRP solutions. For example, configuration ($n = 40$, $TW = all\ day$) yields optimal VRP solutions that contain little idle time for the involved workers. As explained later in Section 6.2.1, where the instance characteristics associated with efficient VRPTR solutions are discussed, this leads to less potential gain when walking and/or carpooling is implemented.

6.1.2. Benefits of joint walking and carpooling

In their decision-making processes, managers have the choice of favoring one configuration over another to either reduce the driving distance (objective denoted as f_{dist}, in km) or the size of the vehicle fleet (objective denoted as f_{car}). Three different scenarios are envisioned by EEP.

Table 5: Aggregated results for the *Park-and-Loop* configuration $\left(P^{|W^\star|}_{no\ walk}\right)$.

Instance	VRP			Park-and-Loop	
	Idle time	$\|W^\star\|$	$d^\star\left(P^{\|W^\star\|}_{no\ walk}\right)$	$d\left(P^{\|W^\star\|}_{walk}\right)$	%gap
20_A	24.8%	2.0	42.1	38.2	9.2%
20_H	22.5%	2.2	56.3	48.6	13.7%
20_Q	30.2%	2.4	64.7	62.5	3.4%
30_A	26.5%	3.0	53.4	46.8	12.4%
30_H	24.6%	3.0	65.3	59.5	8.9%
30_Q	27.2%	3.3	85.9	82.8	3.7%
40_A	8.4%	3.2	60.8	57.5	5.5%
40_H	18.4%	3.7	76.6	69.5	9.3%
40_Q	22.0%	4.0	98.6	93.2	5.5%
50_A	10.5%	4.0	69.3	63.1	9.0%
50_H	8.3%	4.0	87.8	83.9	4.5%
50_Q	16.8%	4.6	112.4	108.0	3.9%

$\left(S^{(dist)}\right)$ focuses on the minimization of f_{dist} (vehicles are removed as long as the incurred detours are compensated); $\left(S^{(car)}\right)$ targets f_{car} (vehicles are removed as long as all jobs can be served on time); and $\left(S^{(car)}_{(dist^\star)}\right)$ represents the balanced scenario where both f_{dist} and f_{car} are simultaneously considered (vehicles are removed as long as the driving distance is below the driving distance from the optimal VRP solution).

Table 6 presents the results for all of the above-mentioned scenarios and provides a comparison with the one-man-one-car models. The reported values are averaged over all instances sharing the same time window size. Instances are aggregated per time window size in order to avoid any misleading interpretation due to the threshold effect that might appear for some configurations (n, TW) (as observed in Section 6.1.1) and hence obtain a general understanding of the average gain that follows from the introduction of walking and carpooling. The "KPI" (Key Performance Indicator) column indicates the considered value $|K|$ (resp. d) for the size of the vehicle fleet (resp. the total driving distance). The "VRP" and "Park-and-Loop" columns reflect some of the values presented in Table 5. Column "%VRP" (resp. "%P&L") gives the percentage gap with respect to the VRP solution (resp. with the *Park-and-Loop* solution).

Table 6 shows that, while keeping the number of workers stable with respect to the optimal VRP solution, the size of the vehicle fleet can be significantly reduced (scenario $S^{(car)}$). This reduction is up to 23% for instances with *all day* time windows and averages 18.4% for all instances. Carpooling allows for a further improvement of the total driving distance compared with the one-man-one-car models (scenario $S^{(dist)}$). The improvement with respect to the VRP averages 6.5% for all instances,

Table 6: Results for both the vehicle fleet and the total driving distance for all scenarios.

Time Window Size	KPI	VRP	Park-and-Loop		$S^{(dist)}$			$S^{(car)}$			$S^{(car)}_{(dist^\star)}$		
			Value	%VRP	Value	%VRP	%P&L	Value	%VRP	%P&L	Value	%VRP	%P&L
All day	$\|K\|$	3.1	3.1	0.0%	2.9	4.1%	4.1%	2.4	23.0%	23.0%	2.7	13.1%	13.1%
	d	56.4	51.4	8.9%	51.2	9.3%	0.5%	55.3	1.9%	-7.7%	52.0	7.8%	-1.2%
Half day	$\|K\|$	3.2	3.2	0.0%	3.1	1.6%	1.6%	2.6	17.3%	17.3%	2.9	7.1%	7.1%
	d	70.4	65.4	7.2%	65.3	7.3%	0.1%	71.3	-1.2%	-9.1%	66.4	5.8%	-1.5%
Quarter day	$\|K\|$	3.6	3.6	0.0%	3.5	0.7%	0.7%	3.0	15.4%	15.4%	3.4	4.9%	4.9%
	d	90.4	86.6	4.2%	86.6	4.3%	0.0%	92.2	-2.0%	-6.5%	87.0	3.7%	-0.5%
Total	$\|K\|$	3.3	3.3	0.0%	3.2	2.0%	2.0%	2.7	18.4%	18.4%	3.0	8.2%	8.2%
	d	72.4	67.8	6.4%	67.7	6.5%	0.1%	73.0	-0.7%	-7.6%	68.5	5.4%	-1.0%

