Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 52 (2021) 549–556

Transportation Research Procedia

www.elsevier.com/locate/procedia

23rd EURO Working Group on Transportation Meeting, EWGT 2020, 16-18 September 2020, Paphos, Cyprus

Assessment of a Shared-Taxi Routing Service for Disabled People: Barcelona Case Study

Gonzalo Recio[a], MªPaz Linares[a,*], Víctor Diví[a], Josep Casanovas[a,b]

[a]Universitat Politècnica de Catalunya, Barcelona 08034, Spain

[b]Barcelona Supercomputing Center, Barcelona 08034, Spain

Abstract

Nowadays, multiple passenger ridesharing and its variants look one of the more promising emerging mobility concepts. However, the real implementation of these systems accounting specifically requirements for users should raise certain challenges, even further for disabled people, which inherits requirements that need special consideration: short ride times, specific vehicle characteristics depending on the mobility handicaps, narrow time windows constraints, etc. This paper presents the real case study of the public transport service that Barcelona city offers to people with reduced mobility, which could program weekly taxi trips within the city at a specific time. The proposed routing algorithm integrated into this service management system is based on a tabu search heuristic approach used to minimize the dimension of the heterogeneous fleet and the total traveled time. Furthermore, this work exhaustively analyses the operational factors of this mobility service to analyze how they affect the service performance and car-sharing factors. The obtained results show that certain operational decisions could make better use of the resources allocated to the sharing services.

Keywords: ride-sharing; routing; tabu search; disabled people.

1. Introduction

In the analysis of the Future of Mobility and New Mobility Business Models, Sullivant and Frost (2015) identify the growing trend of "Ride Sharing" models as one of the consequences of the paradigmatic shift from "car ownership" to "vehicle usage". Furthermore, according to the definition of the European Commission, Demand Responsive Transport (DRT), Dial-a-Ride Transit or Flexible Transport Services are emerging user-oriented forms of public transport characterized by flexible routing and scheduling of small/medium vehicles operating in shared-ride mode between pickup and drop-off locations according to passenger needs.

* Corresponding author: *E-mail address:*mari.paz.linares@upc.edu

10.1016/j.trpro.2021.01.065

A state of the art survey on the variants of ridesharing systems, their alternatives, and likely future evolution can be found in Furuhata et al. (2013). More recently, Mourad et al. (2019) present an exhaustive survey of models and algorithms for optimizing shared mobility. According to their classification, the variant solved in this paper corresponds to what is known as Multidepot Heterogeneous Dial-a-Ride Problem (MD-H-DARP) with heterogeneous users. Although this problem has hardly been studied, Braekers et al. (2014), it is highly relevant since these problem characteristics often jointly arise in practice.

Our research concretely addresses the Dial-a-Ride Problem, whose common application is the door-to-door transportation of elderly and disabled people, which often cannot make use of standard public transportation services because these are not adapted to their needs. The main concern of this mobility service is that adapted vehicles suppose a huge expense and resources are limited. Thus, the main goal is to optimize routes in order to take more advantage of the allocated resources and increase car-sharing among users to minimize service costs.

This contribution is organized as follows. Firstly, Section 2 summarizes a literature review of the Dial-a-Ride problem using non-exact approaches. Then, Section 3 specifies the mathematical formulation of the problem as an optimization model, and Section 4 summarizes the metaheuristic proposed to solve real instances of the problem. Then, Section 5 presents the case study in Barcelona, including the computational experiences proposed to evaluate the presented KPIs, and the obtained results. Finally, Section 6 describes the conclusions and some future research.

2. Literature Review

The problem to tackle is a generalization of the well-known Dial-a-Ride Problem (DARP) or Vehicle Routing Problem (VRP) with user pickup and delivery constraints and time-windows restrictions. This problem is NP-hard in the strong sense, and exact methods are impracticable for real scenarios with hundreds of daily users and, hence, approaches such as linear integer programming to compute optimal solutions should be discarded. Even so, heuristics and metaheuristics have shown great results and performance when dealing with these hard-constrained problems.

