Platoon Route Optimization for Picking up Automated Vehicles in an Urban Network

Mohamed Hadded, Jean-Marc Lasgouttes, Fawzi Nashashibi, Ilias Xydias

▶ To cite this version:

HAL Id: hal-01880388

https://inria.hal.science/hal-01880388

Submitted on 24 Sep 2018

Platoon Route Optimization for Picking up Automated Vehicles in an Urban Network[†]

Mohamed Hadded[1], Jean-Marc Lasgouttes[1], Fawzi Nashashibi[1], Ilias Xydias[1]

Abstract— In this paper, we consider the problem of vehicle collection assisted by a fleet manager where parked vehicles are collected and guided by fleet managers. Each platoon follows a calculated and optimized route to collect and guide the parked vehicles to their final destinations. The Platoon Route Optimization for Picking up Automated Vehicles problem, called PROPAV, consists in minimizing the collection duration, the number of platoons and the total energy required by the platoon leaders. We propose a formal definition of PROPAV as an integer linear programming problem, and then we show how to use the Non-dominated Sorting Genetic Algorithm II (NSGA-II), to deal with this multi-criteria optimization problem. Results in various configurations are presented to demonstrate the capabilities of NSGA-II to provide well-distributed Pareto-front solutions.

Index Terms— Automated Vehicles, Parking, Platoon, Multi-criteria optimization, NSGA-II.

I. INTRODUCTION

The rapid increase in the population of big cities (e.g. more than twelve million inhabitants in Paris urban area) has lead to increase in the number of cars, while the average daily use of each car is only $4\,\mathrm{km}$ [1]. In order to reduce greenhouse gas emissions, there is a need to reduce the number of privately owned vehicles. Carsharing services [2] are an alternative for people who may want to use their vehicles only once in a while, which can reduce transportation problems and lead toward environmentally friendly cities.

We distinguish between two types of car rental services, namely one-way and two-way systems [3]. In two-way systems, the user takes a vehicle at a rental station and returns it to its original station. In contrast, one-way systems allow the users to return the cars to any desired location, as long as it is indicated in advance. The vehicles are parked arbitrarily on the urban network by users, which is problematic since collecting vehicles using trucks is not often feasible in cities.

Following up on the work in [4], the French ANR VALET project proposes a novel approach to solve this redistribution problem using platoons of automated electric vehicles whose leaders are driven by professional drivers. An optimal routing algorithm is in charge of defining the platoon drivers' routes to the parking areas where the following cars are parked in a completely automated mode. The main idea of VALET is to retrieve vehicles parked randomly on the urban parking network by users. The vehicles may be parked in electric charging stations, car parks dedicated to car sharing vehicles or in regular parking places. Once the vehicles have been

collected and guided in a platooning mode, the objective is then to guide them to their allocated parking areas or to their respective parking lots. Then each vehicle is assigned a parking place into which it has to park in an automated mode.

In this work, we tackle the part of this approach that concerns the optimal allocation of vehicles to platoon tours. Each platoon follows an optimized route to collect and guide the parked vehicles back to their allocated parking area or to their respective original stations. The optimization takes into account 3 performance measures for this use case. First, the time needed to collect and guide all the vehicles to their original station (i.e. the longest platoon tour duration) should be minimized. Second, due to our assumptions on the energy costs, the total energy required by the platoon leaders and the parked vehicles to reach the rental station should be minimized. Third, the number of human drivers should also be minimized. Some other choices would have been possible, but the point of this paper is to show how the optimization can be implemented, and how it scales when the system becomes large.

The remainder of this paper starts with a short overview of related work. An Integer Linear Program formulation of the PROPAV problem is then presented in Section III. It can be solved using the multi-criteria optimization algorithm NSGA-II, as described in Section IV. Section V is devoted to extensive experimental results and performance evaluation. Finally, Section VI summarizes the main contribution of the paper and presents future directions.