and it increases to 9.3% for instances with *all day* time windows. A small average gain still exists in terms of total driving distance compared to the *Park-and-Loop*, which shows that the vehicle fleet is more efficiently used. Whereas Table 6 indicates that the presence of *Park-and-Loop* sub-tours is responsible for most of the reduction of the driving distance, scenario $S^{(car)}_{(dist^\star)}$ highlights that with an average increase of 1% of the driving distance with respect to *Park-and-Loop* solutions, savings in the number of used cars are as high as 8%.

From Table 6, we observe that a conflict exists between objectives f_{car} and f_{dist} as well as between the achieved gain and the implemented level of service (i.e., the time window size). Reducing f_{car} yields an increase of f_{dist} for 45.7% of the feasible instances with fewer cars. Indeed, reducing the number of cars might create a need for detours in the drivers' planning (to pick up and drop off non-motorized workers) that cannot be compensated by merging paths. However, we observe that the introduction of both carpooling and walking is able to generate a simultaneous gain, both in terms of fleet size and total driving distance, for 25% of the considered instances. Increasing the service level (i.e., shrinking the time window size) decreases the achieved gain for both f_{car} and f_{dist}. With a smaller time window size, the number of jobs that can be reached on foot decreases, which ultimately leads to an increased need for detours to drop off and pick up non-motorized workers. This will be further analyzed in Section 6.2. However, a reduction of 15.4% in terms of the size of the vehicle fleet is still achieved for instances with *quarter day* time windows.

To go beyond these average results with respect to the associated optimal VRP solutions, the following observations can be made. For *all day* time windows, the largest obtained reduction in terms of driving distance is 16.6% (for instance 40_A_8). For *half day* time windows, the largest achieved gain is 19% (for instance 40_H_9). Finally, for *quarter day* time windows, the largest

observed reduction in driving distance is 8.3% (for instance 50_Q_1). Regarding the reduction of the fleet size, the largest gain is 50% (e.g., for instances 20_A_1, 20_H_5, and 30_Q_10). As several instances (e.g., 20_Q_3, 40_A_3) do not exhibit any gain (over the ten runs) either in terms of driving distance or with respect to the size of the vehicle fleet, the smallest achieved gain is therefore 0%. Detailed results for the 120 considered instances can be found in Appendix B.

6.2. Instance characteristics that favor carpooling

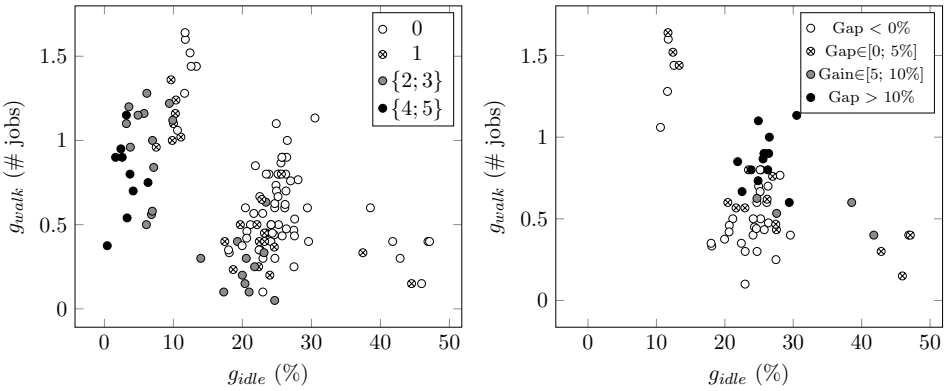
In the previous section, we analyzed the gain offered by the VRPTR formulations over the 120 considered instances. In this section, we focus on understanding which instance characteristics can be linked to the magnitude of the achieved gain.

