A Dial-A-Ride Problem with private vehicles and privacy settings

D. Brevet[1], C. Duhamel[1], M. Iori[2], P. Lacomme[1, 2]

[1] Université Clermont-Auvergne, Laboratoire d'Informatique (LIMOS) UMR CNRS 6158, Aubière, France, david.brevet@uca.fr, {christophe.duhamel, placomme}@isima.fr

[2] University of Modena and Reggio Emilia, Department of Sciences and Methods for Engineering, via Amendola 2, 42122 Reggio Emilia, Italy, manuel.iori@unimore.it

ARTICLE INFO

Article history:
Submitted

Keywords:
DARP, privacy, linear model, metaheuristic, ELS

ABSTRACT

This paper addresses the Dial-A-Ride Problem (DARP) using Private Vehicles and Alternative Nodes (DARP-PV-AN). The DARP consists of creating vehicle routes in order to ensure a set of users' transportation requests. Each request corresponds to a client needing to be transported from his/her origin to his/her destination. Routing costs have to be minimized while respecting a set of constraints like time windows and maximum route length. In the classical DARP, vehicles have to start from a depot and come back to it at the end of their route. In the DARP-PV-AN, the on-demand transportation service can be done either by a public fleet or by clients, using their vehicle (private vehicles). The use of these vehicles adds more flexibility and unclog the public transportation fleet by allowing clients to organize their own transportation. However, it also raises some privacy concerns. The DARP-PV-AN addresses these concerns and focuses on location privacy, i.e. the ability to prevent third parties from learning clients' locations, by keeping both original and final location private. This is addressed by setting several pickup/delivery nodes for the transportation requests, thus masking the private address. A compact mixed integer linear model is presented and an Evolutionary Local Search (ELS) is proposed to compute solutions of good quality for the problem. These methods are benchmarked on a modified set of benchmark instances. A new set of realistic instances is also provided to test the ELS in a more realistic way.

1. Introduction

In our work, we solve a particular dial-a-ride problem where passengers are transported by either a public fleet of vehicles or by means of private vehicles owned by other passengers. Clients that aim at hiding information on their home address can communicate a set of alternative locations where pickup/delivery can be made. The decision problem requires to determine which vehicle is assigned to each client, while satisfying operational constraints such as time windows, maximal route length, maximum riding time, and vehicle capacities. The problem models a large variety of real-world situations, from the standard dial-a-ride, where only the public fleet is given, to the pure car-pooling situation, where only private vehicles can be used. The aim of this work is to formally introduce and model the problem, and to present an effective evolutionary heuristic for its solution.

2. 1.2 Dial-A-Ride Problem

The problem considered here is related to the Vehicle Routing Problem (VRP) class. The purpose of these problems is to meet the demand of a set of clients by using a fleet of vehicles located at a central depot. Each vehicle performs a route that visits a sequence of clients before returning to the depot, and the aim is to serve all clients with the least-cost set of routes. The Capacitated VRP (CVRP) is the most famous problem in the VRP class, and imposes a maximum capacity on the demand loaded on each vehicle. Exact methods can consistently solve CVRP instances with up to 200 clients only (Baldacci et al., 2012), (Pecin et al., 2017), and hence heuristics are required to handle larger instances. Among the metaheuristics, the evolutionary algorithms have obtained good results (Prins, 2004), (Vidal et al., 2012).

The Dial-A-Ride Problem (DARP) is an extension of the CVRP where the clients' requests do not correspond to deliveries (or pickups) anymore. Instead, each client requires a transportation from an origin node (pickup) to a destination node (delivery) addressing additional constraints including time windows, vehicle maximum riding time and fleet size. A transportation request must be ensured by a single vehicle. In addition, the DARP can be defined in a static or in a dynamic context. In the former, all requests are known in advance; in the latter, requests appear dynamically, while vehicles are already performing their trip. We consider here the static version.

Heuristic approaches have been proposed in the 80s and the 90s. In the last decade, several more efficient approaches, based on metaheuristics, have been proposed. A tabu search has been proposed in (Cordeau and Laporte, 2003). Later, a Variable Neighborhood Search (VNS) has been presented in (Parragh *et al.*, 2010). (Masson *et al.*, 2014) used an Adaptive Large Neighborhood Search (ALNS) to address the DARP with transfers. (Chassaing *et al.*, 2016) adapt the Evolutionary Local Search (ELS) and recently (Masmoudi *et al.*, 2017) propose a Hybrid Genetic Algorithm (HGA). For further details, we refer the reader to the recent surveys on pickup and delivery problems for the transportation of goods (Battarra et al., 2014) and of people (Doerner and Salazar Gonzalez, 2014).

Formally, the DARP is defined on a complete directed graph $\mathrm{G} = (\mathrm{N}, \mathrm{A})$, with a heterogeneous fleet F of K vehicles and a set $\mathrm{R} = \{1, .., \mathrm{n}\}$ of transportation requests. $\mathrm{N} = \{0,1, \dots, 2\mathrm{n}, 2\mathrm{n}+1\}$ is the set of nodes. The depot is split into two copies, nodes 0 and $2\mathrm{n}+1$, for, respectively, the beginning and the end of the trips. Given a transportation request i, its pickup node is node i and its delivery node is $n+i$. Thus $\mathrm{N}^{\mathrm{P}} = \{1, \dots, \mathrm{n}\}$ and $N^D = \{\mathrm{n}+1, \dots, 2\mathrm{n}\}$ are, respectively, the pickup and the delivery subsets. For each node i, $[e_i; l_i]$ is its time windows (e_i is the earliest starting time and l_i is the latest starting time), the service duration is d_i and the demand in persons is q_i, such that $q_i > 0$ and $q_{n+i} = -q_i$. Given an arc $(i, j) \in A$, t_{ij} is the transportation time and c_{ij} is the transportation cost. Vehicle k of the fleet has capacity Q_k. For sake of simplicity, the node associated to a pickup node, resp. a delivery node, will be referred to as pickup node, resp. delivery node.

Following the notation by (Cordeau and Laporte, 2003), five types of variables can used to illustrate the time aspects of the DARP. These are illustrated in Fig.1 and can be described as follows for a given a node i:

- A_i is the arrival time of a vehicle,
- B_i is the beginning of the service,
- D_i is the departure time, such that $D_i = B_i + d_i$,
- W_i is the waiting time, such that $W_i = B_i - A_i$,
- R_i is the riding time which corresponds to the time between the end of service at pickup node i and the beginning of service at delivery node $(n+i)$ of a client. Thus $R_i = B_{n+i} - D_i$.

A route can be represented as a node sequence, starting from node 0, ending with node $2n+1$, and such that the pickup node of any handled request is located before the associated delivery node. A solution s is then the assignment of a route to each vehicle, such that all transportation request are handled exactly once. Given a route for the vehicle, the variables A_i, D_i and B_i can be computed in order to satisfy all time constraints.

To summarize, a solution must satisfy the following set of constraints:

1. the pickup and the delivery of a client i must be in the same route, the pickup node being visited before the delivery node,
2. at any time of the route, the number of persons in a vehicle k cannot not exceed its capacity Q_k,
3. for each node, the service time B_i fits within the time window on node i, i.e. $e_i \leq B_i \leq l_i$,
4. the riding time R_i of client i must not exceed a limit L,
5. the trip duration for a vehicle k must not exceed the maximum trip duration Max_{TD},
6. at most K vehicles are used.

Thus, the objective of the DARP is to find a feasible solution s of minimal total travel cost:

$$\min s = \sum_{k=1}^{K} \sum_{i=0}^{|n(k)|-1} c_{n_i(k) n_{i+1}(k)}$$

where $n(k)$ is the nodes sequence for vehicle k and $n_i(k)$ is the node at position i in $n(k)$.

As stressed by (Cordeau and Laporte, 2003), the quality of a solution is measured through the following three criteria:

- *Total Riding Time:* $TRT = \sum_{i=1}^{n} B_{i+n} - D_i$,
- *Total Waiting Time:* $TWT = \sum_{i=1}^{2n} W_i$,
- *Total Duration:* $TD = \sum_{k=1}^{K} A_{|n(k)|} - D_{n_0(k)}$.

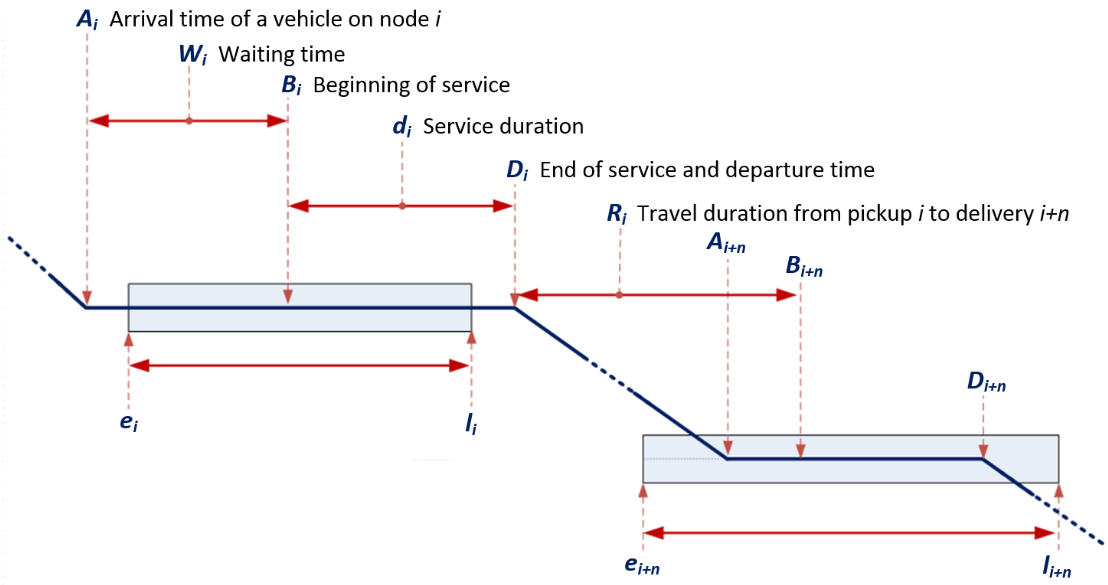

Figure 1: Variables description and notations

1.3 Setting privacy in DARP

Traffic congestion and rising oil prices raise the interest for alternative transportation modes in order to save travel costs and to reduce the travel time and the environmental concerns. Currently, most private vehicles are under-utilized and contain only one traveler during daily trips, leaving most of the seats available. The growing availability of geographical localization systems, mainly based on GPS information, allowes the creation of new kinds of personalized transportation services where private vehicle can be used as a private shared vehicle.