II. RELATED WORK

As mentioned in the previous section, traditional vehicle sharing services are based on the two-way scheme, where the user picks up and returns a car to the same parking station. In recent years, services such as Car2Go [5] or Autolib [6] also permit the one-way scheme, where the user can return the vehicle to a different station. This scheme is much more attractive for users, but the service provider has to develop strategies to relocate the vehicles and restore an optimal distribution of the fleet. Currently, the vehicles are transported using trucks or human drivers. It should be mentioned that, in the carsharing relocation literature, the platooning of the vehicles has also been considered [7], where the platoon is composed of a chain of technologically innovative vehicles, led by a vehicle head. Furthermore, the problem becomes harder when the vehicles are electric.

In [8], the authors propose an algorithm to manage auto transport trucks which are used to collect the vehicles, based
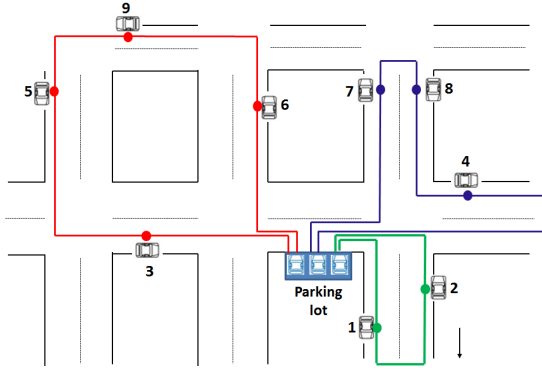
Fig. 1. An example solution to the PROPAV problem

on a Tabu search approach. In [9], the authors extend the previous work and, among the other things, minimize the total travel time of relocation, through three different heuristics. In [10] the authors design a system which is based on a three-step optimization-trend simulation for supporting vehicle sharing operators in relocating the vehicles. In [11], the authors consider that the relocation can be made by both users and staff, with a twofold objective: reducing the number of staff required and minimizing the number of shared vehicles to satisfy the demand. In [12], the authors consider the electric vehicle relocation problem as a pickup and delivery problem where the routes have to start from and end in a single depot without exceeding a given maximum time duration. In this paper, we consider the problem of parked vehicle collection assisted by multi-platoons, which presents many similarities to the Vehicle Routing Problem (VRP) [13]: platoon leaders correspond to vehicles, while parked vehicles that send requests to be picked up correspond to the customers to be served. However, there are differences in the objectives to optimize as well as the constraints, as we will see in the next section.

III. FORMALIZATION OF THE PROPAV PROBLEM

As shown in Fig. 1, given a set of vehicles parked somewhere in a city, one rental station and a set of human drivers, the PROPAV problem's output comprises the optimal routes to be followed by the human drivers to pick up the vehicles localized by their GPS positions. To define the PROPAV problem in a formal way, in this section we specify the data, the objective functions and the constraints.

Using the notations below, the PROPAV problem can be modeled as a graph $G = (V, E)$ where V is the set of vertexes representing the positions of the parked vehicles to be picked up by the head of a platoon, and E is the set of edges representing the shortest path between vehicle positions. In our study, we have used the Open Source Routing Machine (OSRM [14]) to determine the set E.

Let $\{1, \ldots, N\}$ be the set of parked vehicles to be retrieved by human drivers, where N is the number of parked vehicles plus the rental station. Vehicle i is geographically located at coordinates (x_i, y_i). By convention, vertex 0 is the rental station located at position (x_0, y_0). Let $D \geq 1$ be the number

of available drivers. The problem is to design a set of K tours $(r_1, \ldots, r_k)$, one platoon per tour with $1 \leq k \leq D$, that minimizes the longest platoon tour duration, the number of platoons, and the total energy consumed by the platoon leaders and the picked up vehicles under the constraints, that the vehicles have limited energy resources, the tour duration is limited by the energy constraint, each tour begins and ends at a rental station, each parked vehicle should be retrieved by exactly one platoon leader.

The inputs of the model are:
- Q gives the maximum number of cars that can be in the platoon;
- $d_{i,j}$ is the shortest distance required to travel from location i to location j;
- $e_{i,j}$ is the amount of energy (travel cost) required to travel between positions i and j (KWh). Most electric cars cover $100 \, \text{km}$ over $10 \, \text{kWh}$;
- R_i is the remaining range for a vehicle parked at location i, which estimates the distance that can still be covered with the energy remaining in the battery of the parked vehicle (see e.g. [15] for the evaluation of this value);
- ϑ_k is the average speed of the vehicle k.

and the decision variables (with a value of 0 or 1) are:
- X_{ij}^k is equal to 1 if the vehicle parked at position j is retrieved by the platoon leader k immediately after the vehicle parked at position i;
- Y_i^k is equal to 1 if the vehicle parked at position j is retrieved by the platoon leader k;
- Z_{ij}^t is equal to 1 if the vehicle that was parked at position t visited the location j immediately after location i.