6.2.1. Idle time and walking potential

Two indicators, g_{walk} and g_{idle}, are discussed. First, the walkability g_{walk} characterizes the walking potential of an instance. It represents the average number of jobs reachable on foot from a given job. A job j is said to be reachable on foot from job i if the walking time from i to j (τ^f_{ij}) is less than τ^f_M (15 minutes in our experiments) and, if leaving job i as early as possible, the worker arrives on time at job j. More precisely, let J^R_i be the set of jobs reachable from job $i \in J$. Formally, we have $J^R_i = \left\{ j \in J \;\middle|\; \tau^f_{ij} \leq \tau^f_M;\, e_i + p_i \cdot \tau_{ij} \leq l_j \right\}$. Hence, $g_{walk} = \frac{\sum_{i \in J} |J^R_i|}{|J|}$. Depending on the instance, g_{walk} lies between 0 and 1.5 (which is in line with EEP's field observations). Second, the idle time, denoted by g_{idle}, is the percentage of time during which the workers are idle in the corresponding optimal VRP solutions. On the one hand, tightening the time window reduces g_{walk} (fewer jobs can be reached on foot without time window violations) but increases g_{idle} (waiting appears en route for the start of a job). On the other hand, g_{walk} increases with n. Indeed, the density of jobs increases, and hence, on average, more jobs can be reached on foot from a given position. Figure 7 focuses on the results of $\left(P^{|W^\star|-1}_{walk}\right)$ according to g_{idle} and g_{walk}. Each instance is positioned on the x-axis and y-axis according to g_{idle} and to g_{walk}. More specifically, for each instance, Figure 7(a) gives the number of jobs that cannot be served in $\left(P^{|W^\star|-1}_{walk}\right)$. Figure 7(b) plots the gap found with the optimal VRP solution (i.e., $\frac{d^\star(P^{|W^\star|}_{no\ walk}) - d(P^{|W^\star|-1}_{walk})}{d^\star(P^{|W^\star|}_{no\ walk})}$) for the 58 feasible instances for configuration $\left(P^{|W^\star|-1}_{walk}\right)$.

Figure 7(a) indicates that for instances with $g_{idle} < 10\%$, no feasible solution can be found for $\left(P^{|W^\star|-1}_{walk}\right)$. To put that number into perspective, $g_{idle} = 10\%$ represents an idle time of 45 minutes

per worker. Figure 7(b) highlights that the gain increases with the walking potential g_{walk} of the instance. Moreover, for an idle time between 15% and 30%, it is necessary, from a given job, to have on average more than 0.5 jobs reachable on foot to find competitive solutions regarding the driving distance.



(a) Distribution of unserved jobs

(b) Distribution of f_{dist}-gain

Each point corresponds to an instance defined by its indicators g_{idle} (x-axis) and g_{walk} (y-axis)
(a) The number of unserved jobs is denoted by the color code given in the upper right corner.
(b) For the feasible instances, the f_{dist}-gain range is denoted by the color code given in the upper right corner.

Figure 7: Distribution of the feasible instances and f_{dist}-gains for configuration $\left(P_{walk}^{|W^\star|-1}\right)$ (i.e., one car is removed from the VRP optimal solution).

6.2.2. Geographical characteristics

To keep the recommendations as general as possible, some instance characteristics that favored walking and carpooling were not explicitly taken into account in the preceding sections. Denser urban configurations are likely to appear in practice, as the considered instances have a job density between 0.2 and 0.5 jobs per km^2. Ultimately, only 3% of the arcs are actually eligible for traveling on foot. A greater density of jobs per km^2 would increase g_{walk} and, hence, the efficiency of the VRPTR formulation, as discussed in Section 6.2.1. Considering congestion or parking time explicitly would also substantially reduce the gap between walking time and driving time for some arcs. Additionally, walking would sometimes becomes a mandatory option in pedestrian zones.

Other experiments (not reported here) have shown that for the 40 instances with *all day* time windows, considering a non-centered depot (located at one of the corners of the 10 km by 10 km square grid) increases the efficiency of scenario $S^{(car)}$, as the f_{car}-gain goes up from 23% to

29.2% on average, whereas the f_{dist}-gain jumps from 1.9% to 8.8%. Indeed, in such a situation, the optimal VRP solutions are likely to route workers through close paths from the depot to the customer locations at the beginning of the working day and back to the depot at the end of it. A consolidation of these travels to and from the depot is hence expected to arise.

6.3. Expected impact of random perturbations

In this paper, we considered a static case where all the service times and travel data are assumed to be known. The aim of this study is to measure the benefits of walking and carpooling. However, in practice, the actual service and travel times are likely to differ from the forecasted ones. In the following, we discuss the expected impacts resulting from such perturbations and what mechanisms could be envisioned to overcome such issues.

