

Fleet Management: a Vehicle and Driver Assignment Model

Filipe Monnerat^{1,2}, Joana Dias^{1,3}, Maria João Alves^{1,3}

¹ CeBER and Faculty of Economics, University of Coimbra, Av. Dias da Silva 165, 3000 Coimbra, Portugal

² Universidade Federal de Viçosa, Brasil

³ INESC Coimbra, DEEC, University of Coimbra, Polo 2, 3030-290 Coimbra, Portugal

filipemonnerat@gmail.com; joana@fe.uc.pt; mjalves@fe.uc.pt

Abstract

Many institutions have their own vehicle fleet and hired drivers that can be assigned to working trips to ensure that they all take place within the desired time periods. In these situations, it is important that vehicles and drivers are assigned to planned trips in an optimal and automated way. Trips are planned to occur over a given planning horizon and have a number of characteristics that have to be taken into account: a trip is characterized by a starting time, start and end locations, travelling distance and time, number of drivers needed for each assigned vehicle, number of passengers, and so on. This paper considers the problem of assigning both vehicles and drivers to a set of planned trips, having as objective the minimization of total cost. To properly address this problem it is necessary to consider a number of features, like the possibility of vehicles being shared between different trips if the timeframes and destinations are compatible. The mathematical model developed and presented in this paper takes all these features into account. The problem was motivated by a case study in the context of a public Brazilian university. This case study is also described and computational experiments using a general solver are reported. The general solver is capable of calculating optimal solutions in reasonable computational times, but problems where the fleet is mainly composed of small vehicles are more challenging. A *mathuristic* has been developed and tested, that seems to be competitive for these type of problems.

Keywords: Combinatorial optimization, Generalized Assignment Problem, Fleet Management

This is a pre-copyedited version of article published in European Journal Operational Research. The final version of this article is available online at <https://doi.org/10.1016/j.ejor.2019.03.021>

© 2019. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

1. Introduction

In this paper a fleet management problem is considered, assuming that the vehicles and drivers have to be assigned to a set of planned trips over a given time horizon. Let us consider that a given institution has to ensure that a series of predetermined trips take place within predetermined time periods. The

institution has a set of available vehicles with different passenger capacity and operating costs (related basically to the distance travelled). The institution also employs a number of drivers. It wants to determine the best assignment of vehicles and drivers to the planned trips, with the objective of minimizing the total cost. The trip costs to consider are the vehicle operating and driver assignment costs. These latter costs are directly related to the trips made, other than the basic salary of drivers. Examples are the situations where drivers are paid allowances because they are working more than a threshold limit of consecutive hours, or because they are working during the weekend, for instance.

A trip is characterized by its origin and destination, and its departure and returning times. The number of passengers associated with each trip is also known. It is possible for different trips to share resources as long as they are compatible regarding both space (starting and ending location) and time. Trips can have different needs regarding the number of drivers per vehicle. A long distance trip, for instance, may require more than one driver per vehicle to comply with safety regulations.

There is a vast literature related to fleet management problems. Fleet management is a broad concept that incorporates decisions about fleet sizing and configuration, fleet allocation, vehicle routing, considering homogeneous or heterogeneous vehicles. Both deterministic and stochastic problems can be considered. (Baldacci, Battarra, & Vigo, 2008) present an overview of approaches for fleet management in the context of heterogeneous vehicle routing problems. (Hoff, Andersson, Christiansen, Hasle, & Lokketangen, 2010) review the literature which focuses on fleet composition and routing problems for the maritime and road-based transport of goods. The authors state that unless routing and trip scheduling are already predetermined, it is crucial to consider routing and fleet composition decisions simultaneously. (Bielli, Bielli, & Rossi, 2011) identify the most relevant problems in fleet management for different transport modes. The authors refer to vehicle routing and scheduling problems, dynamic fleet management interpreted as a real time vehicle routing problem, private and public urban transport, dial-a-ride transport, and specific problems related to air, maritime, rail and intermodal transport. (Ertogral, Akbalik, & González, 2017) present a literature review summarizing the state-of-the-art regarding the fleet sizing problem. Most of the cited papers are based on models for vehicle routing problems.

A driver-task assignment in the context of container movement with time-windows and uncertainty in task duration is described in (Cheung & Hang, 2003). It is assumed that assignment decisions can change in the course of time if new information arrives. A similar problem has also been tackled in a subsequent work (Cheung, Hang, & Shi, 2005), and this was solved by an adaptive labelling approach. The fleet management problem in the container trucking industry is addressed by (Coslovich, Pesenti, & Ukovich, 2006), where the objective was to minimize total cost (routing, resource assignment and repositioning costs) and satisfy time window constraints. Each driver is modelled as a driver and vehicle combination. The authors solve the problem by breaking it down into three simpler subproblems.

More recently, a fleet management system with a heterogeneous fleet deployed for waste collection is described in (Markov, Varone, & Bierlaire, 2016). Each route can start and end in different depots.