For instance, *Location-Based Services* (LBS) (Artigues et al., 2012; Quercia et al., 2010) and ridesharing services (Furuhata et al., 2013; Agatz et al., 2012, Bruck et al., 2017) are on a steep rise. Instead of being hired by a company, drivers in a ride-sharing system can be seen as private independent entities. These new types of transportation aim at bringing together travelers with similar travel path and time schedule. Yet, the users' privacy is an important issue in current ridesharing systems (Furuhata et al., 2013; Cottrill and Thakuriah, 2015). Since these services heavily depend on web applications, they can be targeted by malwares in order to access private data. Among them, the location data extracted through a privacy breach could allow to guess the user's home (Gambs et al., 2011) and its potential for burglaries and assaults. (Aïvodji et al., 2016) present an overview of main techniques for enforcing location privacy. We consider this problem too and we propose a DARP model in which an operation (pickup or delivery) is not represented by a single position – home, for instance - but by a set of potential nodes (for possible pickup and delivery nodes).

1.4 Contributions of the paper

The DARP with Private Vehicles and Alternative Nodes (DARP-PV-AN) is first presented. Then, an integer compact linear programming formulation is proposed and an Evolutionary Local Search (ELS) metaheuristic is developed using dynamic probabilities for neighborhood exploration, as in (Chassaing et al., 2016). The trip evaluation is based on the algorithm proposed by (Firat and Woeginger, 2011). It is extended in order to automatically select the best alternative node in case this trip is done by a private vehicle.

The remaining of the paper is organized as follows: Section 2 is dedicated to the definition of the DARP-PV-AN and its linear formulation. Section 3 presents the components in the ELS metaheuristic for computing solutions of good quality respecting privacy of users for the DARP-PV-AN. Section 4 reports the numerical results on modified instances from the literature, showing the interest of private vehicles. Concluding remarks are in Section 5.

2. DARP-PV-AN

2.1. Definition

In the DARP-PV-AN, in addition to the public fleet of vehicles, decentralized car-sharing can be organized by the use of a set of private vehicles that are located at a subset of nodes $R' \subseteq R$, each node representing a client that can use his/her car to reach his/her delivery destination possibly transporting other clients. The node-to-node ride sharing systems is an alternative to the public transportation system composed of a fleet of vehicles located at the depot node. In the DARP-PV-AN, these types of vehicles are used: (i) the public fleet located at a depot node and (ii) the private vehicles. Public vehicles provide a door-to-door system thanks to a centralized platform in charge of collecting and storing data and which is modeled by a DARP.

In case a client $i \in R'$ uses his/her private vehicle, the trip starts at i and ends at $i+n$ and its capacity is Q_r and a subset of nodes N^s where a pickup or delivery operation remains possible considering the client's habits and patterns of client i and motivated by is willing to make an acceptable detour.

The privacy for client i is ensured by a set of alternative pickup nodes $N^s_{i^+} \subset N^s$ in addition to its initial pickup node i^+ and by a set of alternative delivery nodes $N^s_{i^-} \subset N^s$ in addition to its initial delivery node i^-. This way, it is harder to guess the user's exact location. Moreover, it is also harder to anticipate where the client will be picked-up and dropped. If the client i is transported by a private vehicle, a node $\lambda^+ \in N^s_{i^+}$ and a node $\lambda^- \in N^s_{i^-}$ have to be selected for the solution, otherwise the initial pickup (i^+) and the initial delivery (i^-) nodes are used. Since two subset N^s_i and N^s_j can share some nodes, this model allows the possibility of so-called meeting nodes. Thus, two clients i and j can be handled at the same location. These meeting nodes will be automatically used if this improves the solution. DARP-PV-AN reduces to DARP when R' is empty and thus it is NP-hard.

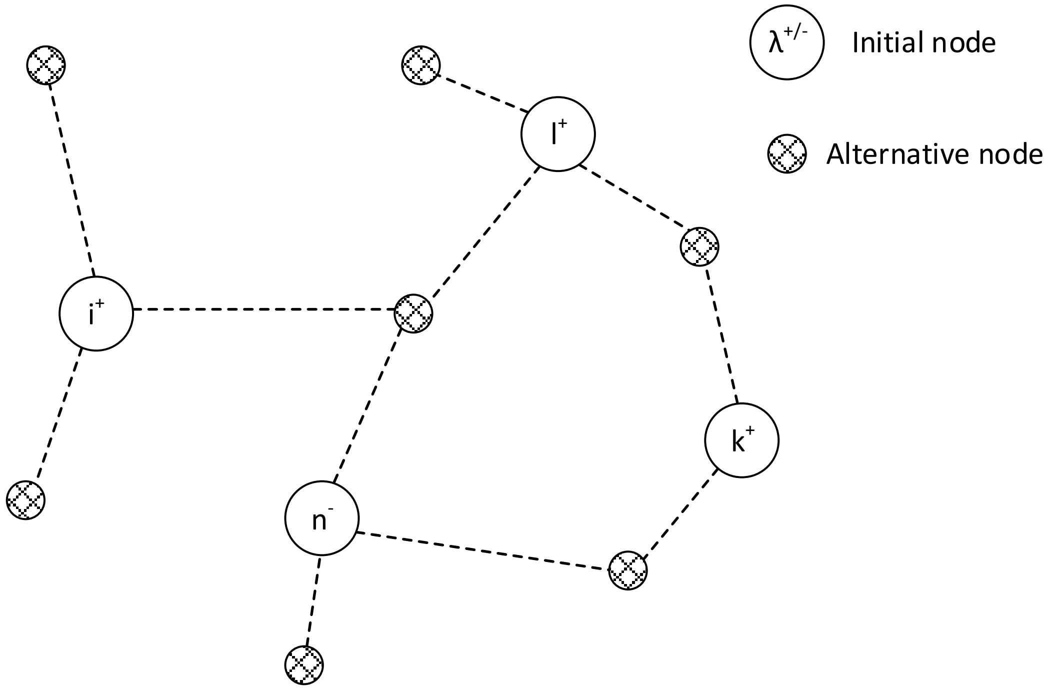

Figure 2: initial nodes with associated alternative nodes

Figure 2 illustrates the relation between initial nodes and alternative nodes. The connection is shown with a dashed line. Thus, for instance, nodes i^+, l^+ and n^- share a common alternative node and a private vehicle can pick clients i and l and drop client n at the same time.

2.2. Mixed Integer Linear Programming compact formulation

In this section, we propose a Mixed Integer Linear Programming (MILP) formulation for the DARP-PV-AN. Our MILP simultaneously combines for the first time the use of:

- both public and private vehicles,
- a set of alternative nodes available for each client when using a private vehicle,
- meeting nodes for clients.

2.2.1 Notation

N	set of initial nodes, $N = N^P \cup N^D \cup \{0\} \cup \{2n+1\}$,
N^P	set of initial pickup nodes, $N^P \subset N$,
N^D	set of initial delivery nodes, $N^D \subset N$,
N_i^S	set of alternative nodes linked to an initial node i, $i \in N$,
N_i^T	total set of nodes for i, containing initial node i and alternative nodes N_i^S, i.e. $N_i^T = N_i^S \cup \{i\}$, $i \in N$,
N^T	total set of all initial and alternative nodes, i.e. $N^T = (\bigcup_{i \in N} N_i^S) \cup N$,
K	number of public vehicles at the depot,
K'	number of private vehicles at the depot,
K^T	total number of vehicles, $K + K' = K^T$,
n	number of clients,
d_i	service duration at node i, $i \in N$,
$[e_i; l_i]$	time window at node i, $i \in N$,
q_i	client demand at node i, $i \in N$,
RT_i	client i maximum riding time, $i \in N$,
$c_{i,j}$	time between node i and node j, $i, j \in N$,
g_k	initial node position of the vehicle k, $k \in K^T$,
Q_k	maximum capacity of vehicle k, $k \in K^T$,
T	maximum trip length,
M	a large positive number.

2.2.2 Variables definition

The MILP uses 6 sets of continuous variables (A_i, B_i, D_i, A_{dpt}^k, D_{dpt}^k and v_i^k) and 2 sets of binary variables ($x_{i,j}^k$, $y_{l,m}$). The continuous variables are related to the arrival times of vehicles A_i and the departure time D_i, and both starting service time B_i and vehicle load v_i^k. Departure time from the depot and return time to the depot are also continuous. The decision variables focus on the arc selection for the trips.

A_i	arrival time of a vehicle at node i, $i \in N \backslash \{0, 2n+1\}$, $A_i \geq 0$,
B_i	beginning time of service at node i, $i \in N \backslash \{0, 2n+1\}$, $B_i \geq 0$,
D_i	departure time of a vehicle at node i, $i \in N \backslash \{0, 2n+1\}$, $D_i \geq 0$,

A_{dpt}^k arrival time of a vehicle k at depot, $k \in K^T$, $A_{dpt}^k \geq 0$

D_{dpt}^k departure time of a vehicle k from the depot, $k \in K^T$, $D_{dpt}^k \geq 0$

v_i^k vehicle k load when arriving at node i, $i \in N$, $k \in K^T$, $v_i^k \geq 0$,

$$x_{i,j}^k = \begin{cases} 1 \text{ if one arc is used between } N_i^T \text{ and } N_j^T \text{ by vehicle } k,\ i,j \in N,\ k \in K^T, \\ 0 \text{ otherwise,} \end{cases}$$

$$y_{l,m} = \begin{cases} 1 \text{ if arc } (l,m) \text{ is used, } l,m \in N^T, \\ 0 \text{ otherwise.} \end{cases}$$

2.2.3 Constraints and objective function

The set of constraints can be partitioned into several subsets: each subset of equations represents one type of constraint.

Time constraints: the first set of constraints ensures that arrival, beginning and departure variables satisfy the time windows of the corresponding node. Precedence constraints are also considered.

