

# A GRASP algorithm for the vehicle-reservation assignment problem

Beatriz Brito Oliveira<sup>1</sup>, Maria Antónia Carravilla<sup>1</sup>, and José Fernando Oliveira<sup>1</sup>

**Abstract** Car rental companies face a critical problem related to the empty repositions (or “deadheading trips”) of their vehicles, which comprise a significant and unnecessary financial and environmental impact. Especially when dealing with special types of vehicles whose number of units is small, the company is forced to empty reposition them between rental stations in order to meet reservations requirements concerning available time and location. This paper proposes a GRASP algorithm to generate a global vehicle schedule that maximizes the company’s profit whilst reducing the costs of empty transfers. Using real instances, the value of this approach is established.

**Key words:** vehicle-reservation assignment; empty transfers; GRASP.

## 1 Introduction

The car rental business is becoming heavily dependent on operational efficiency. As holding costs of assets have been growing faster than the price level, it is important to assure optimal utilization of resources as well as a high service level (?). The work described in this paper proceeds from a project funded by a Portuguese car rental company whose main objective was to redesign and enhance the company’s procedures to assign special types of vehicles to reservations. Due to their unique characteristics, these vehicles are not highly required. Consequently, the number of cars available of each of these special groups is small, thus forcing the company to transfer them empty between rental stations in order to meet the reservations requirements. The company felt the need to improve the assignment process so as to minimize these empty transfers. This paper presents a metaheuristic for this problem. The main objective is to maximize the total profit of the company and assure customer satisfaction by fulfilling as many reservations as possible, whilst reducing the cost of the empty transfers. In fact, the reduction of the empty repositions takes a significant role both in increasing operation efficiency and thus profitability of the business model and in improving the environmental sustainability of the company.

The empty repositions or “deadheading” trips are a studied issue within the transportation industry framework. ? recognize the impact empty flows have on logistic systems, namely on their operational and economic performance, as they generate costs and no revenue.

The car rental logistics short-term problem has not been frequently addressed on the transportation logistics literature. Nevertheless, some important contributions can be found. ? structure the fleet planning process of a car rental company in three sequential phases: pool segmentation, strategic fleet planning, and tactical fleet planning. The first phase consists on clustering the rental locations of the car rental company in geographically and demand-correlated pools; the different rental stations within a pool share the same fleet, whose number of vehicles is determined in the second phase. The third phase consists on determining the number of vehicles that should be available at each station, in each period of time. The problem of empty transfers is herein considered. ? model these three phases and propose solution methodologies considering the hierarchical structure of the decision-making process. None of these works addresses the specific assignment of reservations to the available vehicles. Nevertheless, this could be considered a lower-level (more operational) sub-problem of the tactical fleet planning. To the

---

<sup>1</sup>INESC TEC, Faculty of Engineering, University of Porto, 4200-465, Porto, Portugal.  
e-mail: [beatrizbritooliveira@gmail.com](mailto:beatrizbritooliveira@gmail.com), [mac@fe.up.pt](mailto:mac@fe.up.pt), [jfo@fe.up.pt](mailto:jfo@fe.up.pt)

best of our knowledge, within the car rental logistics optimization framework, the lower-level vehicle-reservations assignment problem has not yet been approached.

The remainder of this paper is organized as follows: the next section briefly describes the problem tackled. The following section presents the solution methods and, finally, the main results, based on real instances, are presented and conclusions are drawn.

## 2 Problem description

The problem described in this paper aims to allocate a certain set of reservations to the available special vehicles.

Each reservation has the following characteristics: the date and station in which the customer wants to pick up the vehicle, the date and stations in which the customer wants to deliver it, and the revenue of the reservation. The vehicles are characterized by their current occupation; as they are currently fulfilling a certain reservation, each vehicle will be available when and where that reservation in progress ends. Other parameters of this problem are the costs and the time of the empty transfers from one station to another (Figure ??). The objective is to maximize the revenue of the assigned reservations deprived of the costs of the empty transfers. The main restrictions of the problem are related to the availability of the vehicles on the moment and location considered.