Real time dynamic fleet management interpreted as a simultaneous task allocation and vehicle deployment problem is addressed by (Billhardt et al., 2014). The authors describe a discrete event-based framework capable of dealing with the arrival of new tasks or unforeseen changes in the current fleet situation. This framework assumes that vehicles are capable of sending information about their current position and state to a fleet tracker system. The authors assume that the driver-vehicle assignment is known. A heuristic based on multiple neighbourhood search is proposed.

The importance of simultaneously optimizing vehicle and human resources assignment has been recognized by several authors. (Freling, Huisman, & Wagelmans, 2003) present different mathematical formulations for the vehicle and crew integrated scheduling problem when there is a single depot, and develop algorithms based on column generation in combination with Lagrangean relaxation approaches. The authors conclude that the integration is most useful when changing the vehicle or driver is not allowed, and crew costs are significant compared with vehicle costs. (Laurent & Hao, 2007) also consider the simultaneous vehicle and driver scheduling problem in the context of a limousine rental company. One differentiating feature of their work is the explicit consideration of the drivers' skills. A driver can only be assigned to a trip if his skills match those required for that trip (like being able to speak a given language). The authors develop a two stage algorithm, based on constraint programming and simulated annealing. Drivers' skills are also taken into account in (Peters, Matta, & Boe, 2007). The authors consider a single depot with bus fleets of three different categories and with a set of drivers with different skills, which determine the driver's primary and secondary jobs (the bus categories the driver can be assigned to, and a preference associated with this assignment). The problem is solved by a combination of Lagrangean relaxation and tabu search.

A multi-period vehicle and crew scheduling problem considering the possibility of outsourcing for both vehicles and drivers is studied by (Zäpfel & Bögl, 2008). The model is applied in the context of defining routes for postal companies, where time windows and vehicle capacities have been considered. The authors present an algorithm framework that can be adapted to different metaheuristics.

(Simão et al., 2009) address the truckload trucking problem, where a set of drivers have to be assigned to a set of loads that need to be transported, where the transportation time is usually long. If the transport distance is long it can sometimes be necessary to assign more than one driver to one load, so that drivers can take rest periods without delaying the transportation. The study is based on a large scale real case problem involving more than 6000 drivers. The authors tackled the problem using an algorithmic approach based on approximate dynamic programming, merged with mathematical programming and machine learning techniques.

Another fleet management problem can be found in car rental logistics problems (Fink & Reiners, 2006; Oliveira, Carravilla, Oliveira, & Toledo, 2014). Decisions that have to be made in these cases are related not only to fleet composition, but also to the assignment of vehicles to clients' requests and repositioning cars at different rental stations.

Vehicle and client compatibility constraints can be found in the model developed by (Detti, Papalini, & Lara, 2017), considering a healthcare application. The authors explicitly consider the condition of the patient to make sure that the assigned vehicle is compatible with that condition (patients that need a wheelchair, for instance). Patients' preferences are also explicitly taken into account. The problem is then solved by a heuristic approach based on tabu search and variable neighbourhood search.

There are also vehicle routing problems with explicit driver assignment concerns related to customer service quality assurance (for a survey, the reader is referred to (Kovacs, Golden, Hartl, & Parragh, 2014)). It is possible to consider arrival time consistency, person-oriented consistency and delivery consistency (Kovacs et al., 2014). One example that considers both time arrival and person-oriented consistency can be found in (Groër, Golden, & Wasil, 2009). In that work the authors take into account the relationships established between the drivers and the clients. Each client must therefore be visited by the same driver whenever the client requests the service, and at approximately the same time. The authors develop a mathematical programming model and an algorithm and test them on a real-world problem involving 3715 customer locations. (Coelho, Cordeau, & Laporte, 2012) address both person-oriented and delivery consistency in the context of vendor-managed inventory systems. The authors develop a *matheuristic* based on an adaptive large neighbourhood search. Driver consistency in dial-a-ride problems is addressed by (Braekers & Kovacs, 2016). The authors examine the possibility of several users sharing the same vehicle, despite having different pick-up and drop-off locations. The maximum number of different drivers the clients will meet over a given planning horizon is upper bounded. The authors present two formulations and apply two algorithmic approaches, namely, branch-and-cut and large neighbourhood search. Driver consistency in vehicle routing problems with unknown demands is described in (Spliet & Dekker, 2016). The stochastic demand is represented by means of scenarios. Driver consistency is not client focused but driver focused, in the sense that the constraints guarantee that each driver will visit at least a lower bound of the customers that are assigned to him. The solution procedure solves the assignment and routing problems sequentially.