Let D_k be the tour duration of the platoon leader k

$$D_k = \sum_{i=0}^{N} \sum_{j=0}^{N} d_{i,j} X_{ij}^k / \vartheta_k$$

and E_ℓ be the amount of energy consumed by vehicle ℓ to reach its final destination.

$$E_\ell = \sum_{k=0}^{D} \sum_{i=0}^{N} \sum_{j=0}^{N} e_{i,j} Z_{ij}^l Y_\ell^k$$

Objectives the following values should be minimized:
- longest platoon tour duration

$$\text{Minimize} \qquad T_T = \max_{k \in [0,D]} D_k, \qquad (1)$$

- total energy required for collection

$$\text{Minimize} \qquad T_E = \sum_{i=0}^{N} E_i, \qquad (2)$$

- number of human drivers

$$\text{Minimize} \qquad N_D = K. \qquad (3)$$

Constraints:
- all platoons have the same starting depot

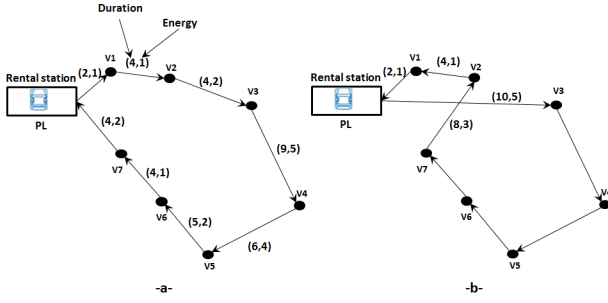$$\sum_{k=1}^{D} Y_i^k = 1 \quad i = 1 \qquad (4)$$

Fig. 2. Conflicting objective functions in the PROPAV problem

- each parked vehicle is retrieved by exactly one platoon leader

$$\sum_{k=1}^{D} Y_i^k = 1, \quad \forall i \in [1, N] \tag{5}$$

- residual stored energy constraint

$$\sum_{k=1}^{D} \sum_{i=0}^{N} \sum_{j=0}^{N} d_{i,j} Z_{ij}^t Y_t^k \leq R_i \quad \forall t \in [1, N] \tag{6}$$

- all platoons have at most Q vehicles

$$\sum_{i=1}^{N} Y_i^k \leq Q \quad \forall k \in [1, D] \tag{7}$$

- correct flow of cars through the edges

$$\sum_{i=0}^{N} \sum_{l=0}^{N} X_{il}^k = \sum_{l=0}^{N} \sum_{j=0}^{N} X_{lj}^k \quad \forall k \in [1, D] \tag{8}$$

- sub-tours are eliminated

$$\sum_{i=0}^{N} \sum_{j=0}^{N} X_{ij}^k = \sum_{i=1}^{N} Y_i^k \quad \forall k \in [1, D] \tag{9}$$

Therefore, the PROPAV problem is defined as follows: minimize (1)–(3) under the constraints (4)–(9) described above. However, the design of platoon-assisted parked vehicle pickup systems raises a problem of conflicting objective functions. For example, Fig. 2-a and Fig. 2-b illustrate two possible routes that can be used by the platoon leader (PL) to pick up the parked vehicles and guide them to the rental station. The first and the second weight value in each edge respectively give the link duration in $\min$ and the amount of energy required in kWh. The first route (Fig. 2-a) is the shortest tour duration, being $38\,\mathrm{min}$, but it is not optimal in terms of total energy consumed $87\,\mathrm{kWh}$, while the second route is the better in terms of total energy $73\,\mathrm{kWh}$ but its longest platoon tour duration reaches $48\,\mathrm{min}$. Due to the high number of possible solutions in the PROPAV problem and the conflicting nature of its objectives, there exists no unique optimal solution, but rather a set of solutions that form a Pareto front [16]. The next section describes this front and presents an algorithm able to produce it.