The first remarkable heuristics to solve the DARP appeared with the proposal of Bodin and Sexton (1986) and Borndörfer et al. (1997), where they first cluster and then route all the nodes. Later, we can find well-known improvement heuristics such as the 2-opt Exchange, proposed by Potvin and Rousseau (1993), or the lambda-exchange introduced by Osman (1993). Finally, in Diana and Dessouky (2004), a regret insertion heuristic is used to solve big DARP instances. However, most of these heuristics have limitations on certain aspects (e.g. assuming divisible loads, or only considering goods deliveries) that make it difficult to generalize to our MD-H-DARP.

Years after, metaheuristics appeared to achieve significant results for solving this problem using methods based on simulated annealing, genetic algorithms, and tabu search approaches. In the literature, we can find the work of Toth and Vigo (1996) in which they solve a MD-H-DARP using an insertion heuristic and a tabu search to improve the solution, similarly to Lau and Liang (2002). Besides, Li and Lim (2001) presented a tabu search with simulated annealing-based restarts. Cordeau et al. (2007) also used a tabu search proposal to obtain good results. Finally, Atahran, Lenté, and T'kindt (2014) proposed a genetic algorithm for the H-DARP, and Braekers, Caris, and Janssems (2014) proposed a deterministic annealing algorithm to solve the MD-H-DARP.

3. Problem formulation

The problem to solve is a minimization problem, whose objective function can be set up to minimize the fleet, the travel distance, the travel time, etc. In our case, we are interested in minimizing firstly the number of used vehicles and secondly the traveled distance. We are going to formulate the problem as an optimization model, using integer linear programming notation to be aware of the constraints needed for our heuristic, and for having a baseline optimal solution for some small instances. To do so, we start from the notation and formulation proposed in the work of Toth and Vigo (2001). In DARP, we can represent the problem as a fully-connected bidirectional graph in which vertices represent pickup and delivery points, and edges are the paths that vehicles travel between points. More formally, given n requests, a request i is identified with two nodes, i and $n+i$, corresponding to pickup and delivery nodes, respectively. We also define the set of pickup nodes $P = \{1,\dots,n\}$ and the set of delivery nodes $D = \{n+1,\dots,2n\}$, and $N = P \cup D$. If a request consists in transporting d_i people from node i to node $n+i$, let us define $l_i = d_i$ and

$l_{n+i} = -d_i$ as the load that the vehicle takes or leaves at each node. In addition, each node $i \in V$ is associated with a time window $[a_i, b_i]$ and a service time s_i to let the user enter or leave the vehicle.

Defining K as the set of vehicles, given that not all vehicle types can serve all types of user (recall that some users can only use adapted vehicles), each vehicle $k \in K$ has a specific associated set $N_k = P_k \cup D_k$,where N_k , P_k and D_k are the appropriate subsets of N , P and D , respectively. For each vehicle k , we also define a graph $G_k = (V_k, A_k)$, having $V_k = N_k \cup \{o(k), d(k)\}$ and $A_k \subseteq V_k \times V_k$, where $o(k)$ is its origin depot node and $d(k)$ its destination depot node (in our case $o(k) = 0$ and $d(k) = 2n+1$ for all the vehicles). Also, each vehicle is associated to a travel distance matrix (c_{ijk}) and a travel time matrix (t_{ijk}), defining respectively the distance and time it takes for vehicle k to travel from i to j . In our problem, distance is identified with cost, so c_{ijk} is also called the cost matrix.

The variables of the problem are x_{ijk} , L_{ik} and T_{ik} . Binary variable x_{ijk} is 1 if edge (i, j) is used by vehicle k and 0 otherwise. Variable T_{ik} denotes at what time vehicle k starts serving at node $i \in V_k$, and L_{ik} denotes the load that vehicle k after serving request at node $i \in V_k$. Note that L_{ik} and T_{ik} are positive integer variables ($T_{ik}, L_{ik} \in \mathbb{Z}^+$).