Another field of research that can be seen as related to fleet management problems is the school bus routing problem (the reader is referred to (Park & Kim, 2010) for a review). (Li & Fu, 2002) consider a case study in Hong Kong for which a multiobjective programming problem is formulated. The objectives have to do with the minimization of costs, but also the minimization of travel times both for students and buses. Drivers' assignments are not explicitly considered. A school bus routing problem which explicitly holds that students from different schools can share the same bus (mixed load) is described in (Park, Tae, & Kim, 2012). The authors develop a constructive heuristic approach that uses as initial solution a schedule that does not consider the possibility of a mixed load and then improves it by considering the possibility of mixed load. A similar problem is described in (Lima, Pereira, Conceição, & Nunes, 2016), where the authors assume the existence of a heterogeneous fleet and the same starting time for all schools. The authors compare five metaheuristics, as well as integrating four local search neighbourhoods especially adapted to deal with mixed loads. They reach the conclusion

that iterated local search integrating a random variable neighbourhood descent local search outperforms the other approaches. School bus routing in urban contexts motivates the development of a model that simultaneously optimizes routes and the choice of bus stops students are assigned to (Riera-Ledesma & Salazar-González, 2012). The authors call this problem the multiple vehicle traveling purchaser problem. A set of valid inequalities that strengthen the linear programming relaxation are presented and these are used in the proposed branch-and-cut algorithm.

The point of view in this paper is different from other works. It is assumed that the set of available vehicles is known and determined a priori, and this fleet is heterogeneous. A number of trips have to be carried out and the scheduling and routing decisions have already been made. The trips are work trips made by members of an institution. So, the scheduling of the trip is defined by the travellers and each trip is a route that starts and ends in the institution, with predefined stops (usually only one destination). The main concern of the institution is to optimize the assignment of its fleet to the planned trips over a given planning horizon so that costs are kept as low as possible. The institution has to consider both the assignment of vehicles to trips and the assignment of drivers to these vehicles. This paper is thus concerned with a resource assignment problem, where two different resources have to be assigned to a given task. These resources cannot be assigned independently of each other because the number of vehicles assigned and the characteristics of the trip will determine the number of drivers that are needed. The problem considered in this paper can thus be regarded as an extension of the generalized assignment problem (GAP). The GAP can be defined as finding the optimal assignment of n jobs (items) to m agents, where both jobs and agents have a limited capacity and one agent can be assigned to more than one task (as long as its maximum capacity is not exceeded). The GAP is known to be NP-hard and the corresponding feasibility problem NP-complete (Öncan, 2007). There are several extensions and variants of GAP (for a review see, for instance, (Morales & Romeijn, 2004; Öncan, 2007; Pentico, 2007)).

(Kogan, Khmelnitsky, & Ibaraki, 2005) describe a dynamic GAP, where an agent can be assigned to many tasks and a task can also be assigned to many agents; tasks are interpreted as being composed of a set of units. Demands for these task units are stochastic. The authors derive several mathematical properties of the model and present a needle time decomposition algorithm. (Moccia, Cordeau, Monaco, & Sammarra, 2009) present three linear integer formulations for the dynamic GAP and develop a column generation algorithm capable of calculating lower bounds for the objective function, embedded in a heuristic procedure.

A dynamic version of GAP is described by (Mazzola & Neebe, 2012). The assignment of tasks to agents has to be guaranteed in each time period of a given planning horizon, assuming that there are transition costs whenever a task is reassigned to a different agent in two consecutive time periods. The authors propose two model formulations and two algorithmic approaches, which are a Lagrangean relaxation based heuristic and a branch-and-bound algorithm. More recently, (Xu, Wang, & Sun, 2018) consider a GAP with cooperative agents in a scalable multi-agent system (distributed GAP). The authors assume

the existence of a large number of distributed heterogeneous agents that should be optimally assigned to tasks aiming to fulfil a common goal. They propose an intelligent routing decentralized heuristic to solve the problem, based on agent modelling. Agents use local information inferred from previous searches to decide which agents are more capable of contributing to the improvement of the current solution.

The multi-resource GAP is a GAP variant where each agent has to use not just one but a variety of resources to perform the tasks that are assigned to him (Gavish & Pirkul, 1991). This problem is extended in (Shtub & Kogan, 1998) by including the possibility of demand variation over time and dynamic capacity assignments. (Karsu & Azizoglu, 2014) consider a bicriteria version of the problem that simultaneously minimizes the total workload assigned over all agents and minimizes the maximum workload assigned among agents. A set of nondominated solutions is then generated.