IV. MULTI-CRITERIA APPROACHES FOR PROPAV OPTIMIZATION

A. Overview of multi-objective optimization

Multi-objective optimization, also known as multi-criteria optimization, is part of multiple criteria decision-making (MCDM), which considers optimization problems involving more than one criterion to be optimized simultaneously. The general Multi-objective Optimization Problem (MOP) can be written as follows:

$$\min_{x \in \mathcal{D}} f_i(x), \quad i \in [1, m] \tag{MOP}$$

where the vector $x = (x_1, \ldots, x_n)^T \in \mathcal{D}$ contains the set of n decision variables, $f_i(x)$ is the objective function, m is the number of objectives, the vector $y = (y_1, y_2, \ldots, y_m)$ is a solution, with $y_i = f_i(x)$, and $\mathcal{D}$ is the feasible solution space. In MOP minimization, we say that solution $x \in \mathcal{D}$ is dominated by solution $x' \in \mathcal{D}$, denoted by $x' \prec x$, if and only if the following two conditions are satisfied:
  i) $f_i(x') \leq f_i(x), \quad \forall i \in [1, m]$
  ii) $\exists i \in [1, m]$ such that $f_i(x') < f_i(x)$

The set of the non-dominated solutions, often called the Pareto front, is denoted by PF:

$$PF = \{z \in \mathcal{D} \mid \nexists z' \in D, z' \prec z\}.$$

In other words, the Pareto front provides the best trade off values for the objective functions considered. NSGA-II [16] and MOPSO [17] are used to find the Pareto front for multi-objective optimization problems. These methods are respectively the multi-objective version of the genetic algorithm and particle swarm optimization in which the solutions explored are classified into Pareto-optimal fronts.

B. NSGA-II algorithm for the PROPAV problem

As shown in Fig. 3, NSGA-II starts from an initial population (P) of randomly generated solution vectors, called individuals. Then, a new population (Q_i) is generated at each iteration by using the crossover and mutation operators. After that, the two populations (P_i) and (Q_i) are merged together to form a single population R_i, which will be sorted based on the non-dominance principle $F_1, F_2, \ldots, F_n$. Finally, only the best solutions passing the selection step are included in the population of the next iteration (P_{i+1}) while the other individuals are deleted. If only some solutions of the last front F_l should be added to P_{i+1}, these solutions are chosen such that they have the best crowding distance. The aforementioned steps are repeated until the stop criterion is met.

In PROPAV, an individual represents a set of platoon tours where the i-th human driver follows the i-th tour. Each platoon tour is defined by the sequence of numbers of parked vehicles that will be retrieved by the same platoon leader. Each individual is divided into two parts, namely the vehicle-part and platoon-part. The first part is composed of a set of integers where each integer represents a parked vehicle retrieved by a platoon leader, and the second part contains platoon leader information. In the platoon-part, the i-th integer represents the number of parked vehicles retrieved by the
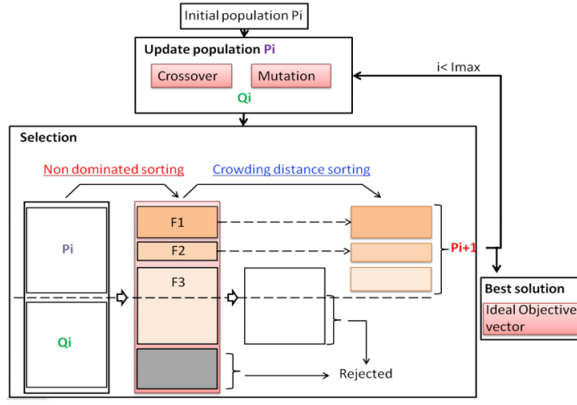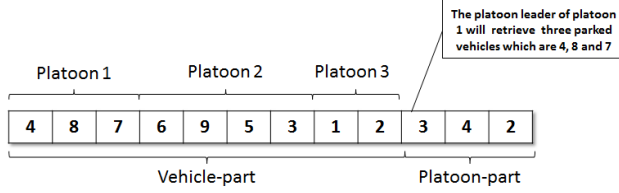
Fig. 3. Overview of the NSGA-II algorithm

Fig. 4. Representation of a possible solution as an individual, where 3 platoons are used, platoon leader 1 retrieves 3 vehicles (4, 8 and 7 in this order). Platoon leader 2 retrieves vehicles 6, 9, 5 and 3 in this order and platoon leader 3 retrieves vehicles 1 and 2.

i-th platoon leader (see Fig. 4 for an example). An individual is considered valid if and only if (1) each vehicle number occurs exactly once in the vehicle-part, (2) the sum of the integers included in the platoon-part is equal to the total number of parked vehicles to retrieve and (3) the size of the platoon-part is less than or equal to the number of available human drivers.