$$\text{(VRPPDTW/MD-H-DARP)} \qquad \min \sum_{k \in K} \sum_{(i,j) \in A_k} c_{ijk} x_{ijk} \qquad (1)$$

subject to:

$$\sum_{k \in K} \sum_{j \in N_k \cup \{d(k)\}} x_{ijk} = 1 \qquad \forall i \in N \qquad (2)$$

$$\sum_{j \in N_k} x_{ijk} - \sum_{j \in N_k} x_{j,n+i,k} = 0 \qquad \forall k \in K, i \in P_k \qquad (3)$$

$$\sum_{j \in P_k \cup \{d(k)\}} x_{o(k)jk} = 1 \qquad \forall k \in K \qquad (4)$$

$$\sum_{i \in N_k \cup \{o(k)\}} x_{ijk} - \sum_{i \in N_k \cup \{d(k)\}} x_{jik} = 0 \qquad \forall k \in K,\ j \in N_k \qquad (5)$$

$$\sum_{i \in D_k \cup \{o(k)\}} x_{i,d(k),k} = 1 \qquad \forall k \in K \qquad (6)$$

$$x_{ijk}(T_{ik} + s_i + t_{ij} - T_{jk}) \le 0 \qquad \forall k \in K, (i,j) \in A_k \qquad (7)$$

$$a_i \le T_{ik} \le b_i \qquad \forall k \in K, i \in V_k \qquad (8)$$

$$T_{ik} + t_{i,n+i,k} \le T_{n+i,k} \qquad \forall k \in K, i \in P_k \qquad (9)$$

$$x_{ijk}(L_{ik} + l_j - L_{jk}) = 0 \qquad \forall k \in K, (i,j) \in A_k \qquad (10)$$

$$l_i \le L_{ik} \le C_k \qquad \forall k \in K, i \in P_k \qquad (11)$$

$$0 \le L_{n+i,k} \le C_k - l_i \qquad \forall k \in K, n+i \in D_k \qquad (12)$$

$$L_{o(k),k} = 0 \qquad \forall k \in K \qquad (13)$$

$$x_{ijk} \in \{0,1\} \qquad \forall k \in K, (i,j) \in A_k \qquad (14)$$

The objective function (1) minimizes the total route cost. Restrictions (2) and (3) impose that each request is served only once and by the same vehicle. Constraints (4)-(6) ensure the flow restrictions: vehicles start and finish at their corresponding depots, and when they enter a node, they leave it). Restrictions (7)-(9) are in charge of fulfilling temporal limits, time-windows, and precedence between PD-pairs. Furthermore, restriction (7) allows a vehicle to wait before visiting a node, in case it arrives before the time-window opens. Last restrictions, (10)-(13), ensure that the vehicle capacity is not surpassed along the routes.

To minimize the fleet size along with the route cost, a high cost is set to edges that leave the depot (i.e. $c_{o(k),j,k} = 10^{10}$ for all $j \in P_k$), so that it is very costly that a vehicle enters into the network. Then we need to include the edge $(o(k), d(k)) \in A_k$, $\forall k \in K$ with cost 0, to allow vehicles to not be used (going directly from depot to depot).

However, this model is not linear, and, therefore, not solvable by integer linear programming solvers. In this work, we propose a way to linearize the model. In particular, constraints (7) and (10) are not linear, involving variables multiplied. Restriction (7) can be linearized by using the *big-M* method, as it is an inequality of the type $xy \le 0$. Given a big enough *M*, the constraint can be expressed as $y \le (1-x)M$, being $x \in \{0,1\}$. Hence, equation (7) can be rewritten as follows:

$$(T_{ik} + s_i + t_{ij} - T_{jk}) \le M(1 - x_{ijk}) \qquad \forall k \in K, (i,j) \in A_k \qquad (15)$$

and, concretely, *M* can take as value $max(b_i + s_i + t_{ij} - a_j, 0)$, given that $a_i \le T_{ik} \le b_i$, $\forall i, k$. Same idea is applied to restriction (10), by using two restrictions as it is a strict equality of the type $xy = 0$ and considering that $x_{ijk} \in \{0,1\}$:

$$-M(1 - x_{ijk}) \le L_{ik} + l_j - L_{jk} \le M(1 - x_{ijk}) \qquad \forall k \in K, (i,j) \in A_k \qquad (16)$$

With these modifications of constraints (7) and (10), we have a completely integer linear model.