Several authors have proposed heuristic and metaheuristic procedures for the GAP, the dynamic GAP and the multi-resource GAP. The multi-resource GAP is tackled in (Mazzola & Wilcox, 2001) by the use of a constructive heuristic structured in three phases that is able to find an admissible solution. This solution is then submitted to an additional improvement procedure. They also test a hybrid procedure combining two different heuristics, including a sub-gradient optimization procedure. A heuristic based on very-large scale neighbourhood search including an adaptive mechanism for adjusting search parameters is presented in (Yagiura, Iwasaki, Ibaraki, & Glover, 2004). A few years later, (Mitrović-Minić & Punnen, 2009) use both very large-scale neighbourhood search and variable neighbourhood search applied to the same problem. The authors were able to find the best solution known at that time for some benchmark instances. (Kiraz & Topcuoglu, 2010) apply hyper-heuristic methods to the dynamic GAP. The low-level heuristics are based on local search procedures based on mutation operators. A tabu search embedding a branch-and-bound procedure is proposed by (Woodcock & Wilson, 2010) for GAP. A parallel genetic algorithm developed for GAP is described in (Liu & Wang, 2015), taking advantage of the inherently parallel computation possibilities of genetic algorithms. Different migration strategies are tried. Algorithms based on differential evolution and embedding three distinct local search techniques are presented in (Sethanan & Pitakaso, 2016). The performance of the described algorithms is compared with a bee-based algorithm and tabu search. (Souza, Santos, & Coelho, 2017) describe a hybrid method based on scatter search and using a tabu list for generating a pool of solutions. The authors were able to improve the best solutions known thus far for 21 GAP benchmark problems and prove the optimality of 15 other solutions.

The model we have developed was motivated by a real case of assigning resources to trips in a public Brazilian University. It is an extension of the GAP but differs from the multi-resource GAP in that in our case a task (trip) needs two separate resources (vehicles and drivers), and the requirements of one of the resources (drivers) will depend on the number of assigned units of the other resource (vehicles). Moreover, tasks do not take place all at the same time, so the time periods in which they do occur have to be taken into account when considering the resources' capacity constraints and sharing possibilities.

Solutions are calculated using a general mixed-integer linear programming solver (*cplex*). A *matheuristic* was also developed, which combines a metaheuristic (a genetic algorithm) with a mathematical programming solver (*cplex*) to solve sub-problems.

In the next section, the real case that motivated the development of the model is described. Section 3 describes the mathematical model. Section 4 illustrates the reasoning behind the model by examining a small size version of the problem. Section 5 presents the results obtained when the model is applied to the real case (instance A) and to the scaled down problem based on the real case (instance B). Section 6 presents the computational results of several instances randomly generated taking instance B as their basis. These instances aim to study the impact of the problem data on computational time and total cost. Changes in the number of vehicles and the number of drivers available, in the fleet composition and in the costs associated with the trips are analysed and discussed. Section 7 describes the *matheuristic* developed and presents and discusses some computational results. Section 8 presents some conclusions and identifies paths for future research.

2. The problem of optimizing work trips in a public Brazilian University

The problem addressed in this paper was motivated by the real need of a public Brazilian University to optimize work trips required by its employees with the aim of saving costs.

The University has its own resources, such as vehicles, drivers, mechanics, auxiliary and car wash personnel, as well as technical-administrative staff. The costs associated with the trips may vary significantly. They depend not only on the requirements (origin/destination, number of passengers, number of drivers needed per vehicle) but also on the resources assigned to each trip (one large capacity vehicle vs. several vehicles, sharing resources with other compatible trips). It is possible that not all drivers are available all the time, since they can have different rest periods and may already be assigned to other trips (work trips that take place on a regular basis, which are not requested by workers). Such unavailability can be reflected in the assignment costs.

Currently, the users (employees, teachers or students) submit requisitions for work trips in a decentralized way. These requisitions are then sent to a centralized department which tries to identify compatible demands based on the routes, destinations, date and time of departure and return. Two or more requisitions are compatible if their origins and destinations match (which means that trips have the same origin/destination or that one trip is a sub-route of another one), and their schedules can also be reconciled, thus enabling resources to be shared. At present the University does not use any optimizing procedure to manage the assignment of resources to planned trips, although some cost reduction is achieved by ad hoc adjustments that allow some compatible requisitions to be grouped together.

The problem consists of assigning vehicles and drivers to trip requisitions. Some trips may require more than one vehicle due to the number of passengers, and more than one driver per vehicle (for long trips). Compatible trips may share vehicles and drivers. In addition, the same vehicle and driver can be used for trips occurring in different periods of time.

To be able to incorporate all the real features of the problem, a very flexible model has been developed. Although in the real problem the time dimension has to be considered, since each trip has a start and end time and the assignment of resources has to take this period into account, it was possible to incorporate this time dimension without reflecting it explicitly in the decision variables. This is an advantage because the model size (in terms of both variables and constraints) will not be a function of the number of time periods to be considered in the planning horizon. This new model, which is detailed in the next section, can be regarded as an extension of the GAP.

3. The proposed model

The requisitions for trips are the centre of the proposed model. Each requisition will require the simultaneous assignment of vehicles and drivers. For each requisition it is necessary to allocate vehicles ensuring that the total capacity (number of available seats) is greater than or equal to the total number of passengers. It is also necessary to allocate drivers, given that the number of drivers required by each requisition will depend on the number of vehicles assigned, as well as the characteristics of the requisition itself. Thus, drivers are not directly assigned to vehicles. They are assigned to requisitions, depending on the vehicles simultaneously assigned to each requisition and the number of drivers needed per vehicle for that requisition.