Constraints (1) ensure that the service time B_i starts in the time window $[e_i, l_i]$:

$$e_i \leq B_i \leq l_i, \qquad \forall i \in N^P \cup N^D. \tag{1}$$

Constraints (2) ensure that service time B_i starts after the arrival time A_i:

$$B_i \geq A_i, \qquad \forall i \in N^P \cup N^D. \tag{2}$$

Constraints (3) ensure vehicle k to leave the depot after its opening time e_0 and to come back before its closing time l_0. It also ensures that the departure from the depot is set before its arrival. Note that private vehicles start and end at the depot, but their travel cost to/from it is set to 0:

$$e_0 \leq D_{dpt}^k \leq A_{dpt}^k \leq l_0, \qquad \forall k \in K^T. \tag{3}$$

Constraints (4) ensure that the departure time D_i is set after the beginning of service B_i plus the duration of the service d_i:

$$D_i \geq B_i + d_i, \qquad \forall i \in N^P \cup N^D. \tag{4}$$

Constraints (5) ensure that the beginning of service B_i on a delivery cannot start before the departure time of its associated pickup D_{i-n} plus the time between them (for instance the shortest possible time between a pickup and its associated delivery). In the DARP-PV-AN, each initial pickup and delivery node has a set of associated alternative nodes: the shortest possible path is then the arc (l,m) of minimum length with $l \in N_{i-n}^T$ and $m \in N_i^T$:

$$B_i \geq D_{i-n} + \min c_{l,m}, \qquad \forall i \in N^D. \tag{5}$$

Constraints (6) deal with arrival time. The arrival time A_j is set after the departure time D_i plus the arc length $c_{l,m}$ if arc (l,m) is used, with $l \in N_i^T$ and $m \in N_j^T$. The big M technique is used here: if arc (l,m) is used, then $y_{l,m} = 1$ and $A_j = D_i + c_{l,m}$. Otherwise, A_j is not constrained by D_i

$$(D_i + c_{l,m}) + (y_{l,m} - 1) \times M \leq A_j \leq (D_i + c_{l,m}) - (y_{l,m} - 1) \times M, \qquad \forall i,j \in N^P \cup N^D. \tag{6}$$

Constraints (7) and (8) deal with arrival times at the depot and from the depot in the same way as constraints (6):

$$\left(D_i + c_{l,m}\right) + \left(y_{l,m} - 1\right) * M \leq A_{dpt}^k \leq \left(D_i + c_{l,m}\right) - \left(y_{l,m} - 1\right) * M, \quad (7)$$
$$\forall i \in N^P \cup N^D, \forall\, k \in K^T, \forall l \in N_i^T, \forall\, m \in N_0^T,$$

$$\left(D_{dpt}^k + c_{l,m}\right) + \left(y_{l,m} - 1\right) * M \leq A_i \leq \left(D_{dpt}^k + c_{l,m}\right) - \left(y_{l,m} - 1\right) * M, \quad (8)$$
$$\forall i \in N^P \cup N^D, \forall\, k \in K^T, \forall\, l \in N_0^T, \forall\, m \in N_i^T.$$

Constraints (9) ensure that the total trip duration is upper bounded by T:

$$A_{dpt}^k - D_{dpt}^k \leq T, \qquad \forall k \in K. \quad (9)$$

Constraints (10) limit the client's maximum riding time:

$$B_{i+n} - D_i \leq RT_i, \qquad \forall i \in N^P. \quad (10)$$

Load constraints: the second set of constraints ensures that the vehicle load does not exceed the capacity at any node of the trip.

Constraints (11) is the MTZ formulation where the vehicle load is updated at each node i. These constraints do not need to be set to equality because the vehicle load will be automatically set to meet the number of available places:

$$v_j^k \geq \left(v_i^k + q_i\right) + \left(x_{i,j}^k - 1\right) \times M, \qquad \forall i,j \in N, \forall k \in K^T. \quad (11)$$

Constraints (12) ensure that all vehicle are empty at the depot node:

$$v_0^k = 0, \qquad \forall k \in K^T. \quad (12)$$

Constraints (13) and (14) ensure that the capacity of a vehicle k is always positive and do not exceed its limit Q_k:

$$v_i^k \geq 0, \qquad \forall i \in N^P \cup N^D, \forall k \in K^T, \quad (13)$$

$$v_i^k \leq Q_k, \qquad \forall i \in N^P \cup N^D, \forall\, k \in K^T. \quad (14)$$

Flow balance constraints: The third set of constraints defines flow conservation for each vehicle.

Constraints (15) ensure that a private vehicle k cannot arrive at its starting position g_k from a node different from the depot, since artificial arc $(0, g_k)$ is used to start the private trip for k. Thus, vehicle k "virtually" starts from the depot with a 0-travel cost; all arcs between the depot and a node different from its starting position are forbidden. This is ensured by constraints (16). Both of them imply that a private vehicle k will use the arc $(0, g_k)$ of cost 0 between the depot and its starting position if k is used:

$$x_{i,g_k}^k = 0, \qquad \forall i \in N^P \cup N^D, \forall\, k \in K', \quad (15)$$

$$x_{0,i}^k = 0, \qquad \forall i \in (N^P \cup N^D)\backslash\{g_k\}, \forall k \in K'. \quad (16)$$

Constraints (17) and (18) ensure that a private vehicle ends at its delivery node in the same way as in constraints (15) and (16):

$$x_{g_k,i}^k = 0, \qquad \forall i \in N^P \cup N^D, \forall k \in K', \quad (17)$$

$$x_{i,0}^k = 0, \qquad \forall i \in (N^P \cup N^D)\backslash\{g_k\}, \forall k \in K'. \quad (18)$$

Constraints (19) ensure that a vehicle cannot start and end at the same position:

$$x_{i,i}^{k} = 0, \qquad \forall i \in N^P \cup N^D, \forall\, k \in K^T. \tag{19}$$

Constraints (20) ensure that an arc (i, j) is visited by at most one vehicle:

$$\sum_{k \in K^T} x_{i,j}^{k} \leq 1, \qquad \forall i, j \in N^P \cup N^D. \tag{20}$$

Constraints (21), (22) and (23) ensure that only one incoming and leaving arc must be used $\forall N_i^T$:

$$\sum_{i \in N} \sum_{k \in K^T} x_{i,j}^{k} \leq 1, \qquad \forall j \in N^P \cup N^D, \tag{21}$$

$$\sum_{j \in N} \sum_{k \in K^T} x_{i,j}^{k} \leq 1, \qquad \forall i \in N^P \cup N^D, \tag{22}$$

$$\sum_{j \in N} x_{i,j}^{k} = \sum_{j \in N} x_{j,i}^{k}, \qquad \forall i \in N, \forall k \in K^T. \tag{23}$$

Constraints (24) ensure that the pickup and the delivery of a client i is performed by the same vehicle:

$$\sum_{i \in N} x_{i,j}^{k} = \sum_{i \in N} x_{i,j+n}^{k}, \qquad \forall j \in N^P \cup N^D, \forall k \in K^T. \tag{24}$$

Constraints (25) ensure that every vehicle is used at most once:

$$\sum_{i \in N^P \cup N^D} x_{0,i}^{k} \leq 1, \qquad \forall k \in K^T. \tag{25}$$

Constraints (26) ensure that exactly one arc is used from N_i^T to N_j^T if they are connected:

$$\sum_{l \in N_i^T} \sum_{m \in N_j^T} y_{l,m} = \sum_{k \in K^T} x_{i,j}^{k}, \qquad \forall i, j \in N^P \cup N^D. \tag{26}$$

Constraints (27) and (28) ensure that the initial node for client i is used when i is handled by a public vehicle (27). On the other hand, if a private vehicle visits i, one of the alternative nodes in N_i^S is used (28). This enforces users' privacy:

$$y_{i,j} = \sum_{k \in K} x_{i,j}^{k}, \qquad \forall i, j \in N, \tag{27}$$

$$\sum_{l \in N_i^S} \sum_{m \in N_j^S} y_{l,m} = \sum_{k \in K'} x_{i,j}^{k}, \qquad \forall i, j \in N. \tag{28}$$

Constraints (29) ensure that an alternative node cannot be linked to another alternative node from the same client:

$$\sum_{l \in N_i^T} \sum_{m \in N_i^T} y_{l,m} = 0, \qquad \forall i \in N. \tag{29}$$

Graph reduction constraints: The last set of consists of cuts that reduce the solution space in order to reduce the computational time.

Constraints (30) and (31) forbid connections from a pickup node to a depot node and from a depot to a delivery node:

$$x_{i,0}^{k} = 0, \qquad \forall i \in N^P, \forall k \in K^T, \tag{30}$$

$$x_{0,i}^{k} = 0, \qquad \forall i \in N^{D}, \forall k \in K^{T}. \tag{31}$$

Constraints (32) prevent connection from a delivery node to a pickup of the same client:

$$x_{i+n,i}^{k} = 0, \qquad \forall i \in N^{P}, \forall k \in K^{T}. \tag{32}$$

For private trips, constraint (33) and (34) forbid the second node to be a delivery and the penultimate node to be a pickup (except the pickup and delivery nodes of the private vehicle itself):

$$x_{i,g_k+n}^{k} = 0, \qquad \forall k \in K', \forall i \in N^{P} \backslash \{g_k\}, \tag{33}$$

$$x_{g_k,i}^{k} = 0, \qquad \forall k \in K', \forall i \in N^{D} \backslash \{g_k\}. \tag{34}$$

Non-negative and integer constraints: All the variables are positives, and binary variables should be set to 0-1 values.

Minimization criteria: The objective function in the MILP minimizes the sum of distances:

.

$$\min \sum_{i \in N^T} \sum_{j \in N^T} y_{i,j} * c_{i,j} \tag{35}$$

Finally, the MILP consists in optimizing the objective function (35) subject to the constraints (1)-(34) and the variables definition. DARP-PV-AN is NP-hard as an extension of DARP.

3. Evolutionary Local Search metaheuristic

The Evolutionary Local Search (ELS) metaheuristic has been first proposed by (Wolf and Merz, 2007). It extends the Iterated Local Search (ILS) proposed by (Lourenço et al., 2003) and it has been then successfully applied to the VRP by (Prins, 2009). At each iteration of the ELS, several copies of the current solution are done. Each copy is modified by a mutation operator and then is improved by a local search. The best resulting solution over the improved copies is kept as new current solution for the next iteration. The rationale behind the algorithmic scheme of ELS is to deeply investigate the neighborhood of the current local optimum before leaving it. It is often combined with a wider exploration mechanism, like in GRASP for instance. It can be even embedded into a simpler multi-start, to manage the diversity during the global solution space exploration. The framework we propose is an extension of the approach in (Chassaing et al., 2016) that has been enriched and fully tuned to solve the DARP-PV-AN. We take into account the use of private vehicles while respecting privacy for users. This extension is integrated by modifying the evaluation and the mutation function. In addition, two new operators dedicated to private trips are developed.