4. Metaheuristic: Tabu search-based algorithm

An implementation of a heuristic algorithm is needed for solving daily scenarios in a reasonable time (like the proposed use case in Barcelona). The heuristic algorithm constructs an initial greedy solution assigning user trips to the minimum number of vehicles possible on a single or few iterations over the user requests. This kind of insertion heuristics can be found in the literature of the MD-H-DARP and its variants (Toth and Vigo (1996) and Lau and Liang (2002)), and are very helpful for obtaining good initial solutions and speeding up the optimization process.

This initial solution is then improved with a tabu search, Glover (1995), a very popular and effective method for solving combinatorial problems. Tabu search is a local search algorithm that uses a tabu list data structure to *prohibit* ("tabu") the use of some operators for a certain number of iterations to prevent the search from undoing improvement movements and going back to previously-visited solutions. The optimization process is performed by exploring the solution space with the traditionally used search operators: Within Route Insertion (WRI), Single Pair Insertion (SPI), and Double Pair Insertion (DPI), Toth and Vigo (1996). We observed that another common operator, Swap Between Routes (SBR), was hardly ever selected for improvement while its computational cost was quite high.

After several trials and executions, we found out that the DPI operator helped to reduce a lot the cost function (number of used vehicles) at the beginning, and, in later iterations, it was hardly ever used. For this reason, a greedy method has been implemented as a previous phase to the tabu search that uses the DPI operator trying to put as many users in the minimum possible number of routes. It works by taking routes with fewer users and trying to place their requests on routes that already have more users.

Pseudocode for **DPI_greedy** (previous phase to Tabu Search)

$S \leftarrow S_0$ initial solution
$k \leftarrow length(routes)$
while max. iterations not reached & getting improvement **do**
    routes ← increasingly sorted routes of *S* by their size
    $half1 \leftarrow routes[1:k/2]$
    $half2 \leftarrow routes[k/2:end]$
    $S \leftarrow \mathbf{DPI}(S,i,j,k)$, where $i, j \in half1$ and $k \in half2$
end while
return *S*

Therefore, the final proposed algorithm begins with the insertion algorithm to start with a good solution, then DPI is greedily used in order to leave empty routes, and finally, tabu search, which implies a higher computational cost, is used to improve an already good initial solution using with all the search operators.

Pseudocode for Tabu search-based metaheuristic

$S \leftarrow S_0$ initial solution with **Insertion Heuristic**
$tabuList \leftarrow \varnothing$
while getting improvement **do** ► Greedy DPI
    $S \leftarrow$ **DPI_greedy**(S)
    $S \leftarrow \mathbf{WRI}(S)$
end while
while not reached max. iteration threshold & getting improvement **do** ► Tabu Search
    $S \leftarrow \mathbf{WRI}(S)$
    Generate neighborhood $\mathcal{N}(S) = \mathbf{SPI}(S) \cup \mathbf{DPI}(S)$
    Find *best* solution $S_{best} \in \mathcal{N}(S)$ such that the operator producing $S_{best} \notin tabuList$
    Add generation operator of S_{best} to $tabuList$ and Update $tabuList$
    $S \leftarrow S_{best}$
end while
return *S*

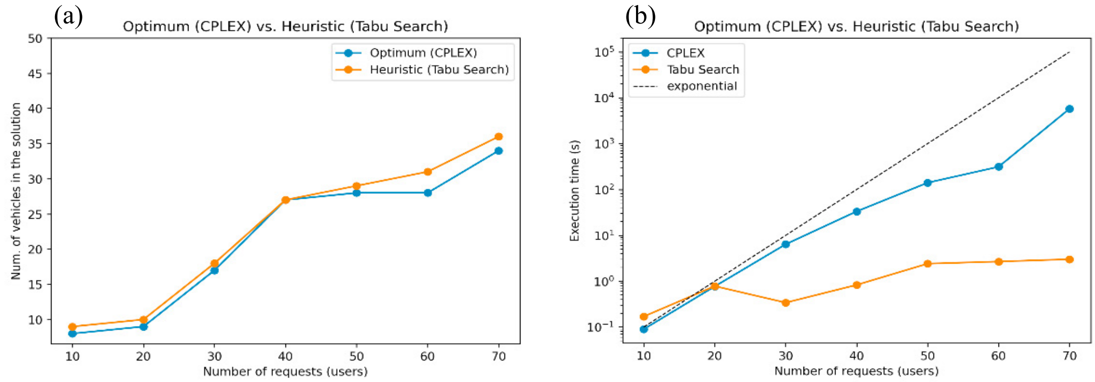

Fig. 1. (a) Minimum number of vehicles computed by each method. (b) Execution times of the methods (in seconds).