Time periods and trip origins/destinations do not need to be explicitly represented in the model, either in the decision variables or in the model constraints. The data that are needed to ensure that the calculated solutions are feasible are represented in two matrices. These data matrices can be automatically created from the existing information about the requisitions. One data matrix (the *non-overlapping matrix*) indicates whether two requisitions have no period of time in common. The other data matrix is the *compatibility matrix*, since it indicates whether two requisitions are compatible and thus may share resources. If the requisitions are compatible, then they can share resources (vehicles and drivers). If the requisitions are not compatible, then the same vehicle/driver cannot be simultaneously assigned to both of them unless they occur in different periods of time, i.e., they do not overlap according to the *non-overlapping matrix*.

To simplify the constraints of the model, it is assumed that compatible requisitions are ordered by increasing costs. This ordering is the same for vehicle and driver assignment costs. This is not a limiting assumption because two requisitions are compatible if both trips correspond to the same route or one trip is a sub-route of the other. Since the costs associated with assigning drivers and vehicles are mainly related to distance and travel time, it is realistic to assume that the same order applies to vehicle costs and driver costs, and even to the number of drivers required by each requisition. It must be noted that

fixed costs such as drivers' salary are not considered because they exist *a priori* and are independent of the requisitions, i.e., they do not affect the solution of the problem. Thus, if two requisitions j and j' such that $j < j'$ are compatible, then $C_{ij} \leq C_{ij'}$, $D_{mj} \leq D_{mj'}$ and $n_j \leq n_{j'}$, where, for a requisition j , C_{ij} refers to the cost of vehicle i , D_{mj} refers to the cost of driver m and n_j is the number of drivers required.

The model's data and decision variables are now detailed.

Data:

$i = 1, \dots, V$ (number of vehicles)

$j = 1, \dots, J$ (number of requisitions)

$m = 1, \dots, M$ (number of drivers)

C_{ij} = cost of allocating vehicle i to requisition j , $\forall i, j$

D_{mj} = cost of allocating driver m to requisition j , $\forall m, j$

P_j = number of passengers of requisition j , $\forall j$

Q_i = capacity (in number of available seats) of vehicle i , $\forall i$

n_j = number of drivers required for each vehicle allocated to requisition j , $\forall j$

Compatibility matrix: $A = [a_{jj'}]$, $\forall j, j': j < j'$

$a_{jj'} = \begin{cases} 1, & \text{if requisitions } j \text{ and } j' \text{ are compatible (i.e., they can share resources)} \\ 0, & \text{otherwise} \end{cases}$

Non-overlapping matrix: $B = [b_{jj'}]$, $\forall j, j': j < j'$

$b_{jj'} = \begin{cases} 1, & \text{if there is no period of time common to requisitions } j \text{ and } j' \\ 0, & \text{otherwise} \end{cases}$

Note: by definition, $a_{jj'} + b_{jj'} \leq 1$, $\forall j, j'$

Decision variables:

$x_{ij} = \begin{cases} 1, & \text{if vehicle } i \text{ is assigned to requisition } j \\ 0, & \text{otherwise} \end{cases} \quad \forall i, j$

$z_{ijj'} = \begin{cases} 1, & \text{if requisitions } j \text{ and } j' \text{ share vehicle } i \\ 0, & \text{otherwise} \end{cases} \quad \forall i, j, j': j < j'$

$z'_{jj'} = \begin{cases} 1, & \text{if requisitions } j \text{ and } j' \text{ share any vehicle} \\ 0, & \text{otherwise} \end{cases} \quad \forall j, j': j < j'$

$X'_{ij} = \begin{cases} 1, & \text{if vehicle } i \text{ is assigned to requisition } j \text{ and to requisitions compatible with } j \\ & \text{sharing resources with } j \\ 0, & \text{otherwise} \end{cases} \quad \forall i, j$

$$Y'_{mj} = \begin{cases} 1, & \text{if driver } m \text{ is assigned to requisition } j \text{ and to requisitions compatible with } j \\ & \text{sharing resources with } j \\ 0, & \text{otherwise} \end{cases} \quad \forall m, j$$

The problem can be formulated as follows:

Constraints

- Constraints (1) ensure consistency between variables X'_{ij} and x_{ij} : vehicle i can be assigned to requisition j and to requisitions compatible with it that share the same vehicle ($X'_{ij} = 1$) only if vehicle i is assigned to requisition ($x_{ij} = 1$).

$$X'_{ij} \leq x_{ij} \quad \forall i, j \quad (1)$$

- Constraints (2) and (3) ensure consistency between variables $Z'_{jj'}$ and $z_{ijj'}$: $Z'_{jj'}$ must be equal to 1 if requisitions j and j' share any vehicle, but should be equal to 0 if requisitions j and j' do not share any vehicle.

