CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

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

Marc-Antoine Coindreau
Olivier Gallay
Nicolas Zufferey

December 2018

CIRRELT-2018-50

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

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

Marc-Antoine Coindreau[1], Olivier Gallay[1,*], Nicolas Zufferey[2]

1. Department of Operations, HEC, University of Lausanne, CH-1015 Lausanne, Switzerland
2. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and 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: olivier.gallay@unil.ch

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2018

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 has recently gained increasing importance. Firms that provide on-site services or parcel deliveries are directly concerned by these issues as a substantial part of their activities take 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, the primary objective for EEP is to minimize the number of technicians to serve all jobs. Then, EEP minimizes the remaining costs implied by the technician routes. These costs are directly proportional to the use of cars (total driving distance and total number of cars required).

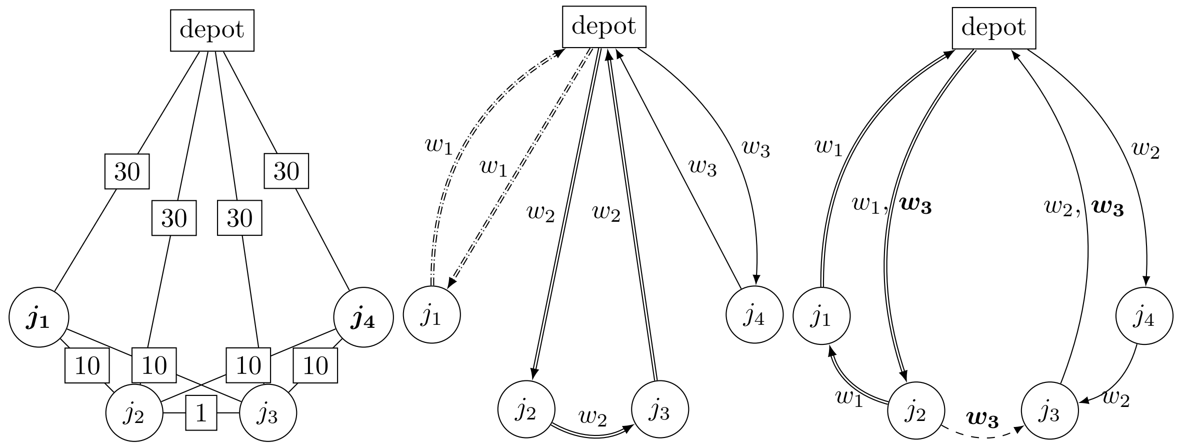
EEP routes thousands of workers in urban areas, and more than a million kilometers are driven every year. In that respect, EEP aims to evaluate the savings potential generated by the use of walking to reduce the total costs of its routes, but also to meet the workers' expectations. EEP observed that their technicians often leave their vehicle to perform clustered jobs on foot even if their planning would recommend driving to the next job. EEP also wants to go one step further by evaluating the savings potential engendered by 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, as several vehicles are traveling on similar paths (e.g., from the depot to concentrated zones of clients).

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 a lot of 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 vehicle and are seen as transportable resources that can move autonomously. We refer to this extension as the *Vehicle Routing Problem with Transportable Resources* (VRPTR), and we allow for the modeling of every situation where workers have to visit clients without any delivery or transportation of heavy equipment, making hence 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. It illustrates how a VRPTR solution works. The characteristics of the instance are given on the left part. Compared with the VRP solution (middle part of the figure), the VRPTR solution (right part of the figure) provides improved efficiency: same number of workers, one car saved, and the total driving distance is reduced by 22.6%.



(a) Values on the edges 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) Vehicle's path is drawn with a specific line style; walking is represented with a dashed line; the label of an edge 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 an efficient neighborhood structure to find competitive solutions that involve both walking and carpooling. Whereas the MILP is able to tackle instances up to 16 customers, the metaheuristic can solve all other instances, which involve up to 50 jobs. Compared with current practices on a representative set of instances capturing urban characteristics, the computational experiments yield an average improvement of 6.5% on 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, for 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 solution methodology, and we detail the considered instances.

Section 5 presents the computational experiments and the results. Section 6 proposes managerial insights (e.g., quantifying the gains compared with current practices and understanding the promising configuration 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 consider the use of autonomous and transportable resources. Next, we describe the solution methodologies that have proven to be efficient for such related problems.

A formulation that considers both walking and driving was originally introduced by Levy and Bodin (1989) for mail delivery purposes. It is referred to as *Park-and-Loop* and it is generalized by Ghiani and Laporte (2001) as *Location-Arc Routing Problem*. It considers situations in which the postman parks his/her car to visit 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 edges, 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 problem has received more attention under the *Truck and Trailer Routing Problem* (TTRP) formulation (Chao 2002, Lin et al. 2009). 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 as the motivation for *Park-and-Loop* sub-tours differs. It 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 requires synchronizing on-foot workers with cars. As reviewed in (Drexl 2012), recent contributions have considered the synchronization of different transportation resources in a VRP context. 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, reduction 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* (FSTSP), 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 main 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 (which stand for the synchronized light resources) 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.

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 light resources alone, and then creating routes for the heavy resources to connect the clusters together (Levy and Bodin 1989, Fikar and Hirsch 2015). The second alternative proposes to first build routes for the heavy vehicles without using the light resources, and then to assign some clients to them (Ghiani and Laporte 2001, Gussmagg-Pfliegl et al. 2011, Murray and Chu 2015). Even if 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 approach, 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 numerous successes 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

3.1. Complexity of the problem

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), Toth and Vigo (2014) for overviews of the various VRP characteristics, their associated models, and their efficient solution approaches).

3.2. Definition and assumptions

From a practical standpoint, 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). Indeed, 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).
- The walking limitations are the maximum daily walking distance (d_M^f) for each worker and the maximum walking time (τ_M^f) to reach the next job.
- 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).
- Both driver and passenger workers can walk to reach a job, but in the driver case, the return path to his/her car is mandatory: a driver WR is a cycle (departure and arrival points must coincide), whereas a passenger WR is either a cycle or a chain (departure and arrival points can be different).
- Idling is allowed for both the drivers and the passengers at job locations.

3.3. Graph modeling and variables

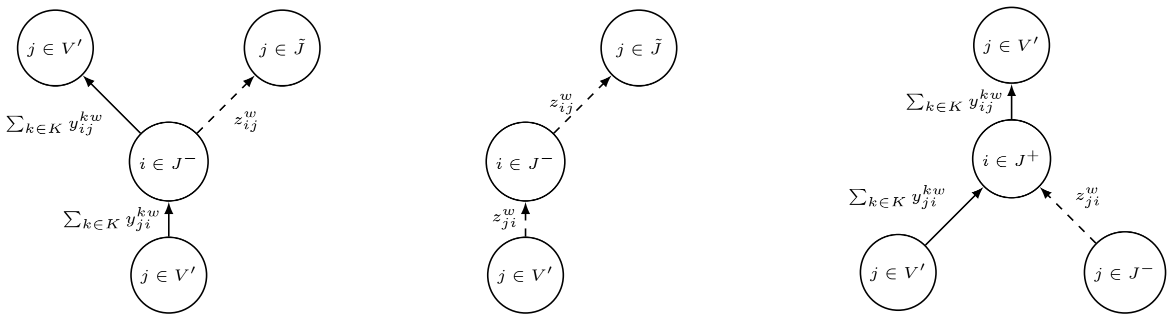
$J = \{1, \dots, n\}$ represents the job set (each job has to be performed/served exactly once by a worker). The depot, where all workers and vehicle routes depart from and return to, is denoted by node 0. A processing time p_j and a time window $[e_j, l_j]$ (within which the service has to take place) are associated with each job $j \in J$. Let W be the set of workers and K be the set of vehicles. q is the maximum number of workers per car. The VRPTR is defined on a complete graph $G = (V, A)$, where $V = J \cup \{0\}$ represents the node set, and $A = \{(i, j) \mid i, j \in V, i \neq j\}$ represents the arc set. A driving time $\tau^d_{i,j}$, a walking time $\tau^f_{i,j}$, and a distance $d_{i,j}$ are associated with each arc $(i, j) \in A$. These quantities are symmetric.

The node set J is duplicated to $J^- = \{1, 2, \dots, n\}$ (entry or intermediate points of a WR) and $J^+ = \{n + 1, n + 2, \dots, 2 \cdot n\}$ (ending points of a WR), $\tilde{J} = J^- \cup J^+$. The nodes $i \in J^-$ and $n + i \in J^+$ represent the same node ($i \in J$). When a worker starts a WR located at node $i \in J$, s/he is dropped off at the corresponding node $i \in J^-$. When a worker walks between two jobs, s/he uses the arcs between two nodes of J^-. When the worker ends his/her WR, s/he goes to the associated node in J^+. The travel time and cost to move from $i \in J^-$ to $n + i \in J^+$ are null, and a car can go from $i \in J^-$ to $n+i \in J^+$ without any driver (as it is the same node in the real network). $G' = (V', A')$ is an augmented graph, where $V' = \{0\} \cup \tilde{J}$, and $A' = \{(i, j) \mid i, j \in V', i \neq j\}$. With these notation, a cycle in G (i.e., a driver parks his/her car at $j \in J$, serves a set of jobs by walking and comes back to his/her car at j) is not a cycle in G' (s/he parks his/her car at $j \in J^-$ and finishes his/her WR at $j + n \in J^+$). The distance, as well as the driving and the walking times associated with an arc in G', are equal to the corresponding value in the real network G.