Finally, we study the quality of our tabu search metaheuristic results by comparing them with the optimal solutions computed with CPLEX. To do so, we translated the linear integer programming formulation (Section 3) into AMPL and generated small dimension instances of the problem up to 70 users with narrow time windows (5min) and small service times (1min); with larger configurations, exact solutions are practically incomputable. For instance, setting a time-window of 10min increases a lot the possible combinations so that CPLEX solver runs out of memory in powerful computers (i.e. 64GB of RAM), and execution times increase disproportionally.

Regarding the computational results, figure 1(a) shows that the heuristic solutions are close to the optimal minimum computed with CPLEX (within 0% and 9% of difference). Also, in figure 1(b), we can observe that CPLEX execution times grow almost exponentially with the number of user requests (exceeding 2h for instances with 70 users) while the heuristic algorithm remains under 10 seconds of execution (note the logarithmic y-axis).

Considering the trade-off between execution time and result quality, the proposed heuristic is definitely suitable for calculating optimized routes in real problem scenarios with hundreds of users within a reasonable time. Additionally, we have seen that scenarios with larger TW than 5min are combinatorically incomputable for exact optimizers, justifying the need for a heuristic approach for real scenarios with more than 100 users.

5. Case Study: Barcelona

5.1. Scenario

The scenario of the mobility service is the traffic network of Barcelona, in particular, the Public Service of Special Transport of Barcelona offered by the city council, in which users can program weekly taxi trips within the city at a specific time. We remark that this service is addressed to people with reduced mobility that, depending on their characteristics, they may need adapted vehicles or not. Given that, we are going to generate artificial instances of user requests following the demand patterns of the current service (~10% of users need adapted vehicles). Also, we are going to consider the traffic network costs (travel distance and time) of the city of Barcelona.

5.2. Experiment design

Since the main concern is to optimize the cost of this public service intended for a sensible collective while preserving service conformity, our interest is to evaluate the impact of operational parameters, which are mainly the time-windows (TW) and the service times (ST). Both parameters are related to the quality of the service: narrow time-windows mean that pickup and delivery times are less uncertain, which is positive for the user. On the other hand, a large service time should be favorable for the user, as this is time reserved for the user to accommodate and settle in the vehicle (we remark that the system is for elder and disabled people and thus service time can easily take more than a minute). For this reason, we want to study up to which point tuning these operational parameters can help to reduce service cost (and with what magnitude), and help to find an adequate trade-off between service quality and saving service costs. We are going to perform a factorial experiment design setting different levels for the number of users, TW, ST, and time-horizon (H) which is a parameter of scenario definition and simulates the temporal-congestion of user requests. Factorial design levels are the following:

- Number of users: 100, 150.
- Time-windows (TW): 5 min, 10 min, 20 min.
- Service time (ST): 2 min, 4 min.
- Time horizon (H): 1 hour, 2 hours.

5.3. Key Performance Indicators (KPI)

From the previous experiments, we extracted several KPIs (Key Performance Indicator) that will help to evaluate the algorithm performance and solution quality with respect to operational parameters. Regarding that the main goal is to save resources and incentivize car-sharing, the proposed KPIs are mainly related to vehicle usage, occupation, and traveled distance:

- **Number of used vehicles**: this is the main objective function that we want to minimize.
- **Mean occupation** (by distance unit): distance weighted by the load carried by the vehicles divided the distance traveled by all the vehicles. This KPI gives an idea of the occupation along the routes, that is, if all the vehicles drive empty, the value of this KPI would be 0, on the other hand, if the vehicles are full all the time, the value will be 4 (max. capacity). Following the notation in Section 3, the KPI is computed as follows:

$$\frac{\textit{Total travelled dist. weighted by vehicle loads}}{\textit{Total travelled dist. by all the vehicles}} = \sum_{\forall k \in K} \sum_{\substack{(i,j) \in A_k \\ x_{ijk}=1}} c_{ij} L_{ik} \Big/ \sum_{\forall k \in K} \sum_{\substack{(i,j) \in A_k \\ x_{ijk}=1}} c_{ij}$$

- **Percentage (%) of car-shared distance**: the ratio of travel distance in which vehicles carry more than one passenger. High values are desired for this KPI.
- **Percentage (%) of empty car traveled distance**: the ratio of travel distance in which vehicles are empty. We are interested in achieving low values for this KPI.