The crossover is one of the main operators used in the production step of NSGA-II to create a new population. It constructs two new child individuals by combining parts from a pair of parents (i.e. two solution vectors). After recombination, each new individual undergoes mutation operator which is applied with a probability less than 0.1. This operator is used to change the values of some genes randomly, and serves as a search strategy to prevent the algorithm from being trapped in local optima. An example of mutation is illustrated in Fig. 5.

V. SIMULATION RESULTS AND PERFORMANCE EVALUATION

In this section, we present and discuss the simulation results obtained by NSGA-II and we compare its performance to MOPSO [17], a popular algorithm for multi-criteria optimization problems. We implemented both NSGA-II and MOPSO using the Java programming language. Moreover, all our simulations were conducted using two Intel Core i7 computers running under Windows 10. We evaluated the performance of these algorithms using 4 configurations (see Table I) with different numbers of vehicles $\{50, 100, 150, 200\}$ located
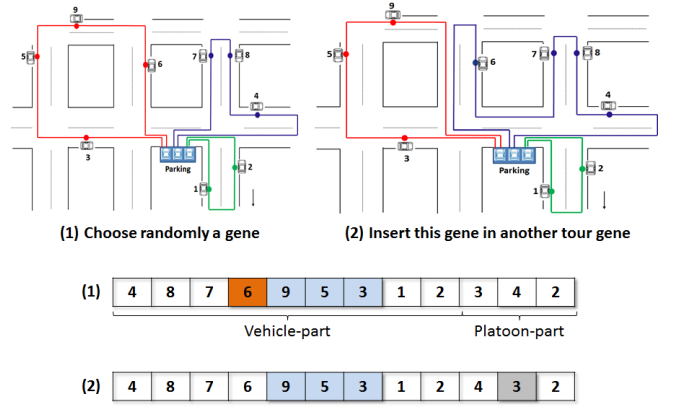

Fig. 5. An example of a mutation operator: The $4th$ gene of the vehicle-part of the individual suffers a mutation which corresponds to vehicle number 6 belonging to the second platoon. After mutation, vehicle 6 belongs to the first platoon (i.e. vehicle number 6 is retrieved by platoon leader 1). In order to make the mutated individual correct, the platoon-part of the individual is updated (-1 in the number of vehicles retrieved by platoon 2 and +1 for platoon 1).

TABLE I

SIMULATION SCENARIOS

Configuration	# of vehicles	Platoon size	# of platoons
Low (U1)	50	5	15
Medium (U2)	100	6	25
High (U3)	150	7	30
Very High (U4)	200	8	40

inside Paris, as shown in Fig. 6. As in [18], we conducted 15 independent simulation runs for each scenario using randomly selected initial populations. Table II shows the simulation parameters used in our experiments. We used the web API of the Open Source Routing Machine (OSRM [14]) to calculate the distances of the shortest paths between the different positions of the parked vehicles. For the all tests carried out, we fixed the computational time for the two algorithms to $10\,\mathrm{min}$ for all configurations under consideration.