We define the variables:

- $x^k_{ij} = 1$ if a vehicle $k \in K$ uses arc $(i, j) \in A'$; $x^k_{ij} = 0$, otherwise,
- $y^{kw}_{ij} = 1$ if a vehicle $k \in K$ transports worker $w \in W$ on arc $(i, j) \in A'$; $y^{kw}_{ij} = 0$, otherwise,
- $z^w_{ij} = 1$ if a worker $w \in W$ walks on arc $(i, j) \in A'$; $z^w_{ij} = 0$, otherwise,
- t^w_i denotes the time at which worker $w \in W$ arrives at node $i \in V'$,
- s_i stands for the time at which the service starts at node $i \in J^-$,
- h^k_i (resp. $\bar{h}^k_i$) denotes the time at which vehicle $k \in K$ arrives (resp. leaves) node $i \in V'$.

Due to carpooling, and contrary to the standard VRP formulations, a worker can visit a node without performing the associated job. Indeed, when a worker is dropped off at a node $i \in J^-$, a vehicle with potentially other passengers stops at node i, but only the dropped off 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., $\sum_{j \in \tilde{J}} z_{ij}^w = 1$). Figure 2 illustrates the flow of workers at nodes in J^- and at nodes in J^+.



(a) Flow $i \in J^-$.　(b) Flow $i \in J^-$.　(c) Flow $i \in J^+$.

Dashed (resp. plain) lines denote an arc traveled on foot (resp. with a car).
(a) If worker w arrives in J^- by car, s/he can exit the node either on foot or by car.
(b) If worker w arrives in J^- on foot, s/he must exit on foot.
(c) Workers arrive in J^+ either by car or on foot but must exit by car.

Figure 2: Different flow configurations in $\tilde{J}$.

3.4. Mathematical formulation

We propose a MILP model for the VRPTR. The number of vehicles and the number of workers are defined as constraints. To minimize these values, the MILP is launched by successively removing one worker or one vehicle until all the jobs can no longer be performed. The objective function and the various constraints are now presented.

Objective function:

$$\text{minimize} \sum_{(i,j) \in A'} d_{ij} \sum_{k \in K} x_{ij}^k \tag{1}$$

Objective (1) is the total traveled distance, which is minimized for a given quantity of resources (i.e., a specific number of workers and vehicles). Indeed, the EEP cost function includes (1) the fixed costs associated with each employed worker (daily salary) and car (daily amortization), and (2) the variable costs that are proportional to the driving distance. As the number of used resources (workers and cars) is given as input to the MILP, the only remaining costs to be minimized are those related to the driving distance.

Walking routes constraints:

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

$$\tau_{ij}^{f} \cdot z_{ij}^{w} \leq \tau_{M}^{f}, \qquad i, j \in V', \, w \in W \tag{3}$$

$$\sum_{i \in V'} \sum_{j \in V'} d_{ij} \cdot z_{ij}^{w} \leq d_{M}^{f}, \qquad w \in W \tag{4}$$

Constraints (2) ensure that all the jobs are performed. Note that, since a node in J^{+} represents the end of a WR, it is possible that no worker will arrive at a node in J^{+} by walking. Constraints (3) and (4), respectively, define an upper bound on the walking time between any pair of jobs, and on the total daily walking distance of a worker.

Flow constraints on workers:

$$\sum_{k \in K} \sum_{j \in V'} y_{ji}^{kw} + \sum_{j \in V'} z_{ji}^{w} = \sum_{k \in K} \sum_{j \in V'} y_{ij}^{kw} + \sum_{j \in V'} z_{ij}^{w}, \qquad i \in J^{-}, \, w \in W \tag{5}$$

$$\sum_{k \in K} \sum_{j \in V'} y_{ij}^{kw} + \sum_{j \in V'} z_{ij}^{w} \leq 1, \qquad i \in V', \, w \in W \tag{6}$$

$$\sum_{j \in J^{-}} z_{ji}^{w} \leq \sum_{j \in \tilde{J}} z_{ij}^{w}, \qquad i \in J^{-}, \, w \in W \tag{7}$$

$$\sum_{k \in K} \sum_{j \in V'} y_{ji}^{kw} + \sum_{j \in J^{-}} z_{ji}^{w} = \sum_{k \in J} \sum_{j \in V'} y_{ij}^{kw}, \qquad i \in J^{+}, \, w \in W \tag{8}$$

Constraints (5) ensure that a worker arriving at node $i \in J^{-}$ (either by walking or by car) ultimately exits the node. Constraints (6) state that a worker cannot use the two different transportation modes (i.e., walking and driving) to leave a node. Constraints (7) guarantee that a worker arriving at node $i \in J^{-}$ by walking has to exit the node by walking (if a worker exits a node by car, it means that the end of the WR has been reached, but according to the model, it is a node in J^{+}). Finally, constraints (8) force any worker arriving at a node $i \in J^{+}$ to exit the node by car.

Flow constraints on vehicles:

$$\sum_{j\in V'} x^k_{ji} = \sum_{j\in V'} x^k_{ij}, \qquad k \in K,\, i \in \tilde{J} \tag{9}$$

$$x^k_{ij} \le \sum_{w\in W} y^{kw}_{ij}, \qquad i \in V',\, j \in V'\backslash\{i+n\},\, k \in K \tag{10}$$

$$y^{kk}_{ij} \ge y^{kw}_{ij}, \qquad i \in V',\, j \in V',\, k \in K,\, w \in W \tag{11}$$

$$\sum_{w\in W} y^{kw}_{ij} \le q \cdot x^k_{ij}, \qquad i, j \in V',\, k \in K \tag{12}$$

$$\sum_{i\in V'} y^{kw}_{ij} \le 1, \qquad k \in K,\, w \in W,\, j \in V' \tag{13}$$

$$\sum_{i\in V'} x^k_{ij} \le 1, \qquad j \in V',\, k \in K,\, w \in W \tag{14}$$

$$\sum_{k\in K}\sum_{i\in V'} y^{kw}_{0i} = \sum_{k\in K}\sum_{i\in V'} y^{kw}_{i0}, \qquad w \in W \tag{15}$$

$$\tag{16}$$

Constraints (9) ensure that vehicles arriving at a node ultimately exit the node. Constraints (10) guarantee that a vehicle does not move without a driver, and constraints (11) impose that a car is always driven by the same driver. Constraints (12) ensure that the vehicle capacity is always satisfied. Constraints (13) and (14) forbid a worker or a vehicle, respectively, to use two different edges. Constraints (15) state that every worker who leaves the depot has to come back to it in a single trip.

Time constraints (with notation $l_{max} = \max_{j\in J} l_j$, $p_{max} = \max_{j\in J} p_j$):

$$\bar{h}^k_i \ge h^k_i, \qquad i \in V'\backslash\{0\},\, k \in K \tag{17}$$

$$s_i \ge t^w_i, \qquad i \in J^-,\, w \in W \tag{18}$$

$$t^w_i \ge s_j + p_j + \tau^f_{ji} - (\tau^f_{ji} + l_{max} + p_{max}) \cdot (1 - z^w_{ji}), \qquad i \in \tilde{J},\, j \in J^-,\, w \in W \tag{19}$$

$$t^w_i \ge h^k_i, \qquad i \in J^-,\, k \in K,\, w \in W \tag{20}$$

$$\bar{h}^k_i \ge t^w_i, \qquad i \in J^+,\, k \in K,\, w \in W \tag{21}$$

$$h^k_j \ge \bar{h}^k_i + \tau^k_{ij} - (\tau^k_{ij} + l_{max} + p_{max}) \cdot (1 - x^k_{ij}), \qquad i \ne j,\, k \tag{22}$$

$$s_i \in [e_i, l_i], \qquad i \in J^- \tag{23}$$

$$\bar{h}^k_0 \ge e_0, \qquad k \in K \tag{24}$$

$$h^k_0 \le l_0, \qquad k \in K \tag{25}$$

Constraints (17) ensure that the departure time of each vehicle always occurs after its arrival time. Constraints (18) state that the service at node $i \in J^-$ begins after the arrival time of the associated worker. Constraints (19) and (20) establish the worker arrival time at node $i \in J^-$. More precisely, constraints (19) are active if a worker arrives at i by walking, whereas constraints (20) are active when a worker is dropped off by a car. Constraints (21) ensure the synchronization between the worker (at the end of a WR) and the vehicle that must pick him/her up. Constraints (22) set the vehicle arrival times after traveling through an arc. Constraints (23) ensure that the time windows are satisfied. Finally, constraints (24) and (25) guarantee that vehicles (and workers) leave and arrive at the depot within the workers' time window.