Compared with VRP solutions, the routes involving carpooling are interdependent. As a consequence, an unexpected event on one route can potentially modify the schedule of all the interconnected routes. For example, considering the VRPTR solution displayed in Figure 6, if worker w_3 is delayed on her/his route (gray line) because of longer service or travel times, s/he will only be able to pick up the non-motorized worker w_4 behind schedule, which would ultimately lead to a late arrival at the depot for both workers. Robust approaches, such as those derived in the VRP context (e.g., (Lu and Gzara 2019)), could be extended to the VRPTR case. In the context of online VRP (see (Pillac et al. 2013) for a recent review), Lorini et al. (2011) and Respen et al. (2017) propose a solution method to rebuild – in real-time – solutions that have been modified by unexpected events. Their methods rely on fast moves, including the reassignment of a limited number of jobs or vehicle diversion (i.e., modify the current destination of the vehicle). In the VRPTR situation, taxi services (Zufferey et al. 2016) could also be envisioned as an urgency back-up. Removing future WRs in driver routes would also be an appropriate technique for overcoming delays in their routes.

7. Conclusion, perspectives, and future works

This study considers a new type of VRP called the *Vehicle Routing Problem with Transportable Resources* (VRPTR), where vehicles and walking workers coexist and must be synchronized to satisfy a given set of jobs spread over a territory. Such a formulation has not yet been introduced in the literature, and it opens up new perspectives in decision-making processes for routing problems. The proposed modeling framework is suitable for each situation in which two distinct transportation

resources are available: the transporter ones (e.g., cars, trucks, or buses), which move autonomously, and the transportable ones (e.g., pedestrians, scooters, bicycles, or drones), which can either move autonomously or be transported by a transporter one for parts of their route.

In this contribution, we treat an application of the VRPTR that comes from an industrial case. It involves cars, walking, and carpooling. Coordinating all these transportation options allows managers to generate a brand new set of solutions in a routing context. Considering an urban context, we evaluate the potential of such a novel formulation, as simultaneous savings can be achieved both in the total driving distance and in the number of vehicles, even with tight time windows. Obviously, when focusing on these two objectives, the gains obtained depend on the instance characteristics. We have identified some conditions under which a significant gain can be expected. For instances involving idle times (inside the routes due to time window constraints or at the end of the routes), the new formulation is able to invest them efficiently in carpooling and walking. Further works could explore in detail the trade-off that arises between decreasing the need for resources and the total en route time, as done in a Green VRP context in Demir et al. (2014).

8. Acknowledgement

The authors would like to thank Professor Guy Desaulniers (Polytechnique Montreal and GERAD, Canada) for proving the optimality of the VRP solutions with GENCOL. Thanks are due to the reviewers for their valuable comments. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Agatz, N., Bouman, P., and Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981.

Baldacci, R., Mingozzi, A., and Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6.

Boysen, N., Schwerdfeger, S., and Weidinger, F. (2018). Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, 271(3):1085–1099.

Chao, I.-M. (2002). A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29(1):33–51.

Cherkassky, B. V., Georgiadis, L., Goldberg, A. V., Tarjan, R. E., and Werneck, R. F. (2009). Shortest-path feasibility algorithms: An experimental evaluation. *Journal of Experimental Algorithmics*, 14:7.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W., and Vigo, D. (2007). Vehicle routing. *Handbooks in Operations Research and Management Science*, 14:367–428.

Daimler (2017). Vans & robots paketbote 2.0. *https://www.daimler.com/innovation/specials/future-transportation-vans/paketbote-2-0.html*.

Dechter, R., Meiri, I., and Pearl, J. (1991). Temporal constraint networks. *Artificial Intelligence*, 49(1-3):61–95.

Demir, E., Bektaş, T., and Laporte, G. (2014). The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232(3):464–478.

Derigs, U., Pullmann, M., and Vogel, U. (2013). Truck and trailer routing problems, heuristics and computational experience. *Computers & Operations Research*, 40(2):536–546.

Desaulniers, G., Lessard, F., and Hadjar, A. (2008). Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404.

Domínguez-Martín, B., Rodríguez-Martín, I., and Salazar-González, J.-J. (2018). The driver and vehicle routing problem. *Computers & Operations Research*, 92:56–64.

Drexl, M. (2012). Synchronization in vehicle routing: a survey of VRPs with multiple synchronization constraints. *Transportation Science*, 46(3):297–316.

Drexl, M. (2014). Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments. *Networks*, 63(1):119–133.

Ferrandez, S. M., Harbison, T., Weber, T., Sturges, R., and Rich, R. (2016). Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management*, 9(2):374.