Since the customers have specific requirements, the company should provide them with exactly what was requested. Yet when that is not possible, it is a common practice in this sector to offer the customer a vehicle from a better group for the same value (upgrade). When that option is not available, the company offers the customer the possibility to rent a vehicle from a worst group with a price discount (downgrade). The possibility to upgrade or downgrade the reservations is beneficial for the company as far as service level is concerned, although it may lead to the fulfilment of reservations for a minor profit.

The inclusion of this issue in the solving method increases the dimension of the analysis (as more vehicles are available for the same number of reservations), introducing additional restrictions as for which groups can be upgraded or downgraded to which groups. Hence, reservations need to be characterized by the group (type of vehicle) required by the customer and vehicles should be associated with a specific group. As for the cost of these decisions, in fact, the company does not incur on any additional cost by allocating a better or worse vehicle than requested. Since the vehicles are available, it is better to seize the possible profit of the reservation than not fulfilling it and thus dissatisfying a customer. Nevertheless, it is commonly a company's policy to avoid upgrading a reservation when not necessary and to execute downgrades only if no other option is possible. Both adjustments require the specific authorization of the customer; nevertheless, the upgrade is virtually always accepted.

It is also important to know the status of the reservation, as far as confirmation to the customer is concerned. As to control customer satisfaction, the company is interested in preferring the allocation of confirmed reservations over non-confirmed ones.

## 3 Problem solution

The main output expected from this tool is a detailed vehicle global schedule with the reservation assignment plans for each car, as it is shown in Figure ?? for three example vehicles.

The company felt it was important to obtain good allocation plans. Nevertheless, it was known that these plans, good for the time being, would probably cease to be so due to the dynamic characteristics of the problem. Heuristics were hence drawn to solve the problem due to the need of quickly obtaining a good solution.

The solution to this problem was also based on the need to recognize and model the two functioning modes of the decision support system. On the one hand, the company needs to be able to rapidly establish whether a certain reservation may or not be allocated, considering the current vehicle schedule, in order to promptly answer the client on the phone, confirming or not such reservation – the *online mode*. On the other hand, a more robust method is needed to improve the vehicle schedule built during the day – the *batch mode*. The latter must consider that every confirmed reservation should be necessarily allocated to some vehicle.

A GRASP algorithm (section ??) was drawn to fulfil the *batch mode*. Due to the inherent dynamic nature of this problem, it is difficult for the company to register the actual global vehicle schedule for each instance. Consequently, in order to enable the quantification of the improvement brought by the

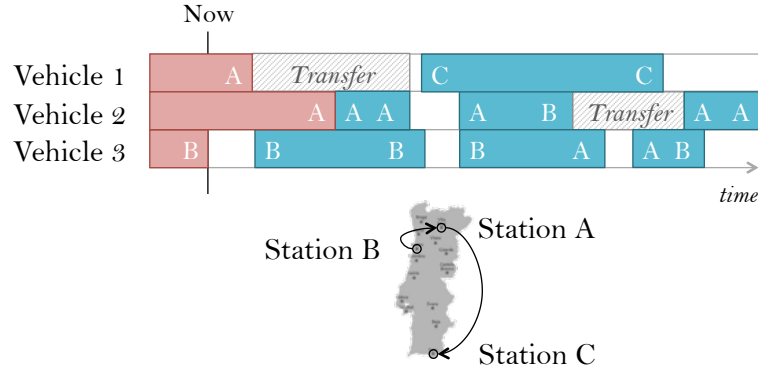


Fig. 1: A vehicle global schedule consists on the scheduling of the reservations fulfilled by each vehicle, which may lead to empty transfer movements.

developed support system, a heuristic that mimicked the *online* decision process of the employees in charge of this process (section ??) was also developed.

### 3.1 Mimicry heuristic

Considering the functioning modes, it is possible to classify the existing procedure in the company as an *online-only* mode. In fact, the employees attempt to allocate each reservation individually, not being able to enhance the global vehicle schedule as a batch. Thus, it is possible to determine that the main inputs for this decision are the single reservation to be allocated and the global vehicle schedule.