Fixing variables to 0*:*

$$z_{ij}^{w} = 0, \qquad i \in J^+,\ j \in V', w \in W \tag{26}$$

$$x_{j,j-n}^{k} = 0, \qquad j \in J^+,\ k \in K \tag{27}$$

$$y_{j,j-n}^{k,w} = 0, \qquad j \in J^+,\ k \in K,\ w \in W \tag{28}$$

Constraints (26) state that a worker cannot exit a node in J^+ by walking (because the nodes in J^+ can only be the ending nodes of the WRs). Finally, constraints (27) and (28) forbid cars and passengers to go from a node of J^+ to the corresponding node of J^-.

4. Methodology

This section starts by describing the general principles of the proposed *Variable Neighborhood Search* (VNS). Next, we present its specific neighborhood structure. Finally, we introduce a dedicated algorithm to speed up the insertion heuristic that is employed as a key procedure of our VNS.

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 $s = \{x, y, z\}$, $d(s) = \sum\limits_{k \in K} \sum\limits_{(i,j) \in A} d_{ij} \cdot x_{ij}^{k}$ is the overall driving distance. $J^{in}(s) = \{j \in J \mid \sum_{i \in \tilde{J}} z_{ji} = 1\}$ is the set of jobs that are inserted in s, and $J^{out}(s) = J \backslash J^{in}(s)$ is the set of non-inserted 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 $c(s) < c(s')$.

$$c(s) = d(s) + \psi \cdot \mid J^{out}(s) \mid \tag{29}$$

4.1. VNS: general principles

As mentioned in Section 2, existing successful solution approaches for related VRP formulations have combined LNS and a local search (LS). We propose to combine these two ingredients within a VNS, (Mladenović and Hansen 1997). Intensification (resp. diversification) is made possible thanks to the use of a LS procedure (resp. a collection of neighborhood structures of different amplitudes, as in LNS). A generic version of the VNS is given in Algorithm 1. It takes as input an initial solution s and 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 involved solution more than $\mathcal{N}_{q-1}$).

The implemented LS is a descent algorithm. Formally, at each step, a neighbor solution $s^{neighbor}$ is generated from the current solution s with the *reinsertion* move. 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 from 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.

Algorithm 1 Variable Neighborhood Search (VNS)

Generate an initial solution s and set $q = 1$.

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

(1) *Shaking:* generate randomly 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. VNS: shaking phase

It is important to consider large neighborhoods to tackle the VRPTR, since the presence of WRs is likely to trap the search in local minima. Indeed, unless a WR is not completely removed from the solution, a natural bias would be to keep inserting the same jobs in the same WR (as an insertion with walking is always cheaper than an insertion with driving). Neighborhood $\mathcal{N}_q$ is explored by means of an LNS-type procedure, based on the sequential use of a *removal* heuristic (Subsection 4.2.1) and an *insertion* heuristic (Subsection 4.2.2). In other words, $q \in \mathbb{N}$ jobs are first removed from the current solution s (i.e., q jobs are moved from $J^{in}(s)$ to $J^{out}(s)$), and then they are inserted back into the solution in hopefully different positions.

4.2.1. Removal heuristic

The removal heuristic is aimed at dropping jobs that currently block the search process in a local minimum. We consider here the *related removal heuristic* (RRH) proposed in 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, $\text{WR}(i, j) = 1$ 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. $R(i, j)$ is computed as follows:

$$R(i, j) = \alpha \cdot d_{ij} + \beta \cdot |h_i - h_j| + \gamma \cdot |l_i - l_j| + \delta \cdot (1 - \text{WR}(i, j)) + \epsilon \cdot (1 - \mathbb{1}_{k_i=k_j}) \tag{30}$$

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.

RRH can now be described as follows. The first removed job is randomly selected in $J^{in}(s)$. Next, as long as q removals have not been performed, jobs in $J^{in}(s)$ are ranked according to their relatedness with one of the removed jobs in a list L (the first job is the most related). The job $L[\lfloor y^\rho \cdot |J^{in}(s)| \rfloor]$ is removed, where 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 (with $\rho = 1$, jobs are randomly removed).

4.2.2. Insertion heuristic

The insertion procedure moves jobs from $J^{out}(s)$ to $J^{in}(s)$ until $J^{out}(s)$ is empty, or until no feasible insertion can be done (i.e., no worker can perform a new job in his/her schedule). In the latter case, the non-feasibility of the resulting solution is penalized in the objective function (29).

Among the related works that use LNS as a diversification process (e.g., Ropke and Pisinger (2006), Masson et al. (2014), Grangier et al. (2016)), best-insertion algorithms are considered: the first inserted job is the one that minimizes the insertion cost (i.e., the additional driving distance in our case). Unfortunately, a best-insertion heuristic is inefficient in the VRPTR context because it creates biases by favoring either (1) insertions involving walking only or (2) insertions in a driver's planning. For (1), it is because walking does not increase the driving distance. For (2), moving a driver to a job only requires one detour with his/her assigned car, whereas assigning this job to a passenger requires two detours: one for the drop-off and one for the pick-up. These two biases generate two main drawbacks. First, it limits 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 the two above biases due to carpooling and walking.

On the one hand, to overcome the problem of over-insertions in drivers' planning, we propose a *Random Worker Best-Insertion* (RWBI) heuristic. In each step of the RWBI, a worker is first randomly chosen, and then the cheapest insertion is achieved. On the other hand, walking has two advantages. First, it does not contribute directly to the objective function. Second, the more the passengers will walk, the less the drivers have to serve transportation requests coming from the passengers and the more they will be able to perform jobs themselves. The major drawback is that walking also reduces the workers' availability and thus augments the risk of having non-performed jobs remaining at the end of the shaking phase. To avoid the overuse of walking, a trade-off is thus proposed here, where the walking range of a passenger is randomly redefined in $[0, \tau_M^f]$ (at the beginning of each application of an RWBI). Additionally, among all the feasible insertion positions, we randomly select a non-dominated insertion. As a result, instead of only considering driving distance augmentation, we also consider increasing the walking time. 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. Because walking reduces the drivers' availability, 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).

4.3. Fast insertion heuristic

The best-insertion move is the core part of both RWBI and LS. As presented in Appendix A, finding the best insertion position for a job requires $\mathcal{O}(n^5)$ feasibility tests for the VRPTR, whereas it only requires $\mathcal{O}(n)$ tests for the VRP. Therefore, being able to fasten the procedure to check the feasibility of an insertion position turns out to be crucially important. In Appendix A, we present a procedure to evaluate in constant time if an insertion position is valid regarding all synchronization constraints. Experiments have shown that using this fast feasibility check can reduce the computation time by 95%.

5. Computational experiments

We describe 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 exposes the results of the MILP, and Section 5.4 analyzes the performance of the proposed VNS. Finally, Section 5.5 gives the results of the VNS for the full instance set. 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 launched on a 2.2 GHz Intel Core i7 with 16 Go 1600 MHz DDR3 of RAM memory. 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.

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 and euclidian distances between jobs are considered. The driving speed is 30 km/h and 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 literature considering the en route synchronization of transportable resources (e.g., Boysen et al. (2018) solve real-world instances for up to 40 customers).

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, namely *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 with 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 (10 instances have been generated for each n and TW), and n_TW denotes the set of 10 instances of size n and time window size TW.

5.2. Notation and considered configurations

Using the solution method developed by Desaulniers et al. (2008), which is acknowledged to be one of the most efficient algorithms for the VRP (see (Baldacci et al. 2012) for instance), the optimal VRP solution has been computed for each instance. Accordingly, we know the associated smallest number of workers $|W^\star|$ required to serve all jobs. In the following, we consider different configurations $\left(P_c^{|K|}\right)$, where $|K|$ designates the number of used vehicles and $c \in \{walk, \textit{no walk}\}$ indicates whether walking is allowed or not. $d\left(P_c^{|K|}\right)$ (resp. $d^\star\left(P_c^{|K|}\right)$) gives the total driving distance for configuration $(P_c^{|K|})$ 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).

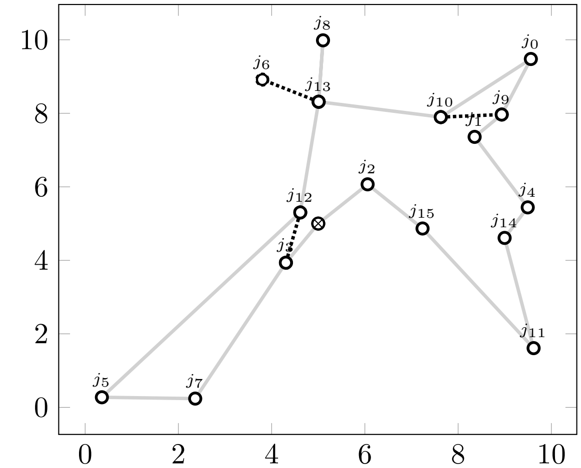
The following five $\left(P_c^{|K|}\right)$ are considered:

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

Instances involving up to $n = 16$ jobs were solved by the MILP with less than 2 hours of computation time. For instances involving $n = 17$ jobs, the MILP could not find feasible solutions after 5 hours of computation. A solution obtained by the MILP is shown in Figure 3. It exhibits both carpooling and walking, and points out the efficient synchronization that arises between a driver and a passenger.