Our goal is to determine the most suitable algorithm for the PROPAV problem among those that are very widely used in the area of multi-criteria optimization (e.g. NSGA-II and MOPSO). To do so, we evaluate the solutions provided by both NSGA-II and MOPSO in terms of the three fitness functions considered in the PROPAV problem. Each simulation produces a set of non-dominated solutions (Pareto front). Then, we build the final Pareto front of each scenario by

TABLE II

PARAMETER SETTINGS OF THE OPTIMIZATION ALGORITHMS

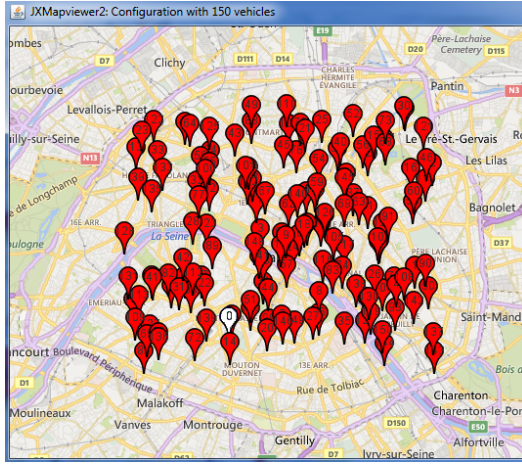| Algorithm	Parameter	Symbol	Value
NSGA-II	crossover probability	C_r	0.9
	mutation probability	μ	0.1
	population size	τ	250
MOPSO	local coefficient	φ_1	2.0
	social coefficient	φ_1	2.0
	inertia weight	w	0.5

Fig. 6. A configuration with one rental station (white) and 150 parked vehicles (red)
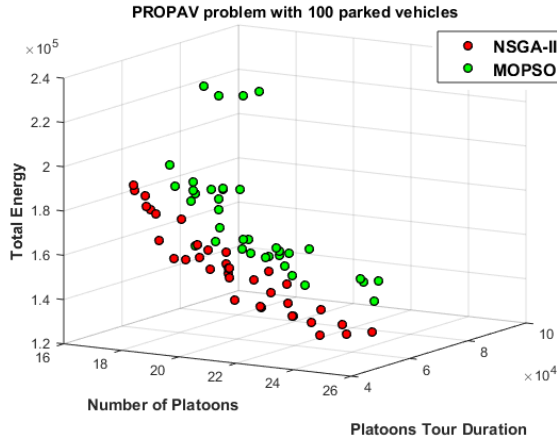
Fig. 8. Pareto front obtained by NSGA-II and MOPSO for the configuration U4.

Fig. 7. Pareto front obtained by NSGA-II and MOPSO for the configuration U2.

TABLE III
PERFORMANCE COMPARISON BETWEEN NSGA-II AND MOPSO IN TERMS OF RNS, SPACING AND SPREAD METRICS.

Scenario	MOEA	Ratio of non-dominated solutions	Spacing	Spread
U1	NSGA-II	0.14	12.26	216.98
	MOPSO	0.07	15.89	143.49
U2	NSGA-II	0.16	33.49	620.62
	MOPSO	0.07	43.12	373.27
U3	NSGA-II	0.19	32.23	861.27
	MOPSO	0.07	53.48	530.94
U4	NSGA-II	0.28	51.51	1656.6
	MOPSO	0.1	70.29	690

gathering all the non-dominated solutions previously obtained. Figures 7 and 8 plot the Pareto front obtained by gathering all the non-dominated solutions found by NSGA-II and MOPSO in the 15 independent runs corresponding to configurations with 100 and 200 vehicles.

The two figures show that the distribution of non-dominated solutions in the Pareto front found by NSGA-II is more localized than with MOPSO, which makes the choice of the suitable solution easier. In addition, we can note that NSGA-II provided the best approximation of the Pareto front.

In order to better compare the performance of these two algorithms, we evaluate the Pareto fronts (PF) obtained by NSGA-II and MOPSO in terms of the Ratio of Non-dominated Solutions (RNS), spacing [19] and spread [20]. The goal of this comparison is to demonstrate the effectiveness of NSGA-II on different configurations making it the best solution for the PROPAV problem. The third column of Table III shows the RNS ratio for the two algorithms. NSGA-II provides a Pareto front with more non-dominated solutions than MOPSO. For instance, in configuration U3, NSGA-II achieves an RNS of 19.03% in contrast to MOPSO which shows a rate of 6.84%
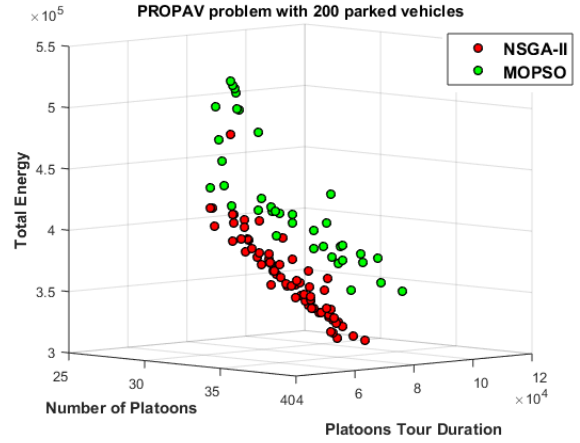
(i.e. approximately 64.06% lower than NSGA-II). Thus, the proposed approach gives a wide range of non-dominated solutions, whereas the MOPSO algorithm produces a Pareto front with a small number of solutions.