$$Z'_{jj'} \geq \frac{1}{V} \sum_{i=1}^V z_{ijj'}, \quad \forall j, j': j < j' \quad (2)$$

$$Z'_{jj'} \leq \sum_{i=1}^V z_{ijj'}, \quad \forall j, j': j < j' \quad (3)$$

- A sufficient number of vehicles with enough capacity must be assigned to each requisition to guarantee that all passengers will travel without exceeding the capacity of each vehicle. The passengers can be distributed by several vehicles and a vehicle can transport passengers from different requisitions, and both situations can occur simultaneously.

$$\sum_{i=1}^V Q_i x_{ij} \geq P_j + \sum_{j': j < j'} P_{j'} Z'_{jj'}, \quad \forall j \quad (4)$$

- A vehicle can only be assigned to more than one requisition if the requisitions are compatible or they have no common period of time (corresponding to a vehicle being used in two requisitions, but over different periods of time).

$$x_{ij} + x_{ij'} \leq (a_{jj'} + b_{jj'}) + 1, \quad \forall i, j, j': j < j' \quad (5)$$

- The number of drivers assigned to a requisition depends on the number of drivers per vehicle required by this requisition and the number of vehicles assigned to the requisition. If compatible

requisitions share vehicles, the number of drivers required by the most demanding requisition is taken (the one with the highest costs).

$$\sum_{m=1}^M Y'_{mj} \geq n_j \sum_{i=1}^V X'_{ij}, \quad \forall j \quad (6)$$

- Two requisitions cannot share a vehicle if they occur in overlapping periods of time or they are not compatible.

$$z_{ijj'} \leq (1 - b_{jj'}) a_{jj'} \frac{x_{ij} + x_{ij'}}{2}, \quad \forall i, j, j': j < j' \quad (7)$$

- If two requisitions are compatible and use the same vehicle, then they share this vehicle.

$$z_{ijj'} \geq a_{jj'} + x_{ij} + x_{ij'} - 2, \quad \forall i, j, j': j < j' \quad (8)$$

- A given driver can only be assigned to two requisitions if they share the same vehicle (because they are compatible) or they occur in non-overlapping periods of time.

$$Y'_{mj} + Y'_{mj'} \leq \sum_{i=1}^V z_{ijj'} + b_{jj'} + 1, \quad \forall m, j, j': j < j' \quad (9)$$

- If a given vehicle is used by just one requisition, then its cost is considered; however, if the vehicle is shared by several compatible requisitions, then only the highest cost should be included.

Therefore, if $\sum_{j': j < j'} z_{ijj'} = 0$ for a given j , i.e., there are no requisitions (or no requisitions with a

longer journey) sharing vehicle i with requisition j , then individual costs of requisition j must be considered. Otherwise, the costs associated with the most expensive requisition will be considered (according to the assumption that requisitions are ordered in increasing order of costs and there is consistency in the growth of costs C_{ij} and D_{mj}). Thus, the last requisition j of grouped trips is the one corresponding to the longest journey, leading to costs C_{ij} and D_{mj} of this journey being higher than those of the previous ones. Accordingly, for requisitions that share the same vehicle i , the variable X'_{ij} will be equal to 1 only for the requisition with the longest journey, i.e., the last index j .

$$X'_{ij} \geq x_{ij} - \sum_{j': j < j'} z_{ijj'}, \quad \forall i, j \quad (10)$$

- If requisitions j and j' are compatible and share the same vehicle i , i.e., $z_{ijj'} = 1$, and driver m is assigned to requisition j' with a longer journey, i.e., $Y'_{mj'} = 1, j < j'$, then Y'_{mj} (which indicates whether driver m is assigned to requisition j or not) must be equal to 0 in order to guarantee that the correct costs are considered in the objective function (the costs associated with the longest journey only). Constraints (11) ensure this requirement. Y'_{mj} is free to be 1 if j is the requisition with the

longest journey between requisitions sharing vehicles or if there is no vehicle being shared between requisition j and other requisitions. Constraints (6) ensure the proper assignment of Y'_{mj} when they are not fixed by constraints (11).

$$Y'_{mj} \leq 2 - z_{ijj'} - Y'_{mj'} , \quad \forall i, m, j, j': j < j' \quad (11)$$

Objective function

$$\text{Min} \sum_{i=1}^V \sum_{j=1}^J C_{ij} X'_{ij} + \sum_{m=1}^M \sum_{j=1}^J D_{mj} Y'_{mj} \quad (12)$$

The objective function aims at minimizing the total cost, including vehicle costs (first term) and driver costs (second term).

The objective function (12) does not consider the number of vehicles that are being shared. This means that there will be many alternative optimal solutions for a given instance, since it will be possible to consider that two requisitions are sharing a vehicle, even if that is not necessary (the vehicles that are being assigned independently to each of the requisitions have the needed capacity, but they fulfil the conditions for sharing, so it will be indifferent, in terms of objective function values, to consider the sharing or not). This situation can be avoided by including a small perturbation in the objective function that will eliminate all the unnecessary sharing:

$$\text{Min} \sum_{i=1}^V \sum_{j=1}^J C_{ij} X'_{ij} + \sum_{m=1}^M \sum_{j=1}^J D_{mj} Y'_{mj} + \varepsilon \sum_{i=1}^V \sum_{j=1}^J \sum_{j':j < j'} z_{ijj'} \quad (13)$$

In (13), ε is a positive very small number.