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

5.4. Performance of VNS on the VRP configuration

The proposed VNS finds results with a deviation never exceeding 1% for the instances solved by the MILP. 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 non-inserted jobs by the VNS (column "% non-inserted") and the average percentage gap of VNS with respect to the optimal values (column "% gap*"). % gap is computed as follows: $\frac{d\left(P^{|W^\star|}_{no\ walk}\right)-d^\star\left(P^{|W^\star|}_{no\ walk}\right)}{d^\star\left(P^{|W^\star|}_{no\ walk}\right)} \cdot 100$. Table 1 shows that VNS finds optimal solutions for 97% of the instances. 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	% non-inserted	% gap	% non-inserted	% gap	% non-inserted	% gap
20	0%	0%	0%	0%	0%	0%
30	0%	0%	0%	0%	0.3%	0%
40	0%	0%	0%	0.04%	0.2%	0.08%
50	0%	0%	0%	1.34%	0.4%	0.82%

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 by 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 4 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.

For the three configurations involving carpooling, Table 2 synthesizes the percentage of instances in each of the boxes displayed in Figure 4. It indicates that for 55% of the instances, the fleet size

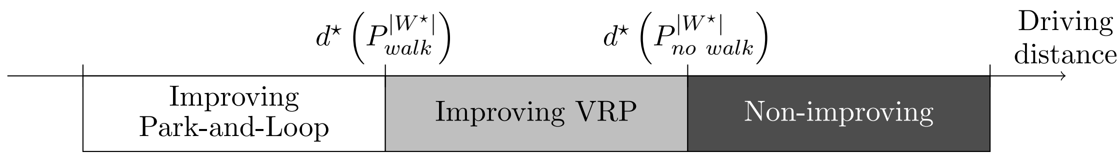


Figure 4: Potential driving distance of the solutions found feasible with less cars than workers.

can be reduced when only carpooling is allowed (configuration $\left(P_{no\ walk}^{|W^\star|-1}\right)$). Moreover, the total driving distance is smaller 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 on top of the implementation of carpooling (configuration $\left(P_{walk}^{|W^\star|-1}\right)$), the number of feasible instances 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, 7.5% of the instances remain feasible. This relatively low number can be explained by the fact that the generated instances require a maximum of 5 workers (average 3.3 workers) to be solved; thus, removing two cars is a drastic reduction.

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

Solution characteristics	$\left(P_{no\ walk}^{\lvert W^\star\rvert-1}\right)$	$\left(P_{walk}^{\lvert W^\star\rvert-1}\right)$	$\left(P_{walk}^{\lvert W^\star\rvert-2}\right)$
Improving *Park-and-Loop*	0.0%	6.7%	0.0%
Improving VRP	6.7%	19.2%	0.0%
Non-improving	48.3%	30.8%	7.5%
Total feasible	55.0%	56.7%	7.5%

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 4) is given in Table 3. The "VRP" columns reflect the characteristics of the VRP optimal solutions: $|W^\star|$, 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 inserted in the solution ($|J^{out}|$) for all configurations involving carpooling (n.b., the driving distance is not displayed for unfeasible solutions). For configuration $\left(P_{walk}^{|W|^\star-1}\right)$, it shows that the solution of instance 40_H_2 improves the one of *Park-and-Loop* (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 representative instances.

Instance	VRP $\left(P^{\lvert W^\star \rvert}_{no\ walk}\right)$				Park-and-Loop $\left(P^{\lvert W^\star \rvert}_{walk}\right)$	Carpooling					
			Idle Time			$\left(P^{\lvert W^\star \rvert -1}_{no\ walk}\right)$		$\left(P^{\lvert W^\star \rvert -1}_{walk}\right)$		$\left(P^{\lvert W^\star \rvert -2}_{walk}\right)$	
	$\lvert W^\star \rvert$	$d^\star$	Route	Depot	d	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$
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

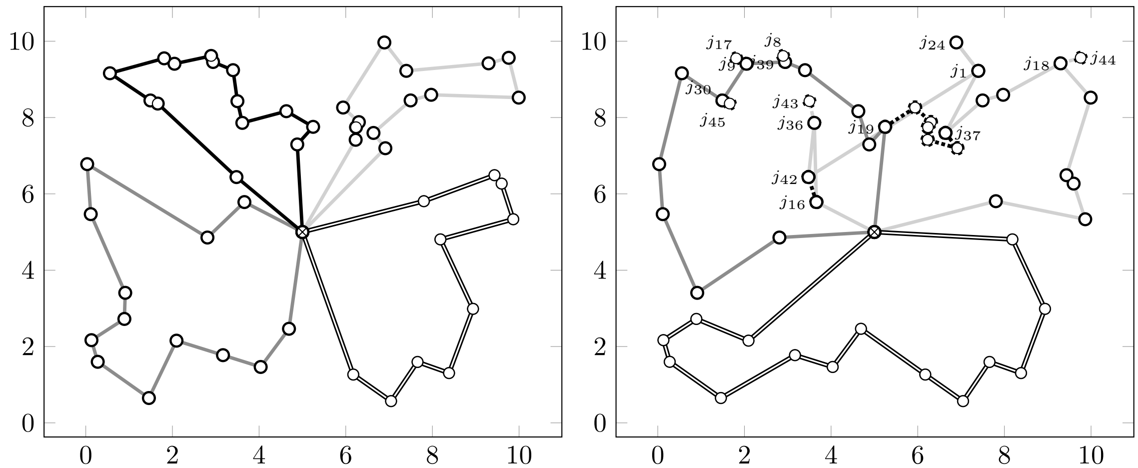
Figure 5 exemplifies a VRPTR solution (right part of the figure) for configuration $\left(P^{\lvert W^\star \rvert -1}_{walk}\right)$ for the instance 50_A_6, and compares it with the VRP optimal solution (left part of the figure). In this example, carpooling and walking allow for improving the VRP optimal 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.

6. Managerial insights

In Section 6.1, we position the obtained VRPTR solutions with respect to the existing industrial practices. Next, we highlight in Section 6.2 the instance characteristics that influence the gain obtained with the VRPTR formulation.

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*).



(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, w_4 returns to the depot with w_3.

Figure 5: Illustration of a VRPTR solution for which one car is saved and the total driving distance is reduced by 1.3%. The VRP optimal solution and a VRPTR solution are presented for the same instance.

6.1.1. Benefits of the Park-and-Loop

Table 4 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 VRP optimal 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 "$d\left(P^{|W^\star|}_{walk}\right)$", and the corresponding percentage gap with respect to the optimal VRP solution values in column "% gap" (computed as follows: $\frac{f\left(P^{|W^\star|}_{no\ walk}\right)-f\left(P^{|W^\star|}_{walk}\right)}{f\left(P^{|W^\star|}_{no\ walk}\right)} \cdot 100$).

Table 4 shows that a significant reduction in the driving distance is achieved when walking is allowed (average gain of 6.4% over the 120 instances). While the achieved gain decreases as the size of the time window gets narrower, it remains stable as the size of the instance gets larger. 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.

Table 4: Aggregated results for the *Park-and-Loop* configuration $\left(P^{|W^\star|}_{no\ walk}\right)$.

Instance	VRP			Park-and-Loop	
	Idle time	$\lvert W^\star\rvert$	$d^\star\left(P^{\lvert W^\star\rvert}_{no\ walk}\right)$	$d\left(P^{\lvert W^\star\rvert}_{walk}\right)$	%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%

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. $\left(S^{(dist)}\right)$ 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 5 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. 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 4. Column "%VRP" (resp. "%P&L") gives the percentage gap with respect to the VRP solution (resp. with the *Park-and-Loop* solution).

Table 5 shows that, while keeping the number of workers stable with respect to the VRP optimal 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

Table 5: 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%

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, and goes up 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 5 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 is as high as 8%.

From Table 5, we observe that a conflict exists between objectives f_{car} and f_{dist}, but also 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 vehicle fleet is still achieved for instances with *quarter day* time windows.

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.