5.4. Result Analysis

Table 1 shows the results of the experiments under the name {*num_users*}-{*time_horizon*}H-{*service_time*}ST-{*time_window*}TW. It presents the KPIs obtained from the computed solutions along with their execution time.

Table 1. Results and KPIs obtained from the experiment design.

Experiment name	Used vehs.	Mean occup.	% car-sharing distance	% empty car dist.	Exec. time
100-1H-2ST-5TW	62	0.92	0.001%	8.22%	13.33s
100-1H-2ST-10TW	51	1.07	11.86%	4.77%	11.63s
100-1H-2ST-20TW	40	1.35	34.36%	1.98%	13.02s
100-1H-4ST-5TW	70	0.92	0.0%	8.15%	9.74s
100-1H-4ST-10TW	59	0.91	1.3%	10.13%	10.04s
100-1H-4ST-20TW	46	1.21	27.06%	5.82%	7.10s
100-2H-2ST-5TW	35	0.8	0.0%	19.99%	14.62s
100-2H-2ST-10TW	33	0.88	7.88%	20.17%	15.28s
100-2H-2ST-20TW	25	1.1	23.4%	14.31%	11.66s
100-2H-4ST-5TW	40	0.81	0.0%	18.68%	17.34s
100-2H-4ST-10TW	37	0.78	0.5%	22.34%	16.68s
100-2H-4ST-20TW	30	1.06	22.13%	16.63%	12.06s

Experiment name	Used vehs.	Mean occup.	% car-sharing distance	% empty car dist.	Exec. time
150-1H-2ST-5TW	78	0.89	0.1%	10.94%	38.64s
150-1H-2ST-10TW	66	0.99	8.65%	9.44%	26.73s
150-1H-2ST-20TW	55	1.28	30.57%	4.66%	26.15s
150-1H-4ST-5TW	90	0.91	0.0%	9.42%	41.08s
150-1H-4ST-10TW	78	0.91	1.69%	10.88%	39.40s
150-1H-4ST-20TW	63	1.2	24.0%	4.72%	17.04s
150-2H-2ST-5TW	48	0.77	0.05%	23.22%	28.17s
150-2H-2ST-10TW	42	0.86	6.69%	20.49%	21.82s
150-2H-2ST-20TW	37	1.08	21.88%	14.78%	28.30s
150-2H-4ST-5TW	53	0.77	0.0%	23.24%	25.92s
150-2H-4ST-10TW	46	0.73	0.76%	27.69%	10.24s
150-2H-4ST-20TW	41	0.98	15.86%	17.59%	28.67s