From the computational tests performed, it was possible to conclude that the introduction of this perturbation in the objective function decreases the computational time needed to solve the problem by the general solver.

The number of variables is: $V \times J$ for variables x_{ij} and X'_{ij} , $V \times \frac{J(J-1)}{2}$ for variables $z_{ijj'}$, $\frac{J(J-1)}{2}$ for variables $Z'_{jj'}$ and $M \times J$ for Y'_{mj} . The total number of variables is $2(V \times J) + (V + 1) \times \frac{J(J-1)}{2} + M \times J$. The total number of constraints is $2J(V + 1) + (2 + 3V + M + V \times M) \times \frac{J(J-1)}{2}$. This model has a large number of variables and constraints, however the general solver used (*cplex*) eliminates a significant number of them in the pre-solve stage.

4. Application of the model to an example

To illustrate the proposed model, let us take a small numerical example with 4 requisitions, 4 vehicles and 5 drivers. The data are presented in Table 1.

As Table 1 shows, requisition 1 (req1) is compatible with requisition 2 (req2), and requisition 3 (req3) is compatible with requisition 4 (req4) (*cf.* Compatibility matrix). The model assumes that compatible requisitions are in order of non-decreasing cost, so $C_{i1} \leq C_{i2}$, $D_{m1} \leq D_{m2}$ and $C_{i3} \leq C_{i4}$, $D_{m3} \leq D_{m4}$ for all vehicles i and all drivers m . Requisitions 1 and 2 do not have any period of time that coincides with requisitions 3 and 4 (*cf. non-overlapping matrix*), so the same vehicles and drivers can be assigned to these two groups, {req1, req2} and {req3, req4}.

An optimal solution to the problem was obtained using the *cplex* solver, with an objective function value (cost) of 622. This solution is shown in Table 2.

As can be seen in Table 2, req1 and req2 share vehicles and drivers in the optimal solution. Passengers of req3 are transported in a vehicle different from the vehicles assigned to req4, although these requisitions are also compatible with each other.

Req1 and req2 have to transport 3+6=9 passengers and there is a vehicle with seating capacity for 16, which would be enough for both requisitions. However, it is not used because it costs more to use this vehicle (plus two drivers) for req2 than to use two vehicles, v1 and v3, with four drivers (two drivers for each vehicle). The req1 trip corresponds to a sub-route of the trip of req2 (req1 requires only one driver), so the req1 passengers can travel with passengers of req2 without increasing the cost of the req2 trip alone. In practice, all the req1 passengers can travel in just one vehicle, v1 or v3, but both vehicles go to the req2 destination.

Req3 has to transport 4 passengers and only one vehicle (v4 with capacity of 5) and one driver (m1) is assigned to this requisition.

Req4 is fulfilled by the assignment of vehicles v1 and v3 (which are also assigned to req1 and req2, but these requisitions do not have periods of time that overlap with req4) and four drivers, since req4 requires two drivers per vehicle. Req4 has to carry 10 passengers. Again, there is a vehicle, v2, with seating for 16 people that could be used to carry all the req4 and req3 passengers. However, the minimum cost solution does not use it; thus, these requisitions are not grouped and three vehicles with five drivers are used to satisfy the two requisitions at minimum cost.

This example illustrates the ability of the model to distribute the passengers of one requisition among several vehicles while simultaneously allowing vehicles to be shared by several requisitions. The example also shows the case in which sharing resources is not economically advantageous (although possible) and, thus, the trips are made using separate vehicles and drivers.

Table 1 – Data of the example

Vehicles (i)	v1	v2	v3	v4	
Requisitions (j)	req1	req2	req3	req4	
Drivers (m)	m1	m2	m3	m4	m5
Costs for vehicles (C_{ij})		req1	req2	req3	req4
	v1	75	250	50	60
	v2	100	400	220	300
	v3	60	95	60	80
	v4	55	280	100	120
Costs for drivers (D_{mj})		req1	req2	req3	req4
	m1	1	2	3	4
	m2	2	3	4	5
	m3	5	6	7	8
	m4	1	2	3	4
	m5	3	4	5	6
Vehicle capacity (Q_i)	v1	v2	v3	v4	
	5	16	5	5	
	req1	req2	req3	req4	
Number of passengers (P_j)	3	6	4	10	
No. of drivers required (n_j)	1	2	1	2	
Compatibility matrix ($a_{jj'}$)		req1	req2	req3	req4
	req1	-	1	0	0
	req2	-	-	0	0
	req3	-	-	-	1
Non-overlapping matrix ($b_{jj'}$)		req1	req2	req3	req4
	req1	-	0	1	1
	req2	-	-	1	1
	req3	-	-	-	0

Table 2 – Optimal solution of the illustrative example

Requisitions	Vehicles	Drivers
req1, req2	v1, v3	m1, m2, m4, m5
req3	v4	m1
req4	v1, v3	m2, m3, m4, m5

5. Case Study

In this section, the mathematical model developed is applied to a case study involving the optimization of internal resources for making planned trips that are requested by the workers of a Brazilian University. This was actually the real situation that motivated the development of the model.