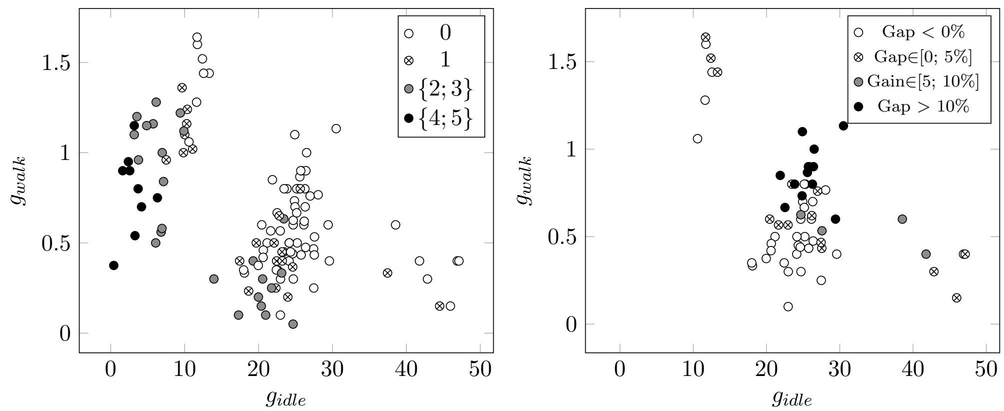
6.2.1. Idle time and walking potential

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 the job can be reached within the associated time window. More precisely, let J^R_i be the set of jobs reachable from job $i \in J$. Formally, we have $J^R_i = \left\{ j \in J \;\mid\; \tau^f_{ij} \leq \tau^f_M;\, e_i + p_i \cdot \tau_{ij} \leq l_j \right\}$ (n.b., this constraint indicates that when working on $i \in J$ as early as possible, a worker has enough time to serve the job and then walk to $j \in J$ before the end of the associated time window). Hence, $g_{walk} = \frac{\sum_{i \in J} |J^R_i|}{|J|}$. Second, the idle time, as introduced in Section 3.2, is denoted by g_{idle}. 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, since the density of jobs increases. Figure 6 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 particularly, for each instance, Figure 6(a) gives the number of jobs that cannot be inserted in $\left(P^{|W^\star|-1}_{walk}\right)$. Figure 6(b) plots the gap found with the VRP optimal 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 6(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 6(b) highlights that the gain increases with the walking potential g_{walk} of the instance. Moreover, for an idle time between 15% and 30%, g_{walk} must be greater than 0.5 jobs per km^2 to find competitive solutions regarding the driving distance.

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



(a) Distribution of non-inserted 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 non-inserted 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 6: 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).

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 travelling 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 averagely goes up from 23% to 29.2%, whereas the f_{dist}-gain jumps from 1.9% to 8.8%. Indeed, in such a situation, the VRP optimal 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.

7. Conclusion, perspectives and future works

This study considers a new type of VRP called the *Vehicle Routing Problem with Transportable Resources* (VRPTR), where heavy and light resources 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 where two distinct transportation modes are available, the heavy one being independent (i.e., it can move autonomously) and the light one being dependent (i.e., it needs to be transported by the heavy one for some parts of its route). Whereas the dependent transportation mode could take the form of drones, scooters, or bicycles, the independent transportation mode ranges from trucks and buses to cars. It is assumed here that the drivers are also eligible to serve jobs.

In this contribution, we treat an application of the VRPTR that comes from an industrial case. It involves cars (independent transportation mode), walking (dependent transportation mode), and carpooling. Coordinating such dependent and independent resources 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 for the total driving distance and for 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).

It is important to note that the VRPTR creates routes that are highly synchronized. Among the possible future works, we propose addressing the robustness of the newly introduced type of solutions by stressing them with random fluctuations of the traveling and service times, as studied in a VRP context in Lorini et al. (2011). In that respect, dynamic delays, such as traffic congestion or breakdowns, would have to be treated carefully since they could render the solution unfeasible (in terms of time window violations).

8. Acknowledgement

The authors would like to thank Professor Guy Desaulniers (Polytechnique Montreal and GERAD, Canada) for proving the optimality of our VRP solutions with GENCOL.

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. Technical report, 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.

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

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 fuer Energie (ifeu) und Umweldforschung Heidelberg GmbH, www.ecotransit.org/download/ecotransit_background_report.pdf.

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.

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.

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.

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.

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.

Toth, P. and Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics.

Wohlsen, M. (2014). The next big thing you missed: Amazon's delivery drones could work–they just need trucks. Technical report, www.wired.com/2014/06/the-nextbig-thing-you-missed-delivery-drones-launched-from-trucks-are-the-future-ofshipping.

Appendix A. Fastening the insertion heuristic

Masson et al. (2013) proposed a *fast feasibility check* (FFC) procedure for interdependent routes. It extends the *forward slack time* introduced by Savelsbergh (1992). In the VRPTR case, an additional modeling effort is required to take into account that all WRs depend on each other (the same worker transported through multiple WRs).

Appendix A.1. Modeling: aggregated nodes

For the vehicle routes, only the entry and exit points of a WR are relevant (as all intermediate nodes are not visited by the vehicles). Accordingly, we introduce two aggregated nodes for a WR: one for the drop-off (at the beginning of the WR) and one for the pick-up (at the end of the WR). These nodes are visited by the vehicles and gather consolidated information about the WR (total duration and resulting time window).

Let $\Omega = \bigcup_{w \in W} \Omega_w$ be the set of all WRs, where Ω_w represents the ordered set of WRs performed by worker $w \in W$ in his/her schedule ($\kappa_{i+1} \in \Omega_w$ is directly performed after WR κ_i in the schedule of worker w, $\forall i < |\Omega_w|$). $w(\kappa)$ is the worker associated with κ, and $i(\kappa)$ is the position of WR κ in the worker's planning. Furthermore, $\rho(\kappa)$ (resp. $\sigma(\kappa)$) represents the predecessor (resp. successor) of κ in the corresponding worker planning. $\rho(\kappa)$ (resp. $\sigma(\kappa)$) is $\emptyset$ if κ is the first (resp. the last) WR done. $[e_\kappa, l_\kappa]$ and p_κ denotes the time window of κ (to serve all jobs of κ on time) and the total processing time (i.e., all processing times, walking times, and waiting times along κ).

$\mathcal{D}$ (resp. $\mathcal{P}$) is the set of all drop-off (resp. pick-up) points. $D_\kappa \in \mathcal{D}$ (resp. $P_\kappa \in \mathcal{P}$) is the drop-off (resp. pick-up) point of κ. $O_W \subset \mathcal{P}$ (resp. $O'_W \subset \mathcal{D}$) represents the set of worker pick-up (resp. drop-off) points at the depot. Moreover, O_K (resp. O'_K) denotes the set of the first (resp. the last) nodes visited by the vehicles. Finally, $\mathcal{M} = \mathcal{P} \cup \mathcal{D} \cup O_K \cup O'_K$ denotes the set of all aggregated nodes for a given solution.

A transportation request arises between the end of a WR κ and the beginning of the next WR $\sigma(\kappa)$, denoted by $(P_\kappa, D_{\sigma(\kappa)})$. Furthermore, transportation requests are required between the pick-up at the depot and the drop-off at the beginning of the first WR, as well as between the pick-up in the last WR (of any worker's planning) and the final drop-off at the depot: (P_{O_w}, D_{κ_1}) (resp. $(P_{\kappa_{|\omega_w|}}, D_{0'_w})$) for the transportation between the depot (resp. last WR $\kappa_{|\omega_w|}$) and the first WR κ_1 (resp. the depot).

For each $v \in \mathcal{M}$, $k(v)$ is the route that visits v, and $i(v)$ is the position of v in the route. $\rho(v)$ (resp. $\sigma(v)$) denotes the predecessor (resp. successor) of v in the route. Finally, for each $v \in \mathcal{D} \cup \mathcal{P}$, $\kappa(v)$ denotes the WR that contains v. Workers' pick-up and drop-off times are set to be null. Each $v \in \mathcal{M}$ can be characterized by an associated time window $[e_v, l_v]$, which corresponds to the time a car must drop off a worker to have an on-time arrival for the jobs composing $\kappa(v)$. For each $v \in \mathcal{D}$, we have $e_v = e_{\kappa(v)}$ and $l_v = l_{\kappa(v)}$, whereas for each $v \in \mathcal{P}$, we have $l_v = \infty$, and e_v depends on the drop-off time at $D(v)$.