The decision process currently followed also depends on the perception that, when there are numerous possible vehicles in which to allocate a certain reservation, the employee may not be able to visually apprehend the occupation of all of them, being limited by the size of the computer screen. Therefore, another important input is the number of vehicles the employee can visualize at the same time and the natural human restrictions when dealing with complex combinatorial problems.

During the decision process, the vehicles are presented in alphabetic order of their license plate. The employee visualizes each vehicle occupation, knowing the starting and returning stations and dates of each reservation allocated to it. The first attempt is to allocate the reservation as the last assignment of one of the visible vehicles, in the best possible position. As there is no information regarding the reservation profit, there is an attempt to reduce the costs by avoiding empty transfers and using as first criterion the coincidence between the returning station of the last reservation and the starting station of the reservation to allocate. The second criterion used is the idle time of the vehicle caused by the hypothetical allocation. The third criterion is to allocate the reservation in the first possible fit between already allocated reservations.

When this procedure is not able to find a feasible solution, a simple switch move within the allocated reservations is tried. For each vehicle, it is verified whether the reservation to be allocated overlaps only one other reservation. If this is the case, it may be possible to allocate the overlapped reservation to another vehicle and insert the unassigned reservation in its previous place. If the feasibility conditions are fulfilled, the simple switch is completed in the first possible fit.

All the previously discussed procedures relate to the visible vehicles on screen. The last alternative procedure is the allocation as the last assignment of a vehicle outside the visible screen; as employees scroll down, if a possible fit is found the reservation is allocated.

### 3.2 GRASP

The methodology selected to solve this problem, as the *batch* mode is concerned, was GRASP (Greedy Randomized Adaptive Search Procedures), first introduced by ?. GRASP is an iterative technique with two sequential phases: the construction of a solution based on a randomized greedy heuristic, and the local search, which applies small adjustments to the solution provided by the first phase with the goal of achieving improvement. Each GRASP iteration comprises these two phases and originates a feasible

solution; throughout the iterations, the best solution found is preserved. This methodology was chosen due to its intuitive structure and relatively simple implementation.

The constructive heuristic, which aims to assign reservations to vehicles, is based on the ranking of both reservations and vehicles. The reservation rank is based primarily on its confirmation status followed by the proximity of the starting date of the reservation and finally, by decreasing profit. Subsequently, for each reservation, starting with the one ranking higher, the vehicles which could be available on the required date and location are ranked from the lowest to the highest transfer cost between the returning station of their current reservation and the starting station of the reservation considered. If there is a tie, the vehicles are ranked in ascending order of idle time – the time span between being available in the required location and the start of the reservation.

Two different approaches were designed and tested for the local search routine. The move that defines the neighbourhood structure is based on the swap of pairs of allocated reservations. The selection of the new incumbent solution is different: in the first case the new incumbent solution is the best found in the whole neighbourhood, a best-improvement approach, whilst the second may be described as a first-improvement approach.

The best-improvement approach generates an all-encompassing neighbourhood structure based on a *LPS* – *List of Possible Swaps* that stores all the possible swaps within the incumbent solution. Each neighbour embodies the incumbent solution modified by one specific swap of the *LPS*. In order to choose the best possible improvement, all neighbours are evaluated. If some improvement in the objective function is possible, the best neighbour becomes the incumbent solution and a new *LPS* is constructed based on it. A new iteration is run and the algorithm stops when no neighbour is able to improve the objective function.

In the first-improvement approach, the local search is also initialized with the selection and listing of the possible swapping pairs within the initial solution (the *LPS*). Each listed pair originates a neighbour – the incumbent solution modified by the swap. The neighbours are only explored until one is found that improves the objective function. In fact, the listed pairs are swapped within each best neighbour that is found and when this happens a new neighbourhood structure is generated. Nevertheless, unlike the previous approach, the algorithm continues to try to swap the pairs listed on the first *LPS* but now within this solution that is now the neighbourhood center. Note that since the listed swaps were selected within a different solution a new feasibility check must be run. Once again, the first neighbour that is able to improve the objective function is selected as the base solution for the neighbourhood generation. The procedure described is repeated until all swaps in *LPS* have been attempted. When an *LPS* has been completely explored, and while it is possible to achieve an improvement, a new *LPS* is generated from the incumbent solution and the process is repeated. It is important to understand that this approach was developed with the objective of obtaining a good, swift routine, which explored the neighbourhood structures in depth rather than in width.