Table III also shows the average spacing and spread for NSGA-II and MOPSO, respectively. We can note from these two figures that the Pareto fronts obtained by NSGA-II are the best on all the scenarios considered in our experiments. For instance, for configuration U4, NSGA-II gives an average spacing of 51.513, whereas it reaches 7028.7 in MOPSO. Thus, the set of non-dominated solutions given by NSGA-II are closely spaced, which can simplify the choice of the best solution from the Pareto front obtained. Moreover, the results presented in Table III prove that NSGA-II is able to provide a Pareto optimal set with a widely spread distribution of solutions.

Table IV compares the best results in the Pareto Front sets given by NSGA-II and MOPSO for each configuration. 15 independent runs are performed for each algorithm. Then, we gathered the final optimal Pareto fronts to identify the global non-dominated solutions set. The best solution is the one that has the smallest normalized Euclidean distance to

TABLE IV

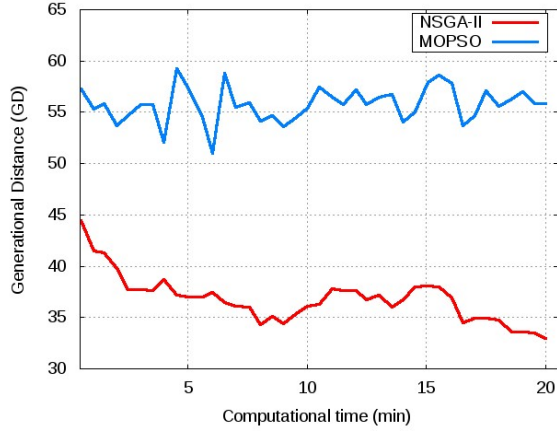| Scenario	MOEA	Number of platoons	Longest tour duration (min)	Energy (kWh)
U1	NSGA-II	12	**27.80**	**4.45**
	MOPSO	12	32.81	4.88
U2	NSGA-II	**20**	**48.74**	**15.00**
	MOPSO	21	51.44	15.93
U3	NSGA-II	**23**	**58.40**	**27.89**
	MOPSO	27	58.99	29.55
U4	NSGA-II	**29**	**65.30**	**39.11**
	MOPSO	32	69.13	44.95

Fig. 9. Generational Distance (GD): It measures how far the Pareto front (PF) is from the true Pareto front (PF*) obtained by gathering all the non-dominated solutions given by the two algorithms in 15 independent runs.

the ideal objective vector[1]. The table clearly shows that the NSGA-II algorithm gives better results regarding the number of platoons, platoon tour duration and the total consumed energy for all configurations under consideration. Figure 9 shows the generational distance [21] for the two algorithms as a function of computational time, where the number of parked vehicles is equal to 200. We can note from this figure that NSGA-II converges very quickly (the generational distance decreases rapidly with time) towards the true Pareto front within $20 \, \text{min}$ compared to the MOPSO algorithm that does not really converge. Therefore, based on these results, we can conclude that the convergence of NSGA-II is better than MOPSO.

VI. CONCLUSION

In this work, we have proposed an optimal routing algorithm for picking up automated vehicles based on NSGA-II. The experiments results presented in this paper show that the solution proposed is able to find near-optimal routes within reasonable computation time, even for a high number of parked vehicles. This routing algorithm can easily be adapted

---

[1]Ideal objective vector gives the best value in the Pareto front for each separate objective.

for a variety of other cases and to solve other constraints related to vehicle redistribution problem. For instance, as the urban traffic flow is often interrupted by the presence of signalized intersections, which can lead to the presence of vehicles outside platoons as well as platoon dispersion, the number of road junctions in the calculated routes should be minimized. Moreover, a parallel implementation of NSGA-II could make it more optimized to run under a shorter duration.