The ELS scheme is described in Algorithm 1. The algorithm starts by creating an initial solution with a randomized heuristic (line 10). This initial solution is improved by a local search (line 10). Then, each of the ne main ELS iterations consists in:

- creating a neighborhood set of the current solution by copies and mutation (line 16),
- improving neighbors using a local search (line 17),
- keeping the best resulting solution for next iteration (line 19),
- updating the neighborhood activation probabilities used in the local search (line 21).

The local search procedure is called at lines 11 and 17 in order to improve the current solution (s and s'' respectively). It uses several VRP operators described in (Lin and Kernighan, 1973), (Potvin and Rousseau, 1995), (Braekers et al., 2014) and (Masson et al., 2014). These operators rely on basic moves and correspond to different ways to explore locally the solution space. An activation probability is associated to each operator and is updated every nr iterations of the ELS. The key features originally introduced in (Chassaing et al., 2016) are:

- an indirect representation for the solutions as a sequence of the requests and a decoding function allowing the creation of a feasible solution from this sequence ;

- a new randomized constructive heuristic to generate good and valid initial solutions, based of graph partition;
- an adaptive local search relying on dynamic probabilities. A probability is associated to each neighborhood structure. Each iteration, a neighborhood is selected according to these probabilities. The probabilities are updated with respect to the capability to improve the current solution (or not);
- a strongly efficient procedure to evaluate trip cost considering the shortest path between node and an iterative minimization of three criteria (the total duration, the total riding time and the total waiting time).

Algorithm 1: ELS()

```
Input
   nd: nb neighbors
   ne: nb ELS iterations
   nr: nb iterations with same probabilities P
Output
   S: solution
Begin
   initialization of P: P[i] = 0.25, ∀i = 1..6
   // step 1. Creation of the initial solution
   s := randomized_constructive_heuristic()
   s := local_search(s,P)
   // step 2.Improvement by ELS
   for i := 1 to ne do
      s' := s
      for k := 1 to nd do
         s" := mutation(s')
         s" := local_search(s",P)
      end for
      s := best of all s"
      if (i mod nr)=0 then
         update(P);
      end if
   end for
   return s
End
```

3.1 Trip evaluation

The solution evaluation method consists in evaluating independently each trip. It takes into account the kind of vehicle (public or private) and integrates the privacy of users. The basic trip evaluation technique has been introduced by (Firat and Woeginger, 2011) providing a linear time trip evaluation that improves the quadratic time method proposed in (Cordeau and Laporte, 2003). Our proposition mainly consists of two steps:

- step 1: allow the use of private vehicle by adding zero cost arcs from the depot to the pickup node corresponding to the private vehicle, and from its delivery node to the depot.
- step 2: when a vehicle visits a client node, automatically choose the node (initial or alternative) where the client will be picked-up or dropped-off.

The modifications achieved in step 1 allow a common evaluation function for both public vehicles and private vehicles. This functionality is provided by the insertion of temporary arcs into the evaluated trip. As stressed in Figure 3, temporary zero cost arcs $c_{1^+,DPT}$ and $c_{DPT,1^-}$ are added (1) from the depot to the initial (1^+) and last (1^-) position of the private vehicle before a private trip is evaluated. Then, the trip is evaluated (2) as for a public trip. Once the evaluation has been done, these additional arcs are removed (3). This modification allows the algorithm to evaluate public and private trips in linear time.

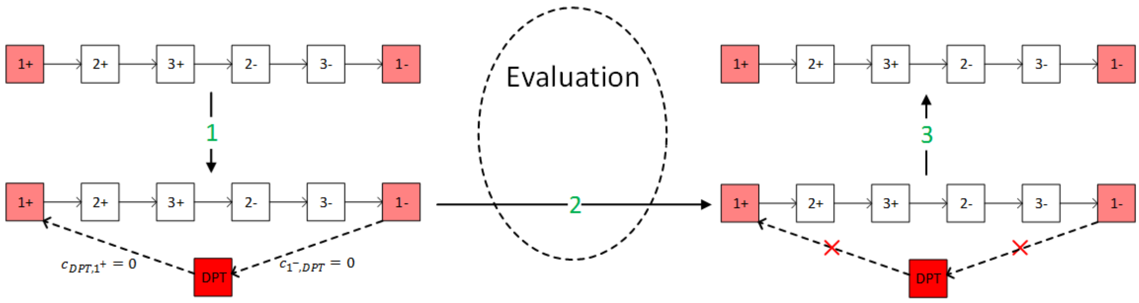

Figure 3: private evaluation trip using temporary arcs

In step 2, we take into account the privacy for the user. Each client i is associated with an initial pickup/delivery node i^+/i^- and sets of alternative nodes $N_{i^+}^S/N_{i^-}^S$. Thus, one pickup node and one delivery node must be selected depending on the type of vehicle used to transport him (public or private). For a public vehicle, the initial pickup and delivery nodes i^+/i^- are used. Otherwise, a pickup node $\alpha^+ \in N_{i^+}^S$ and a delivery node $\alpha^- \in N_{i^-}^S$ must be chosen. This selection is automatically and optimally done by first building a layered graph in which each layer corresponds to a client node. For each layer, the set of nodes the vehicle is allowed to visit is set. Then, the arcs between each successive layer are created. Once the layered graph has been created, a shortest path can be computed, for instance using Dijkstra algorithm (Dijkstra, 1959) in order to minimize the distance for the private vehicle.

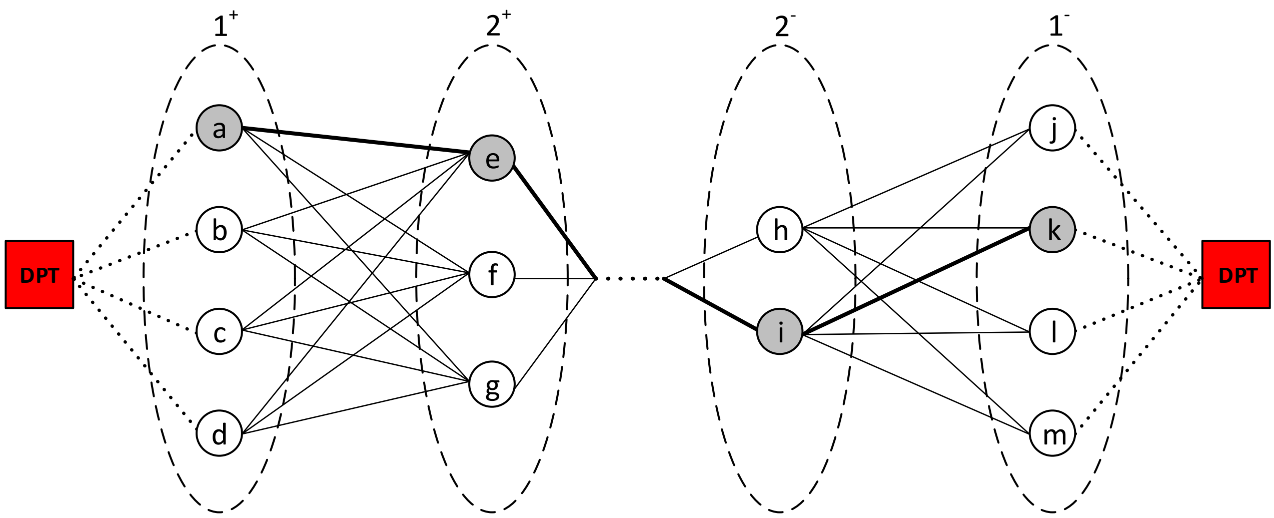

Figure 4: alternative nodes selection for a private trip

The example in Figure 4 illustrates the way alternative nodes are optimally selected in case a private vehicle is used (the private vehicle of client 1 is selected). It starts its trip at node 1^+ and end at node 1^-. Temporary arcs of zero cost are added between depot and $1^+/1^-$. For this example, the vehicle also performs the request for the client 2 in the trip. Because a private vehicle performs the trip, initial nodes cannot be used in order to preserve client's privacy. Thus, an alternative node must be selected for each pickup and delivery position: in our case, alternative nodes $\{a, e, .., i, k\}$ are visited in the shortest path.

It is worth noting the alternative nodes selection is not done explicitly in ELS. It is performed automatically through the shortest path computation in the private trip evaluation. Here, we are assuming the alternative nodes are not set too far from the initial node, in such a way it is highly unlikely that a private trip where alternative nodes are not selected by shortest path algorithm would be valid, while the private trip with the shortest path used would not. Thus, we can evaluate private trips after choosing alternative nodes with a shortest path algorithm. However, if the problem contains alternative nodes at a significant distance from the initial node, this would break our assumption and both the shortest path and the evaluation should be performed at the same time and not sequentially. In such a case, evaluating the shortest path with time windows is NP-Hard (Desrochers et al., 1988). This general case will be considered in a future work.

3.2 Mutation

The mutation is a function initially introduced in the context of genetic algorithms. It is used in ELS and consists in randomly performing a small modification on the current solution s in order to get a new solution s' sharing similarities with s. Two operations have been investigated:

operation 1 : a transportation request is removed from its trip and assigned to a new one,

operation 2 : a crossover between two trips is achieved.

The second operation is an adaptation of the crossover operator from genetic algorithms. We define two trips λ_1 and λ_2 of respective length n_{λ_1} and n_{λ_2}. Two positions p_1 and p_2 are randomly chosen in λ_1 and λ_2, such that $1 \leq p_k \leq n_{\lambda_k}$, $k = 1,2$. All transportation request for which position p_i is between its pickup and its delivery are tagged true in each trip. If a private vehicle is used, the vehicle's client is set aside. All nodes of λ_1 before p_1, associated to untagged transportation requests, are iteratively inserted into the new trip t_1. Then, all nodes of λ_2 after p_2, associated to untagged transportation requests, are iteratively inserted into the new trip t_1. The roles of λ_1 and λ_2 are swapped to build the other new trip t_2. Tagged transportation requests are inserted one by one in one trip at the best position considering the trip cost. If a private vehicle is used, its owner is inserted such as its vehicle is used in the trip: t_1 and t_2 are tested and the best one is kept.

For instance, consider two trips λ_1 and λ_2 in Figure 5 with respective cut position $p_1 = 3$ and $p_2 = 3$. Nodes $5^+, 3^+, 5^-, 3^-$ are tagged because their pickup and their delivery are separated by the cut.