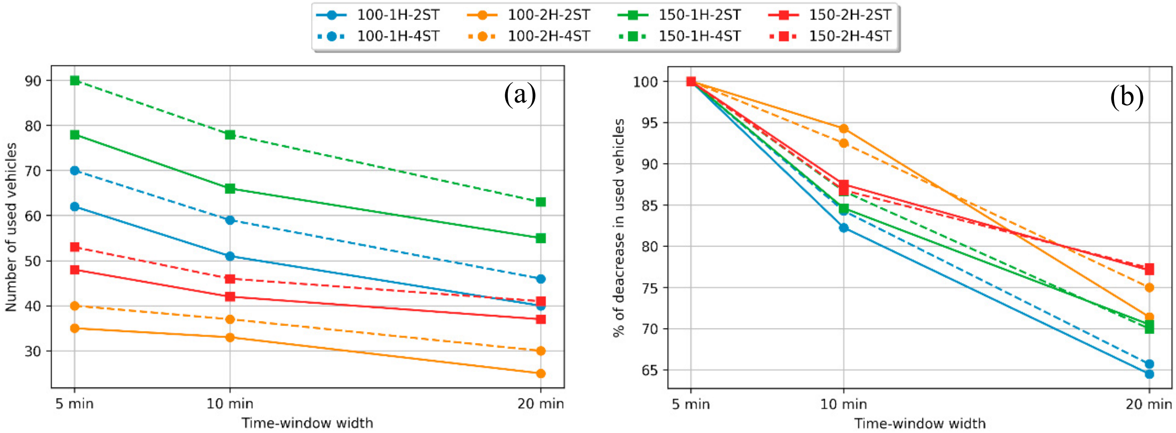

Fig. 2. Number of used vehicles along the time-window sizes. (a) Number of used vehicles; (b) Relative decrease of number of used vehicles.

Observing the results, we note that the time-window parameter (TW) plays the most significant role in minimizing the cost (minimizing the fleet) and improving KPIs. To ease the numerical analysis, we have plotted the KPIs of each experiment along with the variation of the time-windows size. Figure 2(a) shows how the number of used vehicles decreases when the time-windows are widened. Also, in figure 2(a), we can observe that smaller service times are indeed better for reducing fleet sizes (continuous lines are always under the dashed lines), such that setting 2min of ST supposes a ~15% fewer vehicles than having 4min of ST. In figure 2(b), we can see the relative improvement of vehicle usage with respect to the 5min TW: using larger TW of 20min reduces the required fleet in a ~30% compared with the baseline of 5min TW.

Figure 3(a) illustrates the relative improvement of mean occupation and shows that 20min TWs achieve an increase of 30-40% in the mean of users on vehicles per distance unit compared to 5min TWs. Furthermore, in figure 3(b) we notice that using 5min TW, sharing KPIs are very low, having around 0% of car-shared traveled distance, while with 20min TW, we achieve a 20-30% of car-shared traveled distance.

Another important observation is that, when the service time (ST) is large (of 4min, represented with dashed lines), solutions are worse in terms of car-sharing. Figures 3(a), 3(b) and 3(c) show a clear elbow when TW is of 10min and ST is of 4min (dashed lines), illustrating that there is no clear improvement (or it gets even worse) in car-sharing related KPIs in comparison to 5min TW. This means that high service times constrain the problem a lot, such that it is less feasible to make users share vehicles, and makes improvement difficult even if we use wider TWs. That is because the system (and, hence, the optimization algorithm) considers that the vehicles need to be stopped at least that amount of time, and this reduces the exploration of combination space, in other words, it makes the problem more constrained.

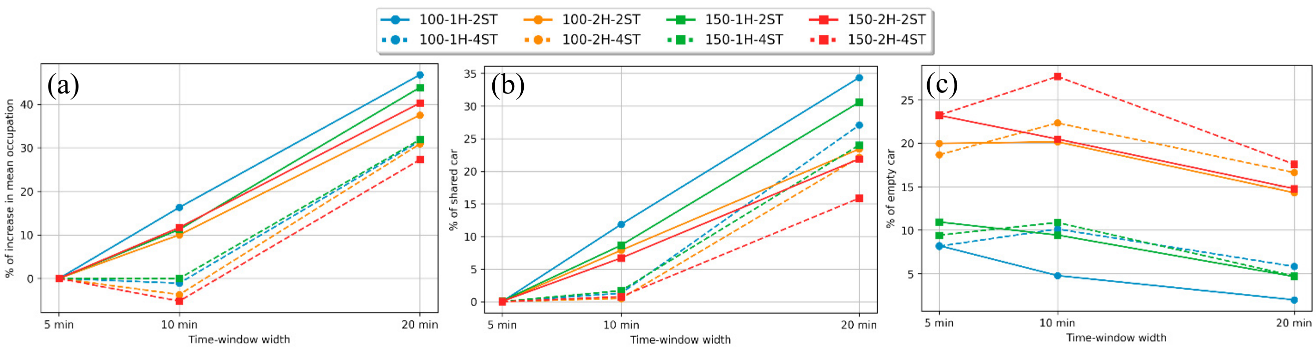

Fig. 3. KPIs for the proposed time-windows sizes: (a) Relative increase of mean vehicle occupation. (b) Percentage of car-sharing distance. (c) Percentage of empty car distance.