## 4 Computational tests and results

The data used to test this approach was retrieved from the company's database in July, before the beginning of the the highest season in the car rental business in Portugal. Therefore, the instances reflect the busiest and most demanding time period faced by the company as far as tactical planning is concerned.

Three instances were selected, each concerning a different vehicle group within the special vehicles fleet and could be classified comparatively as easy (A), average (B), and difficult (C), for both upgrade/downgrade-allowing and -not-allowing situations. These instances were solved using the three routines: the mimicry heuristic was used as a means to simulate the results currently obtained by the company; then, the GRASP algorithm was run using the first-improvement local search as well as the best-improvement one.

Considering the two variations of each of the three instances (allowing for upgrade/downgrade vehicles or not) and the two approaches to the swap local search, twelve different GRASP variants were tested. The algorithms described were developed in a VBA platform, using Microsoft Office Excel as the input/output interface. Each GRASP variant was run for 15 iterations using a standard personal computer with an INTEL i7 2.70 GHz CPU and 8 GB installed memory.

The general results may be found in Table ??, as far as the improvement between the results of the mimicry heuristic and the GRASP algorithm is concerned.

It was possible to verify that the metaheuristic approach lead to better results when the difficulty of the instance increased. In fact, for the easy instance (A), the increase on the profit of the company when

compared to the values obtained by the mimicry heuristic was virtually non-existent, both considering and not considering upgrades. As for the average instance (B), when considering that no upgrades or downgrades to other groups were possible, both local search routines were able to increase the company's profit in 10,7%. When considering upgrading and downgrading vehicles, there was a 5,5% increase. The difficult instance (C), solved by both local search routines, whilst not considering upgrades, lead to an increase of over 8,5% of the profit. When considering these auxiliary vehicles, both routines were able to increase the results of the company by 12,1%. These increasing values of improvement were expected, as the current procedure used by the company, although extremely refined by the experience and knowledge of the operators, meets the limits of the human ability to apprehend large amounts of data and thus tackle big combinatorial problems.

For every instance and upgrading situation, the swift first-improvement (FI) local search solved the problem faster; in fact, the time was perceived to be proportional to the amount of vehicles to assign, increasing when considering upgrades and when solving instances with more vehicles available. For every case, nevertheless, the algorithms were run in an acceptable time, considering that this *batch* approach to the problem is designed to be run during the night. For most cases, the swift local search was able to match the results of the best-improvement routine. Nevertheless, for instance C, considering upgrades, a worst result was obtained using the first-improvement heuristic than the exhaustive best-improvement approach; nonetheless, this difference represented only 0,02% of the profit. For the cases considered, the contribution of the local search to the overall improvement was between 0,5% and 2%, decreasing with the difficulty of the instance.

It should also be noticed that for the average instance (B) and for the upgrade-allowing difficult instance (C), this approach was also able to allocate new reservations that the mimicry heuristic was not able to insert in the global vehicle schedule.

Table 1: Results for the different GRASP variants - percentage of profit improvement and increase in the number of reservations fulfilled when compared to the company's procedures, and global run time.