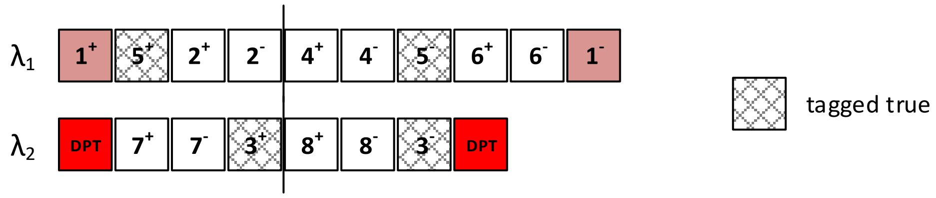

Figure 5: initial trips λ_1 and λ_2

Nodes 1^+ and 1^- are tagged because they correspond to a private vehicle starting and ending node. In Figure 6, nodes $2^+, 2^-, 8^+, 8^-$ are inserted in t_1 and nodes $7^+, 7^-, 4^+, 4^-, 6^+, 6^-$ in t_2. Then nodes $5^+, 5^-, 3^+, 3^-$ are inserted into their best positions in Figure 7.

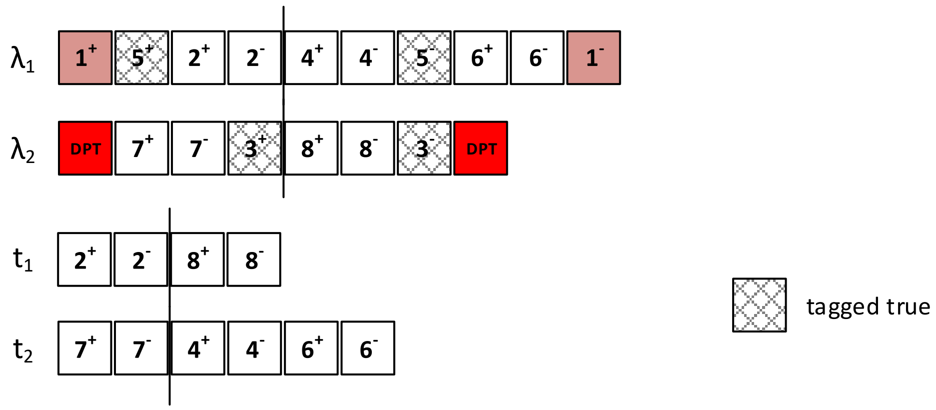

Figure 6: untagged nodes insertion

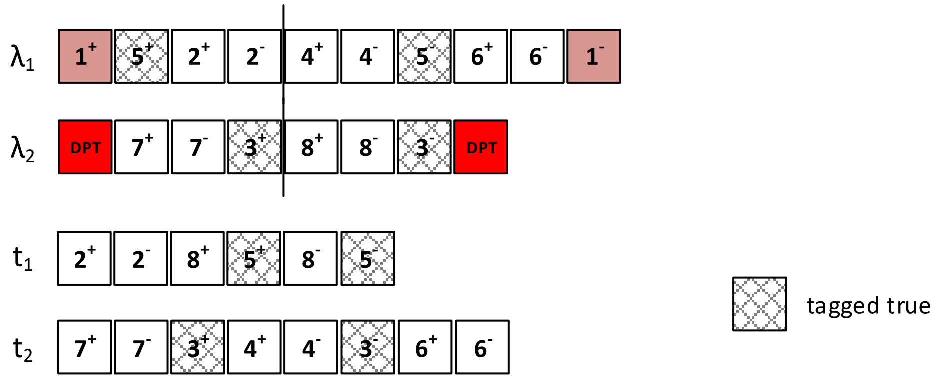

Figure 7: tagged nodes insertion

The last step is illustrated in Figure 8: nodes 1^+ and 1^- that lead to a private vehicle starting and ending position need to be inserted. Both trips are tested and the best one is kept.

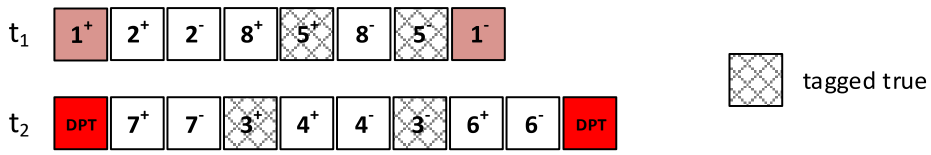

Figure 8: private trip insertion

3.3 Local search

The local search is an iterative heuristic method that tries to reduce the solution cost by exploring several neighborhoods of the current solution. A set of six basic moves, with a probability for each to be called at each iteration, operates on the DARP layer at the core of the problem:

- A 4-Opt move as proposed by (Lin and Kernighan, 1973), where 4 arcs are removed from a trip and all possible ways to reconnect the remaining segments are investigated, keeping the best solution;
- A Cut and Paste that removes a part of a trip to insert it somewhere else, as in a two points crossover;
- A Removal of the worst client from a trip that removes the request responsible for the largest detour and then inserts it at its best position in a random trip;
- A Relocation implementing the method proposed by (Braekers et al., 2014), where a random request is removed from a trip and inserted into another one;
- A 2-Opt* move as proposed by (Potvin and Rousseau, 1995), where an arc is removed from two different trips before recombining the resulting parts;
- A Requests Exchange as proposed by (Braekers et al., 2014), swapping two transportation requests from different trips.

These DARP moves are completed by two new moves addressing the nature of the trip, either public or private:

- `removePrivateTrip()`: remove a trip performed by a private vehicle and insert all its transportation requests into the other trips.
- `createPrivateTrip()`: create a new private trip from a request handled by a public trip.

In `createPrivateTrip()`, we first find a public trip (λ) and the pickup (p_i^+) and the delivery (p_i^-) positions of a client i that can perform a private trip. Then, i is removed from λ and a new private trip λ' made of only client i is created. All the transportation requests in λ whose pickup and delivery are located between p_i^+ and p_i^- are tagged true. Next, all these transportation requests are tentatively and sequentially inserted into λ'.

For instance, consider the public trip λ_5 in Figure 9 with client 5 in position $p_5^+ = 2$ and $p_5^- = 7$ owns a private vehicle (unused). Nodes $1^+, 1^-$ are tagged true since they correspond to a transportation request between p_5^+ and p_5^-.

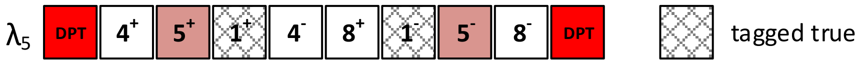

Figure 9: public trip λ_5, at the beginning of `createPrivateTrip()`

Then, in Figure 10, client 5 is extracted from λ_5 and a new private trip α_5 is created with 5. Then the clients tagged true are tentatively inserted in α_5 in Figure 11. As the insertion is possible, the final trips are shown in Figure 12.

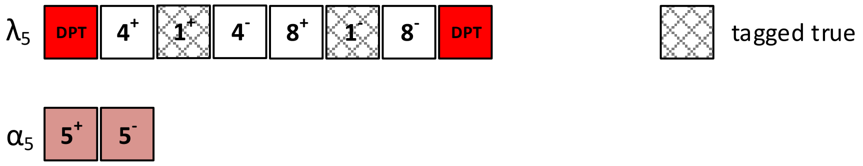

Figure 10: private trip α_5 creation

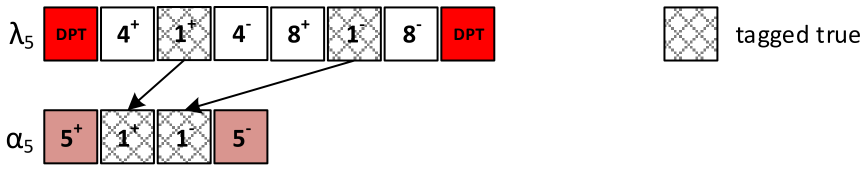

Figure 11: client 1 insertion in private trip α_5

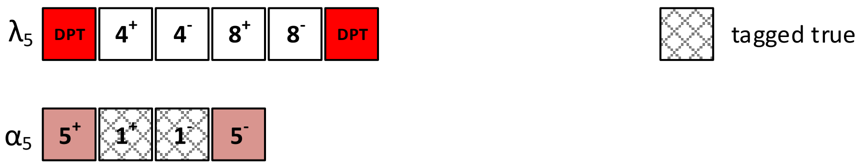

Figure 12: public and private trips at the end of `createPrivateTrip()`

4. Numerical experiments

Both the MILP model and the ELS are tested on three sets of instances:

1. The first set contains 20 instances modified from the instances in (Ropke et al., 2007). They are designed to evaluate the MILP model. The modification from the original instances consists in adding a set of private vehicles, more precisely the clients able to use their vehicle. In addition, alternative nodes linked to initial nodes have been created. Some clients have been removed too in order to keep the instances small enough for the MILP resolution. Preliminary tests showed that 8 requests is a practical limit in our case. Including alternative nodes, this contains up to 47 nodes.
2. The second set is composed of 20 instances extending the set proposed by (Cordeau and Laporte, 2003). These instances are dedicated to evaluate ELS. As for the first set, these instances have been modified to add private vehicles and positions of alternative nodes linked to initial nodes. Clients have not been removed, as ELS is able to handle instances with up to 120 transportations requests.
3. The last set is made of 35 realistic instances we created based on asymmetric routing graphs generated by (Duhamel et al., 2009). These instances contain up to 60 requests. This set is also dedicated to the ELS performance evaluation.

The experiments have been done on an Intel core i7-4790 @ 3.60GHz with a single thread activated. CPLEX solver 12.6 is used to compute optimal solutions for our MILP model and a 1h time limit is set. Considering

(Dongarra, 2014) this computer can be stated about 4130 MFlops. The best-found solution of the ELS are reported. All the methods have been coded in C++.

4.1 MILP evaluation

The MILP is solved with CPLEX. The results are gathered in Table 1: column $N-r$ refers to the number of requests, N^T is the total number of nodes (initial and alternative), K is the number of public vehicles, K' the number of private vehicles and Obj the value of objective function, *i.e.* the sum of travelled distances as defined in (35). LB is the best lower bound found by CPLEX in the time limit. T is the time needed to find the best solution (within the limit of 3600 s) and TT the total time. $\#K$ and $\#K'$ are respectively the number of public vehicles and the number of private vehicles used in the solution. NFS indicates no solution has been found in the time limit, while a result with $\leq$ corresponds to the value of the best feasible solution found in the time limit. It is thus an upper bound.