Fikar, C. and Hirsch, P. (2015). A matheuristic for routing real-world home service transport systems facilitating walking. *Journal of Cleaner Production*, 105:300–310.

Ghiani, G. and Laporte, G. (2001). Location-arc routing problems. *Opsearch*, 38(2):151–159.

Grangier, P., Gendreau, M., Lehuédé, F., and Rousseau, L.-M. (2016). An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *European Journal of Operational Research*, 254(1):80 – 91.

Gussmagg-Pfliegl, E., Tricoire, F., Doerner, K., and Hartl, R. (2011). Mail-delivery problems with park-and-loop tours: a heuristic approach. In *Proceedings of the ORP3 Meeting, Cadiz*.

Jabali, O., Woensel, T., and de Kok, A. (2012). Analysis of travel times and CO2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6):1060–1074.

Knörr, W. (2008). Ecotransit: ecological transport information tool environmental methodology and data. Technical report, Institut fr Energie (ifeu) und Umweldforschung Heidelberg GmbH, http://www.ecotransit.org/download/ecotransit_background_report.pdf.

Lam, E., Van Hentenryck, P., and Kilby, P. (2015). Joint vehicle and crew routing and scheduling. In *International Conference on Principles and Practice of Constraint Programming*, pages 654–670. Springer.

Levy, L. and Bodin, L. (1989). The arc oriented location routing problem. *INFOR: Information Systems and Operational Research*, 27(1):74–94.

Lin, C. (2008). A cooperative strategy for a vehicle routing problem with pickup and delivery time windows. *Computers & Industrial Engineering*, 55(4):766–782.

Lin, S.-W., Vincent, F. Y., and Chou, S.-Y. (2009). Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, 36(5):1683–1692.

Lorini, S., Potvin, J.-Y., and Zufferey, N. (2011). Online vehicle routing and scheduling with dynamic travel times. *Computers & Operations Research*, 38(7):1086–1090.

Lu, D. and Gzara, F. (2019). The robust vehicle routing problem with time windows: Solution by branch and price and cut. *European Journal of Operational Research*, 275(3):925–938.

Masson, R., Lehuédé, F., and Péton, O. (2013). Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers. *Operations Research Letters*, 41(3):211–215.

Masson, R., Lehuédé, F., and Péton, O. (2014). The dial-a-ride problem with transfers. *Computers & Operations Research*, 41:12–23.

Meisel, F. and Kopfer, H. (2014). Synchronized routing of active and passive means of transport. *OR spectrum*, 36(2):297–322.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100.

Murray, C. C. and Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109.

Parragh, S. and Cordeau, J.-F. (2017). Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. *Computer & Operations Research*, 83:28–44.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11.

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435.

Poikonen, S., Wang, X., and Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1):34–43.

Respen, J., Zufferey, N., and Potvin, J.-Y. (2017). Impact of vehicle tracking on a routing problem with dynamic travel times. *RAIRO-Operations Research*.

Rochat, Y. and Semet, F. (1994). A tabu search approach for delivering pet food and flour in switzerland. *Journal of the Operational Research Society*, 45(11):1233–1246.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.

Savelsbergh, M. W. (1992). The vehicle routing problem with time windows: minimizing route duration. *ORSA Journal On Computing*, 4(2):146–154.

Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., and Dueck, G. (2000). Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2):139–171.

Semet, F. (1995). A two-phase algorithm for the partial accessibility constrained vehicle routing problem. *Annals of Operations Research*, 61(1):45–65.

Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report, Department of Computer Science, University of Strathclyde, Scotland.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. *Lecture Notes in Computer Science*, 1520:417–431.

Smilowitz, K. (2006). Multi-resource routing with flexible tasks: an application in drayage operations. *Iie Transactions*, 38(7):577–590.

Tilk, C., Bianchessi, N., Drexl, M., Irnich, S., and Meisel, F. (2017). Branch-and-price-and-cut for the active-passive vehicle-routing problem. *Transportation Science*, 52(2):300–319.

Wohlsen, M. (2014). The next big thing you missed: Amazon?s delivery drones could work–they just need trucks. *Wired. com.[http://www. wired. com/2014/06/the-nextbig-thing-you-missed-delivery-drones-launched-from-trucks-are-the-future-ofshipping/. Accessed 10/15/2014.]*.

Zufferey, N., Cho, B. Y., and Glardon, R. (2016). Dynamic multi-trip vehicle routing with unusual time-windows for the pick-up of blood samples and delivery of medical material. In *Proceedings of the 5th International Conference on Operations Research and Enterprise Systems, ICORES 2016, Rome, Italy, February 23 25*.