		As for GRASP iterations				Global	
		average	std dev	worst	best	time (min)	new reserv
A	No Up BI	0,1%	0,2%	-0,2%	0,5%	6	0
A	No Up FI	0,0%	0,2%	-0,3%	0,5%	1	0
A	Up BI	-0,1%	0,2%	-0,4%	0,3%	59	0
A	Up FI	-0,1%	0,2%	-0,5%	0,3%	7	0
B	No Up BI	10,5%	0,4%	10,0%	10,7%	2	23
B	No Up FI	10,5%	0,4%	10,0%	10,7%	1	23
B	Up BI	5,4%	0,1%	5,3%	5,5%	91	23
B	Up FI	5,4%	0,1%	5,3%	5,5%	19	23
C	No Up BI	8,4%	0,1%	8,3%	8,5%	2	0
C	No Up FI	8,4%	0,1%	8,3%	8,5%	1	0
C	Up BI	12,0%	0,1%	11,8%	12,1%	149	12
C	Up FI	11,9%	0,1%	11,8%	12,1%	47	12

Note: Up - Allowing upgrades/downgrades;

No Up - Not allowing upgrades/downgrades;

BI - Best-improvement; FI - First-improvement.

## 5 Conclusions

This paper reported the problem faced by a car rental company in managing the tactical planning of its special vehicles fleet, as far as allocation of reservations to available vehicles is concerned. The main objective was to maximize the total profit of the company, by means of reducing empty vehicle repositioning transfers between rental stations. The ultimate goal was thus to provide the company with a functioning tool that was able to improve the global vehicle schedule considering the required rental groups that belonged to the *special vehicle* fleet.

In the system developed, the user is able to select the rental groups to improve, allow upgrades and/or downgrades, and select the local search approach, considering the time available to run it. The outputs of

each run consist on a detailed schedule for the vehicles of each group and a chronogram with the vehicle occupation.

For the most difficult instances and groups, the company is able to increase its profit up to 12%. It is significant to remark that this value, referent to a volume of reservations that historically comprehends most of the volume of the high season demand, represents a significant financial impact for the company. Moreover, the company may also be able to fulfil more reservations, increasing customer satisfaction and market share. Another major advantage is the re-allocation of two qualified and experienced employees to other value-adding tasks, namely within the strategic rather than tactical planning level. In fact, this software may also provide the company with insights related to the strategic fleet sizing problem, as it can be used as a simulation tool for the sale and purchase of vehicles.

Nevertheless, it is still possible to improve the approach to this problem. One of the main characteristics of this problem is the extreme inflexibility of the starting and finishing times of the reservations. If there are many reservations concentrated in a specific time period, the problem becomes even more rigid and the solutions more difficult to improve by this method, since small adjustments made to a specific solution lead often to infeasible results. In fact, for such cases, it is the randomness applied to the solving method that leads to the major improvement. Therefore, a new approach could be attempted, namely solving this combinatorial problem by an optimization method.

As future work, it would be also interesting to tackle the issue raised by ?, related to the measuring of the environmental impacts; the authors state that travel time is a better estimate to degree of pollution caused, as this can be reduced by travelling for shorter times (and at better speeds). Therefore, it could be interesting to reformulate the problem as to minimize the empty transfer times, rather than the costs.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments that helped to improve the paper.

This work was partially funded under project “Smart Manufacturing and Logistics - NORTE-07-0124-FEDER-000057” by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

## References

- Dejax, P. J. and Crainic, T. G. (1987), ‘A review of empty flows and fleet management models in freight transportation’, *Transportation Science* **21**(4), 227–247.
- Feo, T. A. and Resende, M. G. C. (1995), ‘Greedy Randomized Adaptive Search Procedures’, *Journal of Global Optimization* **6**, 109–133.
- Fink, A. and Reinert, T. (2006), ‘Modeling and solving the short-term car rental logistics problem’, *Transportation Research Part E: Logistics and Transportation Review* **42**(4), 272–292.
- Pachon, J. E., Iakovou, E., Ip, C. and Aboudi, R. (2003), ‘A synthesis of tactical fleet planning models for the car rental industry’, *IEEE Transactions* **35**(9), 907–916.
- Pachon, J., Iakovou, E. and Ip, C. (2006), ‘Vehicle fleet planning in the car rental industry’, *Journal of Revenue and Pricing Management* **5**(3), 221–236.
- Sbihi, A. and Eglese, R. W. (2007), ‘Combinatorial optimization and Green Logistics’, *4OR-Q J Oper Res* **5**(2), 99–116.