A simple observation relates the number of private and public vehicles used according to the number of private vehicles available. As can be seen from instance PCD_2v_12n_0p to instance PCD_5v_14n_7p (second block of instances), the more private vehicles are available, the less public vehicles are used. This leads to a solution from 32.36 with two public vehicles to 19.20 with five private vehicles. We also observe the difficulty of our model to solve instances with more than 7 requests: the time limit of 3600 seconds is reached and the optimal solutions are not found for five instance. For three out of these five instances (PCD_0v_14n_2p, PCD_1v_16n_1p and PCD_0v_16n_2p) no feasible solution has even been found. This is the reason we proposed a metaheuristic to handle instances with realistic size in the next section.

One can also note, adding a single potential private vehicle, that is going from $K'=0$ to $K'=1$, can improve substantially the value of the solution. Yet the required computing time increases sharply as well. Moreover, the configuration with 0 private nodes corresponds to classic DARP, while the configuration with 0 public node corresponds to car sharing. Thus, our model can handle a wide range of combinations between these two transportation modes. The last instance in each block corresponds to the unrestricted context with respect to the requests, for which all clients can use their own vehicle.

Table 1: Results for the MILP on the modified instances from (Ropke et al., 2007)

Name	N-r	N^T	K	K'	Obj	LB	T*(s)	TT (s)	#K	#K'
PCD_2v_10n_0p	5	31	2	0	32.90	32.90	0.84	0.95	2	0
PCD_1v_10n_1p	5	31	1	1	28.92	28.92	1.67	1.89	1	1
PCD_2v_10n_1p	5	31	2	1	28.92	28.92	2.34	2.51	1	1
PCD_0v_10n_2p	5	31	0	2	29.62	29.62	14.45	14.91	0	2
PCD_5v_10n_5p	5	31	5	5	20.10	20.10	5.04	7.11	0	3
PCD_2v_12n_0p	6	37	2	0	32.36	32.36	16.65	20.94	2	0
PCD_1v_12n_1p	6	37	1	1	27.64	27.64	110.25	114.89	1	1
PCD_2v_12n_1p	6	37	2	1	27.64	27.64	273.78	367.58	1	1
PCD_0v_12n_2p	6	37	0	2	25.05	25.05	21.98	38.58	0	2
PCD_5v_12n_6p	6	37	5	6	19.20	19.20	288.34	316.64	0	5
PCD_2v_14n_0p	7	42	2	0	35.98	35.98	195.06	203.47	2	0
PCD_1v_14n_1p	7	42	1	1	32.10	32.10	1682.64	2438.23	1	1
PCD_2v_14n_1p	7	42	2	1	32.10	32.10	984.80	1975.21	1	1
PCD_0v_14n_2p	7	42	0	2	/	26.50	/	3600.00	/	/
PCD_5v_14n_7p	7	42	5	7	24.37	21.90	3521.21	3600.00	0	6
PCD_2v_16n_0p	8	47	2	0	48.19	48.19	380.57	1485.53	2	0
PCD_1v_16n_1p	8	47	1	1	/	34.46	/	3600.00	/	/
PCD_2v_16n_1p	8	47	2	1	48.78	34.58	3320.82	3600.00	2	1
PCD_0v_16n_2p	8	47	0	2	/	33.50	/	3600.00	/	/
PCD_5v_16n_8p	8	47	5	8	31.60	27.65	3316.85	3600.00	0	6

4.2 ELS evaluation on (Cordeau and Laporte, 2003) instances

The first evaluation of our ELS metaheuristic is performed on 5 instances proposed by (Cordeau and Laporte, 2003) and results are shown in Table 2. These instances address the classical DARP problem: no private vehicle are available, thus alternative nodes will not be used. In this table, BKS refers to the best-published results of (Chassaing et al., 2016) and the column Obj refers to the results provided by our ELS. Optimal solutions were found on instances R1a and R2a, otherwise solutions with a gap less than 0.5% are given for instances R3a and R4a. These results illustrate ELS also provides good results on the classical DARP, which is a specific version of DARP-PV-AN.

Table 2: Results for ELS on the (Cordeau and Laporte, 2003)'s instances

Type	Name	N-r	N^T	K	BKS	Obj	GAP%	T*(s)	TT(s)	#K
DARP	R1a	24	48	3	190.02	190.02	0.00	0.00	15.00	3
	R2a	48	96	5	301.34	301.34	0.00	24.00	60.00	5
	R3a	72	144	7	532.00	534.28	0.43	53.80	150.00	7
	R4a	96	192	9	570.25	572.95	0.47	324.00	456.00	8
	R5a	120	240	11	626.93	640.27	2.13	360.00	498.00	11

In this table, T^* is the time to reach the best solution and T the total time of the metaheuristic. The gap has been computed with the following formula:

$$GAP = \frac{Obj * 100}{BKS} - 100$$

,where Obj is the value of the objective function found by our algorithm and BKS the best know solution of the original instance. The BKS column reports values obtained in (Chassaing et al., 2016).

In Table 3, we use instances from Table 2, modified to include private vehicles and alternative nodes. For each original DARP instances, the number of allowed private vehicles goes from 1 to 3. Column UB refers to the best-found solution of the corresponding DARP instances, which is, obviously, an upper bound for the DARP-PV-AN instances. The objective function should decrease with respect to the number of available private vehicles. This can be observed for instance R1a and, to a lesser extent, for instances R2a and R3a. However, for R4a and R5a, the size of the instances (650 and 819 nodes) prevents the method from reaching near-optimal solutions.

Table 3: Results for ELS on the modified instances from (Cordeau and Laporte, 2003)

Type	Name	N-r	N^T	K	K'	UB	Obj	GAP%	T*(s)	TT(s)	#K	#K'
DARP-PV-AN	R1a-1p-pn	24	168	3	1	190.02	190.02	0.00	0.00	15.00	3	0
	R1a-2p-pn	24	168	3	2	190.02	190.02	0.00	0.00	15.00	3	0
	R1a-3p-pn	24	168	3	3	190.02	186.90	-1.64	504.00	15.00	2	1
	R2a-1p-pn	48	318	5	1	301.34	299.84	-0.50	38.40	60.00	5	1
	R2a-2p-pn	48	318	5	2	301.34	294.39	-2.31	44.40	60.00	5	2
	R2a-3p-pn	48	318	5	3	301.34	294.92	-2.13	32.40	60.00	5	2
	R3a-1p-pn	72	479	7	1	532.00	529.90	-0.39	85.20	150.00	7	1
	R3a-2p-pn	72	479	7	2	532.00	530.39	-0.30	72.00	150.00	7	2
	R3a-3p-pn	72	479	7	3	532.00	522.57	-1.77	76.20	150.00	7	3
	R4a-1p-pn	96	650	9	1	570.25	574.94	0.82	298.80	438.00	8	0
	R4a-2p-pn	96	650	9	2	570.25	572.99	0.48	283.80	438.00	8	0
	R4a-3p-pn	96	650	9	3	570.25	573.10	0.50	256.80	438.00	8	1
	R5a-1p-pn	120	819	11	1	626.93	629.18	0.36	367.20	498.00	11	1
	R5a-2p-pn	120	819	11	2	626.93	633.76	1.09	231.00	498.00	11	2
	R5a-3p-pn	120	819	11	3	626.93	638.50	1.85	190.20	498.00	11	3

4.3 ELS evaluation on new instances

Previous tests on our ELS metaheuristic were done on classical instances tuned for the DARP-PV-AN. The algorithm has also been benchmarked on a new set of instances dedicated to the DARP-PV-AN, with medium time window on each node and common nodes between requests for carpooling (meeting nodes). To our knowledge, such test instances are not available in the literature for the version of the DARP studied in this paper.

The randomly generated instances use asymmetric routing graphs generated by (Duhamel et al., 2009) that give a set of node with their positions and a set of oriented arcs. Based on these routing schemes, we created 35 new instances that contain from 10 to 60 requests. Each request is composed of 1 initial pickup and 1 initial delivery node, and, for each of them, from 1 to 4 associated alternative nodes. These alternative nodes are adjacent to their initial one in the graph. For each instance, pickup and delivery locations are generated using a procedure that creates clusters of vertices. This positioning has been set to model daily commuting: pickup nodes (resp. delivery nodes) are chosen in the same area. For each node, the service time d_i is equal to 2 and the load is equal to either 1 or -1 depending on whether the node is a pickup or a delivery. The depot is located at the middle of the graph. For each arc $(i, j) \in A$, its routing cost $c_{i,j}$ is equal to the shortest path from i to j.

A time window $[e_i; l_i]$ is associated to each node i with $0 \leq e_i \leq l_i \leq 1440$. In order to simulate realistic requests, we assume that each delivery could represent one of the three following possible situations: (i) the beginning of a working day, (ii) the return to home or (ii) an arbitrary move. Then, for each delivery, l_{i^-} has a 0.4 probability to be in $[360; 600]$, a 0.4 probability in $[960; 1200]$ and a 0.2 probability in $[0; 1440]$. The shortest path SP_i is computed in the graph between the pickup and the delivery of the request i. Finally, pickup time windows $[e_{i^+}; l_{i^+}]$ and delivery time window $[e_{i^-}; l_{i^-}]$ of a request i are computed as follow:

- $e_{i^+} = l_{i^-} - \alpha * SP_i$,
- $l_{i^+} = l_{i^-} - SP_i$,
- $e_{i^-} = e_{i^+} + SP_i$,

, with $\alpha \in \mathbb{N}$ representing the additional time tolerance compared to the shortest path.

In all instances, the maximum route duration is set to 480, maximum capacity of a vehicle is equal to 10 and maximum riding time is equal to 240. For these instances with medium time windows on each node, a large number of public vehicle is provided in order to help the algorithm at finding initial solutions. As we can see in results Table 3, few of them are really used in the solutions. The number of private vehicles ranges from 0% of clients (classical DARP problem) to 100% of clients and their positions are randomly selected.