Appendix A.2. Vehicle constraints

A vehicle route is an ordered set of aggregated nodes that must satisfy the following constraints, where $D(v)_{v \in \mathcal{P} \cup O_W}$ (resp. $P(v)_{v \in \mathcal{D} \cup 0'_W}$) designates the drop-off (resp. pick-up) in the pick-up and drop-off couple. More precisely, $D(v) = P_{\rho(\kappa(v))}$, $\forall v \in \mathcal{P}$, and $D(v) = P_{0,w(\kappa(v))}$, $\forall v \in O_W$. $\mathbb{1}_{i=P} = 1$ (resp. $\mathbb{1}_{i=D} = 1$) if i is a pick-up (resp. drop-off) aggregated node, and 0 otherwise. Constraints (A.1) ensure that the nodes of a pick-up and drop-off couple are managed by the same vehicle, and the pick-up must happen before the drop-off. Constraints (A.2) ensure that a vehicle cannot move without its associated driver by scheduling the driver's pick-up directly after his/her drop-off in the vehicle route. Constraints (A.3) ensure that the vehicle capacity q is never exceeded. A set of routes is feasible if the above constraints are satisfied, and if it fulfills the temporal constraints that are detailed in the next subsection.

$$k(v) = k(D(v)) \text{ and } i(v) < i(D(v)) \qquad \forall v \in O_W \cup \mathcal{P} \tag{A.1}$$

$$k(D_{\kappa(v)}) = k(P_{\kappa(v)}) \text{ and } i(P_{\kappa(v)}) = i(D_{\kappa(v)}) + 1 \qquad \forall v \in \mathcal{D}/\ w(\kappa(v)) \text{ is a driver} \tag{A.2}$$

$$\sum_{v \in k} (\mathbb{1}_{v=P} - \mathbb{1}_{v=D}) \leq q \qquad \forall k \in K \tag{A.3}$$

Appendix A.3. Temporal constraints

A solution to the VRPTW is feasible if and only if each of its routes satisfies temporal feasibility. The temporal constraints are modeled using a *Simple Temporal Problem* (as described by Dechter et al. (1991)), for which efficient algorithms and representations exist in the literature. Temporal constraints are expressed as follows in Equations (A.4)–(A.6), where h_v represents the service time at the aggregated node $v \in \mathcal{M}$:

$$h_{\sigma(v)} \geq h_v + \tau^d_{v,\sigma(v)}, \qquad \forall v \in \mathcal{M} \backslash O'_K \tag{A.4}$$

$$h_{P_\kappa} \geq \max\{h_{D_\kappa}, e_{D_\kappa}\} + p_\kappa, \qquad \forall \kappa \in \Omega \tag{A.5}$$

$$h_v \leq l_v, \qquad \forall v \in \mathcal{M} \tag{A.6}$$

Equations (A.4) set the temporal constraints in a route, for which the arrival time at a node depends on the departure time at the previous node. Equations (A.5) specify the time at which a worker is available to be picked up after completing a WR. The time at which a worker starts working on a WR depends on both the drop-off time h_{D_κ} and on the time window e_{D_κ} of the WR. Finally, Equations (A.6) state that the service time cannot start after the end of the corresponding time window.

This set of equations can be modeled with a precedence graph, called G^p, where constraints of type $h_u - h_v \geq a_{uv}$ (a_{uv} is a real number) represent an arc from u to v with a cost of a_{uv}. Node o is introduced to represent the beginning of the planning horizon, and, for every drop-off point $D \in \mathcal{D}$, a virtual node $D^{(dup)}$ is introduced to get rid of the max function in Equations (A.5). $\mathcal{D}^{(dup)}$ is the set of duplicated nodes. Equations (A.4) to (A.6) can therefore be rewritten as follows:

$$h_{\sigma(v)} - h_v \geq \tau^d_{v,\sigma(v)}, \qquad \forall v \in \mathcal{M} \backslash O'_K \tag{A.7}$$

$$h_{P_\kappa} - h_{D^{(dup)}_\kappa} \geq p_\kappa, \qquad \forall \kappa \in \Omega \tag{A.8}$$

$$h_{D^{(dup)}_\kappa} - h_{D_\kappa} \geq 0, \qquad \forall \kappa \in \Omega \tag{A.9}$$

$$h_{D^{(dup)}_\kappa} - h_o \geq e_v, \qquad \forall \kappa \in \Omega \tag{A.10}$$

$$h_o - h_v \geq -l_v, \qquad \forall v \in \mathcal{M} \tag{A.11}$$

$$h_v \geq 0, \qquad \forall \kappa \in \Omega \tag{A.12}$$

$$h_o = 0 \tag{A.13}$$

Checking the feasibility of the VRPTR set of temporal constraints is equivalent to showing that there is no cycle of negative length in the precedence graph. This can be done using the so-called BFCT algorithm, which has a complexity of $\mathcal{O}(|\mathcal{M}| \times |A'|)$ (Cherkassky et al. 2009). For any solution satisfying the temporal constraints, the precedence graph is a direct acyclic graph.

Figure A.7 presents the precedence graph associated with Figure 1 using the above-introduced notation. In Figure 1, the solution with carpooling and walking contains three WRs, which can be denoted as $\Omega = \{\kappa_1 = \{j_1\}, \kappa_2 = \{j_2, j_3\}, \kappa_3 = \{j_4\}\}$. It involves six pick-up and drop-off couples denoted as $(P_{O_{w_1}}, D_{\kappa_1})$, $(P_{\kappa_1}, D_{0'_{w_1}})$, $(P_{O_{w_2}}, D_{\kappa_2})$, $(P_{\kappa_2}, D_{0'_{w_2}})$, $(P_{O_{w_3}}, D_{\kappa_3})$ and $(P_{\kappa_3}, D_{0'_{w_3}})$.

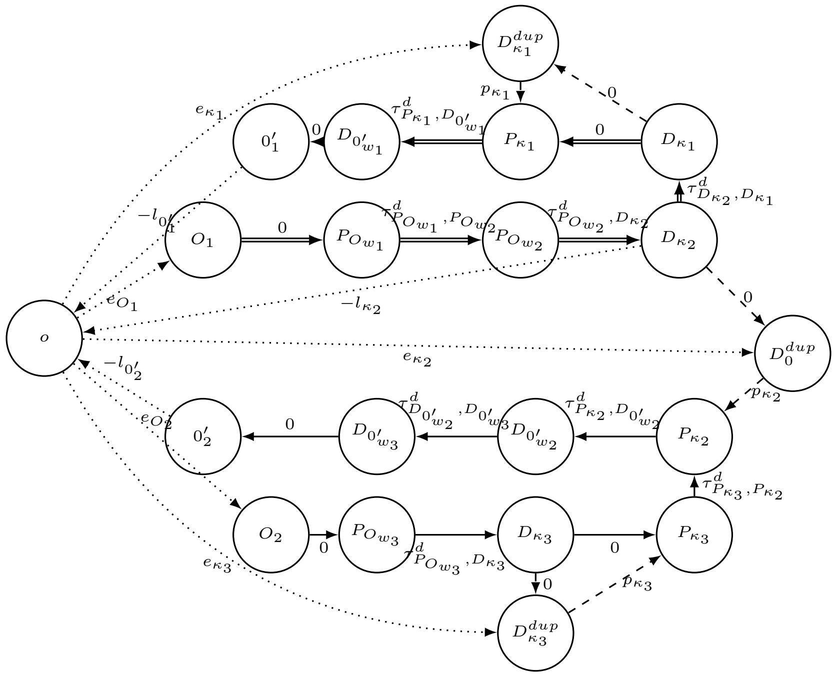


Figure A.7: Precedence graph representing the VRPTR solution of Figure 1. Dotted arcs represent time window constraints (for the sake of clarity, not all time window constraints are drawn), dashed arcs represent precedence constraints due to WRs, both double and normal arcs represent precedence constraints due to the routes. The order of the nodes in the route must satisfy the constraints in Equations (A.1)–(A.3).

Appendix A.4. Insertion positions

All the insertion positions are tested in order to find the cheapest one. For the VRPTW, the number of different insertion positions increases in $\mathcal{O}(n)$, but this number increases in $\mathcal{O}(n^5)$ in the VRPTR case. More precisely, in all possible insertion positions, job j can be inserted as a single WR, denoted here as $\kappa_j = \{j\}$, which is then inserted between two WRs, denoted here as κ_i and κ_{i+1}. Hence, there is a maximum number of n positions, between two existing WRs, that can be tested. As a consequence of this insertion, two new transportation requests must be inserted in the vehicle routes: one from the end of κ_i (node P_{κ_i}) to the beginning of κ_j (node $D_{\{j\}}$), and one from the end of κ_j (node $P_{\{j\}}$) to the beginning of κ_{i+1} (node $D_{\kappa_{i+1}}$). Four nodes must therefore be inserted in the solution, which correspond to $\mathcal{O}(n^4)$ insertion positions to test. As there are n potential insertions between two WRs, the total number of required tests grows to $\mathcal{O}(n^5)$.

Note that even removing a job from a VRPTR solution might require the insertion of a transportation request. Indeed, in passenger planning, when removing a job that stands alone in a

WR κ, a new transportation request from $P(D_\kappa)$ to $D(P_\kappa)$ has to be satisfied, replacing the two obsolete transportation requests $(P(D_\kappa), D_\kappa)$ and $(P_\kappa, D(P_\kappa))$. Therefore, a removal procedure in the VRPTR context has a complexity of $\mathcal{O}(n^2)$, whereas it can be done in constant time in the VRP case.