REFERENCES

[1] *World streets: the politics of transport in cities*, project information: https://worldstreets.wordpress.com/2010/12/20/.

[2] B. Zhou and K.M. Kockelman, *Opportunities for and Impacts of Carsharing: A Survey of the Austin, Texas Market*, International Journal of Sustainable Transportation, vol. 5, no. 3, pp. 135–152, 2011.

[3] M. Nourinejad and M. J. Roorda, *Carsharing operations policies: a comparison between one-way and two-way systems*, Transportation, vol. 42, no. 3, pp. 497–518, 2015.

[4] M. Marouf, E. Pollard and F. Nashashibi, *Automatic parallel parking and platooning to redistribute electric vehicles in a car-sharing application*, IEEE Intelligent Vehicles Symposium IV, Dearborn, Michigan, United States, Jun 2014, pp. 486–491.

[5] *Car2go*, www.car2go.com.

[6] *Autolib*, www.autolib.eu/en/.

[7] P. Daviet and M. Parent, *Platooning techniques for empty vehicle distribution in the Praxitéle project*, Pro 4th Mediterr Symp New Dir Control Automat, Maleme, Greece, 1996.

[8] M. Dror, D. Fortin and C. Roucairol, *Redistribution of self-service electric cars: A case of pickup and delivery*, Technical Report W.P. 3543, INRIA-Rocquencourt, France, 1998.

[9] N. Hafez, M. Parent and J. M. Proth, *Management of a pool of self-service cars*, IEEE Intelligent Transportation Systems ITSC, Oakland, USA, Aug 2001, pp. 943–948.

[10] A. G. Kek, R. L. Cheu, Q. Meng and C. H. Fung, *A decision support system for vehicle relocation operations in carsharing systems*, Transportation Research Part E: Logistics and Transportation Review, vol. 45, no. 1, pp. 149–158, Jan 2009.

[11] A. Febbraro, N. Sacco, M. Saeednia, *One-way carsharing: Solving the relocation problem*, Transportation Research Board 91st Annual Meeting, 2012.

[12] M. Bruglieri, A. Colorni and A. Lu, *The vehicle relocation problem for the one-way electric vehicle sharing, an application to the Milan Networks*, Social and Behavioral Sciences, vol. 64, no. 4, pp. 292–305, 2014.

[13] B. M. Baker and M. Ayechew, *A genetic algorithm for the vehicle routing problem*, Computers & Operations Research, vol. 30, no. 5, pp. 787–800, 2003.

[14] *OSRM: Open Source Routing Machine*, http://project-osrm.org/.

[15] A. Bolovinou, I. Bakas, A. Amditis, F. Mastrandrea and W. Vinciotti, *Online prediction of an electric vehicle remaining range based on regression analysis*, IEEE International Electric Vehicle Conference (IEVC), Florence, Italy, Dec 2014, pp. 1–8.

[16] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, *A fast and elitist multi-objective genetic algorithm: NSGA-II*, Evolutionary Computation, IEEE Transactions on, vol. 6, no. 2, pp. 182–197, 2002.

[17] H. Ali, W. Shahzad and F. Khan, *Energy-efficient clustering in mobile adhoc networks using multi-objective particle swarm optimization*, Applied Soft Computing, vol. 12, pp. 1913–1928, 2012.

[18] M. López-Ibáñez, and T. Stutzle, *Automatically improving the anytime behaviour of optimisation algorithms*, European Journal of Operational Research, vol. 235, no. 3, pp. 569–582, 2014.

[19] J. R. Schott, *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.

[20] S. R. Ranjithan, S. K. Chetan, and H. K. Dakshima, *Constraint Method-Based Evolutionary Algorithm (CMEA) for Multi-objective Optimization*, EMO, LNCS, No. 1993, pp. 299–313, 2001.

[21] D. A. V. Veldhuizen and G. B. Lamont, *Multiobjective Evolutionary Algorithm Research: A History and Analysis*, Technical Report TR-98-03, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Ohio, 1998.