Table 4: Results for ELS on the new set of instances

Name	N-r	N^T	K	K'	Obj	Gap%	T*(s)	TT(s)	#K	#K'
PCD_20_0VP	10	61	4	0	811.35	/	0.00	60.00	3	0
PCD_20_2VP	10	61	4	2	730.65	-9.94	30.00	60.00	3	1
PCD_20_4VP	10	61	4	4	730.65	-9.94	30.00	60.00	3	1
PCD_20_6VP	10	61	4	6	658.24	-18.87	30.00	60.00	2	4
PCD_20_10VP	10	61	4	10	435.52	-46.32	34.80	60.00	0	7
PCD_40_0VP	20	125	8	0	1135.31	/	6.60	120.00	4	0
PCD_40_2VP	20	125	8	2	1028.41	-9.42	60.60	120.00	4	2
PCD_40_6VP	20	125	8	6	747.73	-34.14	118.80	120.00	1	6
PCD_40_8VP	20	125	8	8	667.43	-41.21	78.00	120.00	1	6
PCD_40_10VP	20	125	8	10	647.31	-42.98	60.60	120.00	1	8
PCD_40_20VP	20	125	8	20	565.69	-50.17	88.80	120.00	0	12
PCD_60_0VP	30	191	20	0	1530.00	/	46.20	180.00	6	0
PCD_60_2VP	30	191	20	2	1493.06	-2.41	115.80	180.00	4	2
PCD_60_8VP	30	191	20	8	1240.68	-18.91	165.00	180.00	4	5
PCD_60_10VP	30	191	20	10	1044.67	-31.72	178.80	180.00	2	8

PCD_60_16VP	30	191	20	16	916.88	-40.07	146.40	180.00	1	9
PCD_60_30VP	30	191	20	30	961.22	-37.18	97.20	180.00	0	17
PCD_80_0VP	40	243	20	0	2060.88	/	2.40	240.00	7	0
PCD_80_2VP	40	243	20	2	2060.88	0.00	2.40	240.00	7	0
PCD_80_10VP	40	243	20	10	1991.44	-3.37	129.60	240.00	5	2
PCD_80_14VP	40	243	20	14	1840.34	-10.70	189.00	240.00	4	5
PCD_80_20VP	40	243	20	20	1509.49	-26.76	240.00	240.00	2	11
PCD_80_40VP	40	243	20	40	1447.02	-29.79	162.00	240.00	1	18
PCD_100_0VP	50	307	50	0	2415.43	/	81.60	300.00	7	0
PCD_100_4VP	50	307	50	4	2386.15	-1.21	163.20	300.00	7	3
PCD_100_14VP	50	307	50	14	2029.22	-15.99	268.80	300.00	4	8
PCD_100_18VP	50	307	50	18	2078.92	-13.93	294.60	300.00	4	9
PCD_100_26VP	50	307	50	26	2079.45	-13.91	292.20	300.00	4	10
PCD_100_50VP	50	307	50	50	1888.48	-21.82	242.40	300.00	1	21
PCD_120_0VP	60	371	50	0	2928.59	/	257.40	360.00	10	0
PCD_120_4VP	60	371	50	4	2804.60	-4.23	268.20	360.00	8	1
PCD_120_16VP	60	371	50	16	2465.74	-15.80	335.40	360.00	5	5
PCD_120_20VP	60	371	50	20	2407.83	-17.78	320.40	360.00	6	4
PCD_120_30VP	60	371	50	30	2581.43	-11.85	269.40	360.00	7	3
PCD_120_60VP	60	371	50	60	2106.29	-28.08	360.00	360.00	2	20

The use of private vehicles while respecting privacy of users significantly improves solutions of a classical DARP, as stressed on the Gap column in Table 4. For instance PCD_40, the saving goes up to 50.17% when all clients can use their vehicle (even if only 12 are really used). This behavior holds for all instances. Note that the saving also depends on private vehicle capacity, which has been set to 10 here. For a more restricted capacity, the improvement would have been more limited. For all instances, ELS uses all the allowed CPU time. Yet, the time to find the best solution increases when private vehicles can be used, compared to the configuration with no private vehicle (*i.e.* $K' = 0$).

We also evaluated the quality of the solution with the criteria defined by (Cordeau and Laporte, 2003): TRT, TWT, and TD. The figures 13, 14 and 15 reports average values from solutions in Table 4. As can be seen on Figure 13, all of these criteria are improved, and TWT is close to 0% when all the clients can use their private vehicle (which does not mean all of them are used).

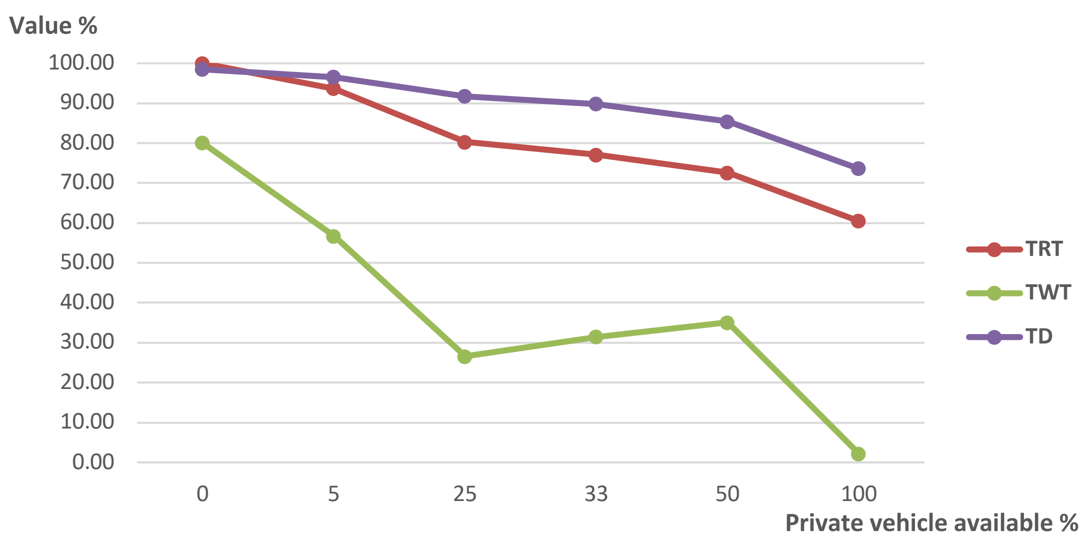

Figure 13: Quality of service

We can also evaluate the impact of private vehicles on the global traffic by measuring the number of arcs used according to the number of private vehicles. As shown in Figure 14, the number of arcs used goes down from 324 in average with 0% of private vehicle to 219 with 100% of private vehicles available. This reduces the network utilization and thus, improves traffic flow.

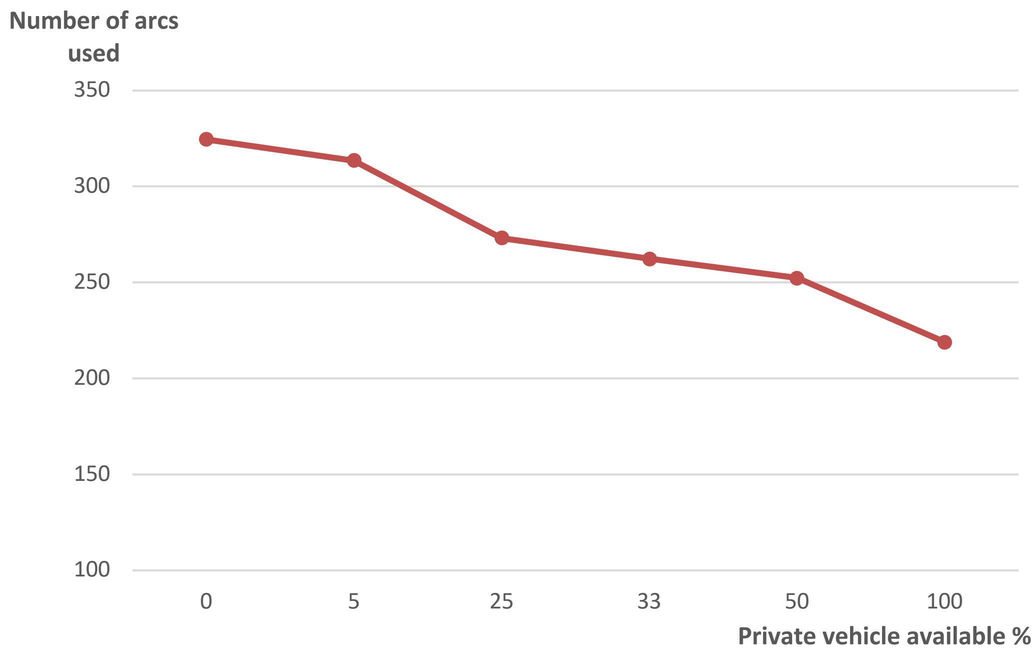

Figure 14: Number of arcs used

Our next test evaluates the average deviation from the shortest path for each request. It is computed as the ratio between the current trip distance and the distance of the shortest trip from his/her pickup to his/her delivery. Note that some other requests could be handled in the same time by the same vehicle, which would further increases the deviation. Thus, the path from its pickup node to its delivery node may not be the shortest one. We studied this average deviation according to the number of private vehicle used. As shown in Figure 15, in which 100% means the shortest possible path, the average length of a transport request corresponds to 161.44% of the shortest path when no private vehicle are used. With private vehicles, this deviation reduces progressively to 129.29%. This also means an average 18.75% time saving for a request.