Appendix A.5. Feasibility of an insertion

Algorithm 2 tests in constant time if an insertion position violates the temporal constraints exposed in Appendix A.3. It extends the algorithm proposed by Masson et al. (2013). More precisely, it adds a specific feasibility check that corresponds to the precedence constraints arising between the WRs performed by the same worker. As stated above, when inserting a job, four nodes have to be simultaneously inserted in the graph representation. The FFC procedure pre-computes, for each aggregated node $v \in \mathcal{M}$, the earliest service time (h_v), the latest departure time (λ_v), and the matrix of waiting times between all aggregated nodes ($(ST_{uv})_{u,v \in \mathcal{M}}$). After inserting any of the four nodes, the delay at any other node can be computed in constant time (the delay after the insertion is reduced by the smallest waiting time between the two nodes). If the delay does not exceed the latest departure time of the successor, all jobs will still be served within their time window, and the other nodes can be tested for insertion. Otherwise, the insertion position is detected to be unfeasible. The four nodes to be inserted, called here (P_1, D_1) and (P_2, D_2), are inserted respectively in routes k_1 and k_2, and positions (i_1, j_1) and (i_2, j_2) are evaluated. h_v (resp. λ_v) is the length of the longest path from node 0 to v (resp. v to 0) in the precedence graph. As the precedence graph G^p represents a feasible solution, it does not contain any cycle of positive weight; therefore, the longest path is the shortest path in $-G^p$, where the arcs of $-G^p$ have the opposite weight of the arcs in G^p. The precedence graph is a direct acyclic graph where the shortest paths can be computed in linear time. $(ST_{uv})_{u,v \in \mathcal{M}}$ is the matrix of the shortest paths in the precedence graph, where the arcs are weighted with the waiting time in the solution at the terminal node of the arc (i.e., the waiting time at node v is equal to $\max\{0, e_v - h_v\}$), and it can be computed in $\mathcal{O}(n^2)$. All these shortest paths are computed once, and then the $\mathcal{O}(n^4)$ insertion positions are tested in constant time.

Algorithm 2 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 $\bar{h}_{P_1} = \max\{h_{P_1}, h_{i_1} + \tau^d_{i1,P_1}\}$, and return FALSE if $\bar{h}_{P_1} > l_{P_1}$
- set $\bar{h}_{\sigma(i1)} = h_{P_1} + \tau^d_{P_1,\sigma(i_1)}$, and return FALSE if $\bar{h}_{\sigma(i1)} > \lambda_{\sigma(i_1)}$
- set $\delta_{\sigma(i_1)} = \max\{\bar{h}_{\sigma(i_1)} - h_{\sigma(i_1)}, 0\}$

Evaluate the insertion of D_1

- set $\bar{h}_{j_1} = h_{j_1} + \max\{\delta_{\sigma(i_1)} - ST_{\sigma(i_1),j_1}, 0\}$ and set $\bar{h}_{D_1} = \max\{h_{j_1} + \tau^d_{j_1,D_1}, E_{D_1}\}$, and return FALSE if $\bar{h}_{D_1} > l_{D_1}$
- set $\bar{h}_{\sigma(j_1)} = a_{D_1} + \tau^d_{D_1,\sigma(j_1)}$, and return FALSE if $\bar{h}_{\sigma(j_1)} > \lambda_{\sigma(j_1)}$
- set $\delta_{\sigma(j_1)} = \max\{\bar{h}_{\sigma(j_1)} - h_{\sigma(j_1)}, 0\}$

Evaluate the insertion of P_2

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

Evaluate the insertion of D_2

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

Return TRUE

Appendix B. Detailed results for all instances

Tables B.6 to B.9 detail the results found by VNS for all VRPTR configurations over the 120 generated instances. The table contents correspond to those of Table 3, a description of which can be found in Section 5.5.2.

Table B.6: Detailed results for instances involving 20 jobs.

Instance	VRP $\left(P^{\lvert W^\star \rvert}_{no\ walk}\right)$				Park-and-Loop	Carpooling					
			Idle Time		$\left(P^{\lvert W^\star \rvert}_{walk}\right)$	$\left(P^{\lvert W^\star \rvert -1}_{no\ walk}\right)$		$\left(P^{\lvert W^\star \rvert -1}_{walk}\right)$		$\left(P^{\lvert W^\star \rvert -2}_{walk}\right)$	
	$\lvert W^\star \rvert$	$d^\star$	Route	Depot	d	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$
20_A_1	2	37.8	0.0%	26.1%	32.4	47.2	0	40.3	0	0	6
20_A_2	2	46.2	0.0%	24.1%	41.6	59.0	0	50.0	0	0	6
20_A_3	2	43.2	0.0%	29.6%	39.4	48.8	0	45.6	0	0	6
20_A_4	2	42.0	0.0%	26.3%	39.1	48.7	0	46.6	0	0	6
20_A_5	2	41.5	0.0%	25.2%	35.3	48.4	0	49.6	0	0	7
20_A_6	2	46.0	0.0%	23.0%	43.8	64.9	0	69.0	0	0	8
20_A_7	2	44.3	0.0%	24.0%	40.1	66.1	0	66.1	0	0	7
20_A_8	2	42.3	0.0%	19.7%	39.1	55.4	0	55.4	0	0	7
20_A_9	2	33.6	0.0%	25.3%	29.9	40.0	0	38.0	0	0	6
20_A_10	2	43.8	0.0%	24.7%	41.2	49.0	0	48.2	0	0	6
20_H_1	2	49.0	16.0%	7.5%	41.4	-	1	-	1	0	7
20_H_2	2	53.6	5.0%	17.4%	52.2	-	1	-	1	0	7
20_H_3	2	52.1	14.8%	12.7%	51.4	77.8	0	66.7	0	0	8
20_H_4	2	51.1	13.5%	10.7%	49.4	-	1	-	1	0	9
20_H_5	2	52.6	8.4%	14.2%	47.6	80.1	0	80.1	0	0	7
20_H_6	2	51.1	16.4%	5.4%	50.3	-	1	-	1	0	9
20_H_7	2	59.4	11.4%	9.0%	55.3	-	2	-	2	0	8
20_H_8	2	51.7	3.5%	13.9%	46.6	-	2	-	1	0	8
20_H_9	2	44.2	13.0%	9.8%	39.8	-	1	-	1	0	7
20_H_10	2	54.6	14.6%	7.5%	52.0	-	1	-	1	0	7
20_Q_1	2	61.2	15.0%	5.5%	60.1	-	3	-	2	0	9
20_Q_2	2	59.4	6.1%	14.9%	58.0	-	3	-	3	0	7
20_Q_3	2	63.9	23.3%	1.3%	63.9	-	2	-	2	0	9
20_Q_4	2	68.6	15.9%	4.1%	67.0	-	3		3	0	11
20_Q_5	2	74.9	14.4%	2.9%	71.0	-	2	-	2	0	10
20_Q_6	3	72.1	40.6%	3.9%	70.5	-	1	-	1	-	1
20_Q_7	3	65.4	38.0%	8.0%	60.8	70.6	0	64.9	0	-	1
20_Q_8	2	66.4	3.5%	10.5%	63.1	-	3	-	2	-	9
20_Q_9	3	54.6	40.5%	6.4%	50.7	66.0	0	54.8	0	-	1
20_Q_10	3	60.7	37.5%	9.6%	59.8	61.6	0	60.3	0	-	0

Table B.7: Detailed results for instances involving 30 jobs.

Instance	VRP $\left(P^{\lvert W^\star \rvert}_{no\ walk}\right)$				Park-and-Loop	Carpooling					
			Idle Time		$\left(P^{\lvert W^\star \rvert}_{walk}\right)$	$\left(P^{\lvert W^\star \rvert-1}_{no\ walk}\right)$		$\left(P^{\lvert W^\star \rvert-1}_{walk}\right)$		$\left(P^{\lvert W^\star \rvert-2}_{walk}\right)$	
	$\lvert W^\star \rvert$	$d^\star$	Route	Depot	d	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$
30_A_1	3	48.8	0.0%	25.6%	41.3	49.5	0	43.0	0	-	2
30_A_2	3	53.0	0.0%	26.5%	46.6	53.4	0	46.8	0	-	1
30_A_3	3	57.1	0.0%	27.4%	51.0	54.3	0	54.3	0	-	2
30_A_4	3	52.8	0.0%	30.5%	46.3	49.8	0	45.8	0	-	1
30_A_5	3	55.9	0.0%	24.9%	48.4	54.3	0	44.8	0	-	2
30_A_6	3	57.3	0.0%	23.8%	51.7	54.0	0	49.5	0	-	2
30_A_7	3	54.7	0.0%	29.4%	45.5	53.3	0	46.7	0	-	1
30_A_8	3	50.6	0.0%	22.5%	45.4	52.9	0	44.6	0	-	3
30_A_9	3	49.5	0.0%	26.3%	41.9	48.1	0	43.0	0	-	1
30_A_10	3	53.8	0.0%	27.6%	49.5	54.5	0	49.9	0	-	1
30_H_1	3	62.5	20.2%	3.3%	57.6	-	2	-	2	-	-
30_H_2	3	61.5	4.4%	20.8%	57.2	78.3	0	65.3	0	-	1
30_H_3	3	74.3	14.0%	10.8%	67.6	84.5	0	75.5	0	-	4
30_H_4	3	68.2	19.8%	8.3%	63.7	88.0	0	71.2	0	-	4
30_H_5	3	68.0	5.2%	17.8%	61.1	79.4	0	67.9	0	-	3
30_H_6	3	71.1	9.3%	12.3%	65.0	80.6	0	70.1	0	-	3
30_H_7	3	66.5	12.1%	15.4%	58.1	72.6	0	64.7	0	-	3
30_H_8	3	59.2	8.1%	13.1%	52.7	76.3	0	65.0	0	-	4
30_H_9	3	56.5	10.0%	15.2%	50.7	64.3	0	59.5	0	-	3
30_H_10	3	65.2	14.9%	10.8%	61.3	75.1	0	67.8	0	-	3
30_Q_1	3	88.6	18.5%	0.7%	87.9	-	3	-	3	-	-
30_Q_2	3	73.8	8.3%	14.9%	68.7	-	1	-	1	-	-
30_Q_3	3	85.4	18.1%	4.9%	81.6	115.4	0	115.4	0	-	7
30_Q_4	3	89.9	18.9%	5.8%	86.3	-	1	-	1	-	-
30_Q_5	3	94.9	14.1%	4.6%	90.6	-	2	-	1	-	-
30_Q_6	4	102.9	35.5%	2.0%	97.7	-	1	-	1	-	-
30_Q_7	3	94.3	19.8%	3.3%	94.3	-	3	-	3	-	-
30_Q_8	3	78.6	13.1%	5.0%	77.7	110.4	0	104.7	0	-	4
30_Q_9	4	74.1	32.2%	9.6%	68.6	77.1	0	67.9	0	78.5	0
30_Q_10	4	77.5	37.1%	5.8%	74.3	78.6	0	74.8	0	89.1	0