6. Conclusions

This work presents an integer linear programming model for solving the MD-H-DARP/VRPPDTW with optimum solutions (minimizing the fleet size along with route costs), and a tabu search metaheuristic to find near-optimal solutions in a reasonable time. The results obtained from the heuristic are close to the optimal solutions computed by CPLEX (from 0%-9%). Furthermore, the execution times and resources required with CPLEX grow exponentially with the size of the instances, justifying the need for a heuristic approach for real scenarios with hundreds of users.

A factorial experimental design has been proposed to evaluate the impact of operational parameters (such as time-windows and service-time) as factors of the solution quality. Our research shows that the time-window parameter (TW) plays a significant role in minimizing the service cost (minimizing the fleet size). Using a 20min TW reduces the required fleet size in a ~30% vs. using 5min TW. Also, with 5min TW, car-sharing KPIs are very low, while with 20min TW, we obtain a 20-30% increase of car-shared traveled distance and a 30-40% increase of the mean occupation, achieving much better shared-use of the vehicles. Besides, large service times (ST) penalize significantly the quality of the results which sometimes cannot be compensated by widening a bit the time-windows.

For future research, we could propose a more general formulation for the MD-H-DARP, allowing the definition of user subsets to solve scenarios where some user groups cannot share vehicles. Additionally, it would be interesting to work on solution post-processes to group users into pickup or delivery points to ease the door-to-door restriction.

Acknowledgements

Throughout this work, the authors have benefited from the support of inLab FIB at Universitat Politècnica de Catalunya. This research was funded by Secretaria d'Universitats i Recerca de la Generalitat de Catalunya (2017-SGR-1749).

References

Atahran, A., Lenté, C., & T'kindt, V. (2014). A Multicriteria Dial-a-Ride Problem with an Ecological Measure and Heterogeneous Vehicles. Journal of Multi-Criteria Decision Analysis, 21(5-6), 279-298.

Bodin, Lawrence & Sexton, Thomas. (1986). The multi-vehicle subscriber dial-a-ride problem. TIMS Studies in Management Science. 26.

Borndörfer, R., Klostermeier, F., Grötschel, M., and Küttner, C. (1997). Telebus Berlin: vehicle scheduling in a dial-a-ride system, Technical report SC 97-23, Konrad-Zuse-Zentrum für Informationstechnik, Berlin.

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

Crevier, B., Cordeau, J. F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. European journal of operational research, 176(2), 756-773.

Diana, M., & Dessouky, M. M. (2004). A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. Transportation Research Part B: Methodological, 38(6), 539-557.

Glover, F. (1995). Tabu search fundamentals and uses (pp. 1-85). Boulder: Graduate School of Business, University of Colorado.

Furuhata, M. et al. (2013). Ridesharing: The state-of-the-art and future directions. Transportation Research Part B 57, 28–46.

Lau, H., Liang, Z. (2002). Pickup and delivery problem with time windows. International Journal on Artificial Intelligence Tools 11 (03), 455-472.

Li, Haibing & Lim, Andrew. (2001). A Metaheuristic for the Pickup and Delivery Problem with Time Windows. International Journal on Artificial Intelligence Tools. 12. 160-167. 10.1109/ICTAI.2001.974461.

Mourad, A., Puchinger, J., Chu, C. (2019). A survey of models and algorithms for optimizing shared mobility. Transportation Research Part B 123, 323-346.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of operations research, 41(4), 421-451.

Potvin, J. Y., & Rousseau, J. M. (1995). An exchange heuristic for routing problems with time windows. Journal of the Operational Research Society, 46(12), 1433-1446.

Sullivant & Frost (2015). Intelligent Mobility 3.0 Future of Mobility & New Mobility Business Models. In Sullivan and Frost, ed. London. Available at: http://www.frost.com/sublib/displayreport.do?id=MB85-01-00-00-00.

Toth, P., Vigo, D. (1996). Fast local search algorithms for the handicapped persons transportation problem. (In book "Metaheuristics: Theory and Applications"), pp. 677-690. Kluwer Academic Publishers, 1996.

Toth, P., Vigo, D. (2002). The Vehicle Routing Problem (Monographs on Discrete Mathematics and Applications). SIAM, 2002.