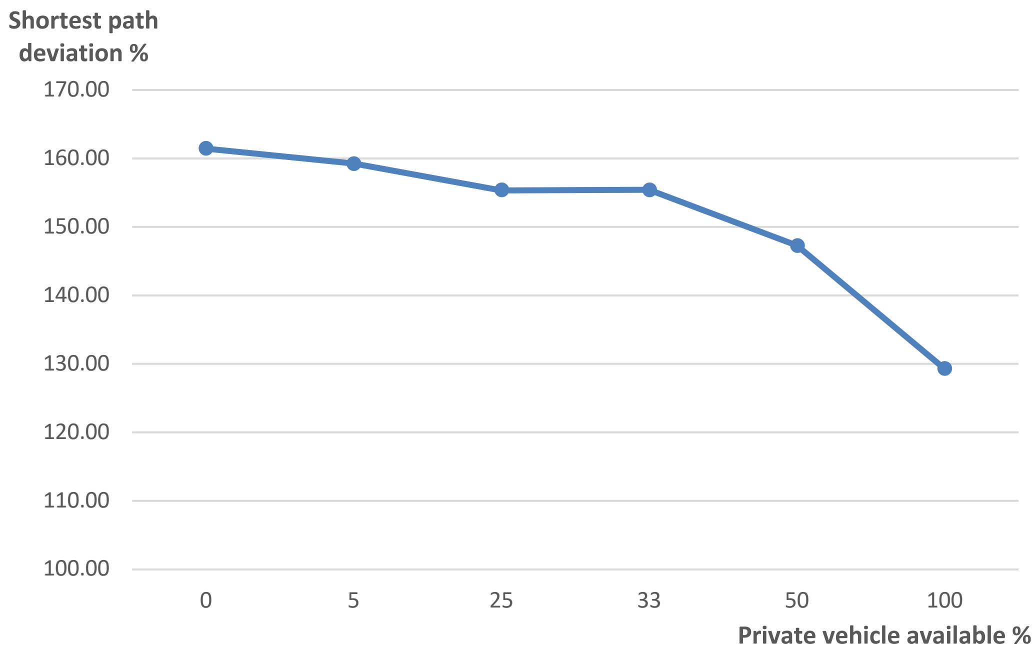

Figure 15: Average shortest path deviation

Our last observation focuses on the use of meeting nodes by private vehicles. In Figure 16, some clients share alternative nodes. Thus, these alternative nodes can be used as meeting node to handle several requests at the same time in order to optimize the objective function. In this example, a part of a private trip is presented; the private vehicle trip is shown in bold lines, the initial nodes handled in the trip are in grey and the alternative nodes visited are dashed. As can be seen, the alternative node 205 is shared by the requests whose initial nodes are 191, 204 and 220. Since the alternative node 205 is visited, the associated three requests can be performed at the same time. In this case, this is a triple drop-off, but it could correspond to any combination of pickup and delivery

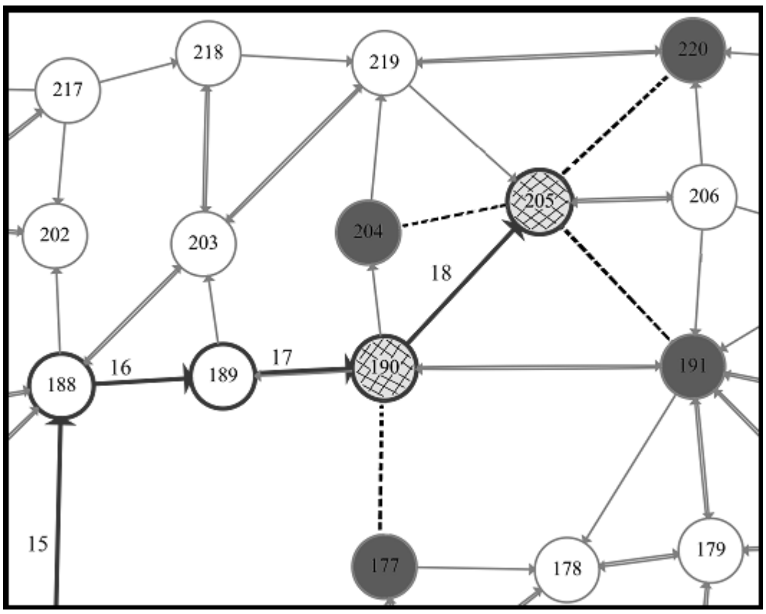

Figure 16: Node 205 as an alternative node in turn 7 of instance PCD_100_14VP

5. Concluding remarks

In this paper, we have introduced the new Dial-A-Ride Problem with Private Vehicles and Alternative Nodes (DARP-PV-AN). This model allows the combination of two fleets: a public fleet of vehicles located at a depot and a set of private vehicles, each one owned by a client. The former fleet corresponds to a centralized system (transportation company, either public or private) while the latter is decentralized. The pickup and the delivery locations for each transportation request are associated to an initial node as well as to several alternative nodes. These alternative nodes are used to improve the client's privacy. The initial node is used by the public fleet, while the alternative nodes are used by private vehicles (except for the pickup and the delivery of the vehicle's owner). Alternative nodes prevent from learning one-colleague locations, i.e. keeping both original and final location private. As an additional feature, these alternative nodes allow the definition of meeting nodes by sharing some alternative nodes between several transportations requests. A mixed integer linear programming formulation has been proposed and benchmarked as well as an evolutionary local search (ELS) metaheuristic. A new set of instances, dedicated to the DARP-PV-AN, has been created too. Results show that the use of private vehicles can lead to some improvement on several nodes. First, the sum of travelled distances is significantly reduced since the private vehicles do not need to leave and return to the depot. The quality of service is improved and the travel distance for each client is closer to the shortest distance. In addition, the number of arcs used is reduced, which means a lower impact on the global traffic. Coupled with a lower total distance, this could also lead to lower global NOx/CO2 emissions. This work is currently being extended into the analysis of the economical interplay between both types of services. Namely, given a company, we are looking on the way to set financial incentives for the employees to offer ridesharing opportunities to their colleagues. Moreover, the coupling between ridesharing and multimodal transportation should be further investigated as well as a generalization of the optimal alternative node selection and trip evaluation.

6. References

Aïvodji, U. M., S. Gambs, M.-J. Huguet, and M.-O. Killijian. "Meeting nodes in ridesharing: A privacy-preserving approach." *Transportation Research Part C: Emerging Technologies* 72: 239-253, 2016.

Artigues, C., Y. Deswarte, J. Guiochet, M.-J. Huguet, M.-O. Killijian, D. Powell, M. Roy, C. Bidan, N. Prigent, E. Anceaume, S. Gambs, G. Guette, M. Hurfin, F. Schettini. "Amores: an architecture for mobiquitous resilient systems." *Proceedings of the 1st European Workshop on AppRoaches to MObiquiTous Resilience*. ACM, 2012.

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

Battarra, M., J.-F. Cordeau, and M. Iori, "Pickup and Delivery Problems for Goods Transportation", in Toth P, Vigo D editors, Vehicle Routing: Problems, Methods, and Applications, 2nd edition, SIAM, Monographs on Discrete Mathematics and Applications, ISBN 978-1-611973-58-7, pp. 161-192 (2014).

Braekers, K., A. Caris, and G. K. Janssens. "Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots." *Transportation Research Part B: Methodological* 67: 166-186, 2014.

Bruck, P.B., V. Incerti, M. Iori, and M. Vignoli, "Minimizing CO_2 emissions in a practical daily carpooling problem", Computers & Operations Research 81: 40-50 (2017).

Chassaing, M., C. Duhamel, and P. Lacomme. "An ELS-based approach with dynamic probabilities management in local search for the Dial-A-Ride Problem." *Engineering Applications of Artificial Intelligence* 48: 119-133, 2016.

Cordeau, J.-F., and G. Laporte. "A tabu search heuristic for the static multi-vehicle dial-a-ride problem." *Transportation Research Part B: Methodological* 37(6): 579-594, 2003.

Cottrill, C. D. "Location privacy preferences: A survey-based analysis of consumer awareness, trade-off and decision-making." *Transportation Research Part C: Emerging Technologies* 56: 132-148, 2015.

Desrochers, Martin, and François Soumis. "A generalized permanent labelling algorithm for the shortest path problem with time windows." *INFOR: Information Systems and Operational Research* 26.3 (1988): 191-212.

Dijkstra, E. W. "A note on two problems in connexion with graphs." *Numerische mathematik* 1(1): 269-271, 1959.

Doerner, K.F., and J.J. Salazar Gonzalez, "Pickup-and-Delivery Problems for People Transportation", in Toth P, Vigo D editors, Vehicle Routing: Problems, Methods, and Applications, 2nd edition, SIAM, Monographs on Discrete Mathematics and Applications, ISBN 978-1-611973-58-7, pp. 193-212 (2014).

Dongarra, J., 2014. Performance of Various Computers Using Standard Linear Equations Software, (Linpack Benchmark Report), University of Tennessee Computer Science Technical Report, CS-89-85.

Duhamel, C., P. Lacomme, C. Prodhon, and C. Prins. "A GRASP×ELS approach for real-life location routing problems." *Proceedings of the 2009 International Conference on Computers & Industrial Engineering*, IEEE, 2009.

Firat, M., and G. J. Woeginger. "Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh." *Operations Research Letters* 39(1): 32-35, 2011.

Furuhata, M., M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig. "Ridesharing: The state-of-the-art and future directions." *Transportation Research Part B: Methodological* 57: 28-46, 2013.

Gambs, S., M.-O. Killijian, and M. Núñez del Prado Cortez. "Show me how you move and I will tell you who you are." *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*. ACM, 2010.

Lin, S., and B. W. Kernighan. "An effective heuristic algorithm for the traveling-salesman problem." *Operations Research* 21(2): 498-516, 1973.

Lourenço, H. R., O. C. Martin, and T. Stützle. "Iterated local search." *Handbook of metaheuristics*. Springer, Boston, MA, 320-353, 2003.

Masmoudi, M. A., K. Braekers, M. Masmoudi, and A. Dammak. "A hybrid genetic algorithm for the heterogeneous dial-a-ride problem." *Computers & Operations Research* 81: 1-13, 2017.

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

Parragh, S. N., K. F. Doerner, and R. F. Hartl. "Variable neighborhood search for the dial-a-ride problem." *Computers & Operations Research* 37(6): 1129-1138, 2010.

Pecin, D., A. Pessoa, M. Poggi, E. Uchoa, "Improved branch-cut-and-price for capacitated vehicle routing", Mathematical Programming Computation 9 (1), 61-100, 2017.

Potvin, J.-Y., and J.-M. Rousseau. "A parallel route building algorithm for the vehicle routing and scheduling problem with time windows." *European Journal of Operational Research* 66(3): 331-340, 1993.

Prins, C. "A simple and effective evolutionary algorithm for the vehicle routing problem." *Computers & Operations Research* 31(12): 1985-2002, 2004.

Prins, C. "A GRASP× evolutionary local search hybrid for the vehicle routing problem." *Bio-inspired algorithms for the vehicle routing problem*. Springer, Berlin, Heidelberg, 35-53, 2009.

Quercia, D., N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft. "Recommending social events from mobile phone location data." *Data Mining (ICDM), Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM '10)*, 2010.

Ropke, S., J.-F. Cordeau, and G. Laporte. "Models and branch-and-cut algorithms for pickup and delivery problems with time windows." *Networks* 49(4): 258-272, 2007.

Vidal, T., T.G. Crainic, M. Gendreau, W. Rei, N. Lahrichi, "A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems", Operations Research 60 (3), 611-624, 2012.

Wolf, S., and O. Merz. "Evolutionary local search for the super-peer selection problem and the p-hub median problem." *International Workshop on Hybrid Metaheuristics*. Springer, Berlin, Heidelberg, 2007.