Table B.8: Detailed results for instances involving 40 jobs.

Instance	VRP $\left(P^{\lvert W^\star \rvert}_{no\ walk}\right)$				Park-and-Loop	Carpooling					
			Idle Time		$\left(P^{\lvert W^\star \rvert}_{walk}\right)$	$\left(P^{\lvert W^\star \rvert-1}_{no\ walk}\right)$		$\left(P^{\lvert W^\star \rvert-1}_{walk}\right)$		$\left(P^{\lvert W^\star \rvert-2}_{walk}\right)$	
	$\lvert W^\star \rvert$	$d^\star$	Route	Depot	d	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$
40_A_1	3	60.0	0.0%	2.4%	59.2	-	4	-	4	-	-
40_A_2	3	59.9	0.0%	3.2%	58.5	-	4	-	4	-	-
40_A_3	3	63.8	0.0%	2.6%	63.2	-	3	-	3	-	-
40_A_4	3	56.9	0.0%	10.0%	50.4	-	2	-	1	-	-
40_A_5	4	63.3	0.0%	24.9%	52.8	61.0	0	53.3	0	60.3	0
40_A_6	3	62.7	0.0%	3.5%	62.7	-	4	-	3	-	-
40_A_7	3	58.6	0.0%	7.0%	57.8	-	2	-	2	-	-
40_A_8	4	66.2	0.0%	21.9%	56.4	65.0	0	53.8	0	68	0
40_A_9	3	57.5	0.0%	3.2%	56.5	-	3	-	3	-	-
40_A_10	3	59.3	0.0%	4.9%	57.7	-	3	-	3	-	-
40_H_1	4	75.5	19.6%	5.3%	68.4	90.3	0	81.1	0	-	3
40_H_2	4	73.1	0.0%	25.8%	63.9	75.1	0	62.1	0	94.6	0
40_H_3	4	82.6	14.1%	10.6%	71.5	85.3	0	76.0	0	-	1
40_H_4	3	80.2	3.1%	3.2%	78.1	-	4	-	4	-	-
40_H_5	4	75.4	8.9%	14.5%	65.9	81.0	0	72.3	0	98.7	0
40_H_6	4	73.1	13.7%	12.7%	63.6	75.4	0	63.5	0	87.9	0
40_H_7	3	76.3	2.8%	1.3%	76.3	-	4	-	4	-	-
40_H_8	4	78.3	5.2%	15.3%	69.1	85.6	0	75.2	0	-	3
40_H_9	4	71.5	11.3%	14.4%	57.9	72.0	0	64.1	0	84.1	0
40_H_10	3	79.8	0.0%	1.6%	79.8	-	5	-	5	-	-
40_Q_1	4	96.5	21.1%	1.3%	90.8	109.0	1	108.8	1	-	-
40_Q_2	4	85.6	4.0%	20.3%	80.2	98.2	1	88.6	0	101.7	3
40_Q_3	4	101.7	15.1%	7.3%	97.9	116.4	0	106.3	0	114.1	3
40_Q_4	3	117.4	0.0%	0.4%	117.4	103.6	6	98.5	5	-	-
40_Q_5	4	104.8	15.7%	4.3%	96.5	123.0	0	106.3	0	114.9	3
40_Q_6	5	100.3	33.7%	4.9%	92.0	103.9	0	94.1	0	103	0
40_Q_7	4	91.0	23.2%	3.2%	84.4	108.8	0	97.9	0	93.8	3
40_Q_8	4	98.9	14.6%	3.3%	96.9	114.8	0	104.4	0	71.9	3
40_Q_9	4	92.8	16.8%	6.4%	87.1	90.3	2	87.7	1	-	-
40_Q_10	4	97.1	19.5%	4.7%	88.6	106.5	0	98.6	0	101.3	3

Table B.9: Detailed results for instances involving 50 jobs.

Instance	VRP $\left(P^{\lvert W^\star \rvert}_{no\ walk}\right)$				Park-and-Loop	Carpooling					
			Idle Time		$\left(P^{\lvert W^\star \rvert}_{walk}\right)$	$\left(P^{\lvert W^\star \rvert -1}_{no\ walk}\right)$		$\left(P^{\lvert W^\star \rvert -1}_{walk}\right)$		$\left(P^{\lvert W^\star \rvert -2}_{walk}\right)$	
	$\lvert W^\star \rvert$	$d^\star$	Route	Depot	d	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$	d	$\lvert J^{out} \rvert$
50_A_1	4	65.4	0.0%	10.4%	60.5	74.4	0	74.4	0	-	-
50_A_2	4	69.0	0.0%	11.7%	62.5	87.9	0	72.1	0	-	4
50_A_3	4	71.4	0.0%	9.7%	65.9	-	1	-	1	-	-
50_A_4	4	68.1	0.0%	13.3%	58.2	85.8	0	68.0	0	-	4
50_A_5	4	68.3	0.0%	6.2%	64.2	-	1	-	1	-	-
50_A_6	4	71.8	0.0%	11.7%	66.1	78.7	0	70.9	0	-	4
50_A_7	4	71.9	0.0%	12.6%	63.7	78.5	0	74.7	0	-	3
50_A_8	4	71.7	0.0%	5.7%	70.9	-	2	-	2	-	-
50_A_9	4	67.4	0.0%	11.6%	58.2	76.4	0	70.4	0	-	4
50_A_10	4	68.3	0.0%	12.4%	60.8	76.3	0	66.1	0	-	5
50_H_1	4	92.5	6.4%	0.8%	92.5	-	3	-	3	-	-
50_H_2	4	81.1	0.0%	10.3%	75.4	-	2	-	1	-	-
50_H_3	4	89.4	2.0%	5.5%	84.8	-	2	-	1	-	-
50_H_4	4	86.9	2.4%	8.7%	82.3	106.8	0	106.8	0	-	5
50_H_5	4	88.4	0.0%	3.7%	87.9	-	3	-	3	-	-
50_H_6	4	90.9	8.2%	1.2%	88.4	-	2	-	2	-	-
50_H_7	4	88.5	4.5%	6.1%	82.6	-	1	97.5	0	-	6
50_H_8	4	88.8	1.7%	2.0%	85.9	-	4	-	4	-	-
50_H_9	4	82.4	3.2%	6.7%	74.2	-	1	-	1	-	-
50_H_10	4	89.3	7.8%	2.1%	84.8	-	2	-	2	-	-
50_Q_1	5	104.1	19.6%	5.0%	95.5	117.5	0	107.4	0	109.7	1
50_Q_2	4	110.2	1.3%	5.5%	110.2	-	4	-	3	-	-
50_Q_3	4	124.9	1.0%	2.3%	124.9	-	5	-	4	-	-
50_Q_4	4	121.7	3.2%	3.7%	121.7	-	4	-	2	-	-
50_Q_5	5	113.2	17.9%	2.7%	104.4	130.7	0	117.0	0	116.4	2
50_Q_6	5	110.7	23.1%	2.5%	106.4	-	1	-	1	-	-
50_Q_7	5	104.3	22.1%	4.9%	96.1	111.0	0	101.4	0	106.5	2
50_Q_8	5	113.5	18.7%	1.9%	111.5	132.0	0	117.3	0	147.1	1
50_Q_9	4	113.9	3.3%	2.8%	111.6	-	3	-	3	-	-
50_Q_10	5	107.6	21.3%	4.9%	97.5	112.3	0	103.3	0	126.2	1