Data corresponding to a one-month period were used for this case study. The total number of requisitions is 122. A total of 28 vehicles are available, with nine different capacities. There are 24 drivers in total, who can be clustered in two different groups for travel assignment costs.

This instance was built and *cplex* solver (version 12.7) was used to find the optimal solution, using an Intel Xeon Silver 4116, 2,1 GHz, 12-core processor, 128 GB RAM. *Cplex* was able to find the optimal solution in 852.54 seconds. In the pre-solve stage it eliminated 209583 variables and 5649534 constraints, resulting in an adjusted problem with 14226 variables and 129484 constraints.

In this section we present the optimal solution found. Furthermore, for illustration purposes, another instance with only 40 requisitions was considered. The optimal solution for the reduced instance is also presented and discussed below. This dataset will also be used as the starting point for a number of instances that are randomly generated and that will allow a better understanding of the characteristics of the model. All data and calculated solutions are available as supplementary material.

5.1 Complete dataset

We will refer to the original case study as instance A, which considers the complete dataset. This instance was selected taking data from a typical work month. These data gather information about trips that have all the same departure point, destinations within a radius of up to 600 kilometres from the origin, and return to the point of origin. Trips have an average distance (round trip) of around 440 kilometres and range from 97 to 1067 kilometres. The travel requests, represented by $\{r_1, \dots, r_{122}\}$ consider different number of passengers $P = \{1, \dots, 52\}$. The number of drivers required per vehicle, $N = \{1, 2\}$, was also defined according to the current legislation, considering both the distance and the duration of each trip. A fleet of 28 vehicles ($V = \{v_1, \dots, v_{28}\}$) that were available in this period was considered. These vehicles had different characteristics such as the cost per kilometre and capacity representing the maximum number of passenger seats ($Q = \{3, 4, 16, 18, 19, 20, 26, 42, 46\}$). For the same period, a pool of drivers $M = \{m_1, \dots, m_{24}\}$ was available. These drivers are divided into two groups, each with its own travel cost per kilometre, due to the different working contracts.

Two matrices (compatibility and non-overlapping matrices) were created based on the known characteristics of the trips, namely, date, time and destination (as explained in Section 3). The capacity of each vehicle is shown in Table 3.

Table 3 – Vehicle capacity

Vehicle	v01	v02	v03	v04	v05	v06	v07	v08	v09	v10	v11	v12	v13	v14	v15	v16
Q	26	26	26	42	42	46	4	4	4	4	4	4	16	16	16	3

Vehicle	v17	v18	v19	v20	v21	v22	v23	v24	v25	v26	v27	v28
Q	18	19	20	4	4	4	4	4	4	4	4	4

Table 4 presents a summary of the solution obtained.

Table 4– Results for Instance A

	Nr.	%
Objective function value	43 309.92	
Total Requests	122	
Total Passengers	837	
Total Vehicle Assignments (TVA)	97	
Total Driver Assignments	107	
Total Seats Available on Assigned Vehicles	980	
% Empty Seats on Total Assigned Vehicles		14.59
Trips sharing resources	91	74.59
Trips not sharing resources	31	25.41
Passengers in trips sharing resources	513	61.29
Passengers in trips not sharing resources	324	38.71
Vehicles shared by trips	52	53.61
Vehicles not shared by trips	45	46.39
Drivers Assigned to shared trips	57	53.27
Drivers Assigned to non-shared trips	50	46.73
% Empty Seats in vehicles on shared trips		8.23
% Empty Seats in vehicles on non-shared trips		23.04

It was possible to guarantee the fulfilment of all requested trips. The model enabled the passengers of a given requisition to be assigned to more than one vehicle, even in cases where vehicles are shared. The use of shared vehicles proves to be a valuable option since a large number of trips share vehicles, thus contributing to a cost reduction. Only 31 of the 122 travel requests do not share any vehicle.

A total of 91 travel requisitions share vehicles and drivers in various proportions. Vehicle sharing also helps to increase the seat occupation in the vehicles used: 91.77% of the vehicles' total capacity is used in shared vehicles, against 76.96% in non-shared vehicles.

The optimal solution found by the general solver for this dataset was compared with two other solutions: the optimal solution found for the same problem but without the possibility of shared vehicles and shared drivers; the actual vehicle and driver assignments made by the institution in the month considered. The comparison between these solutions in terms of cost is shown in Figure 1.

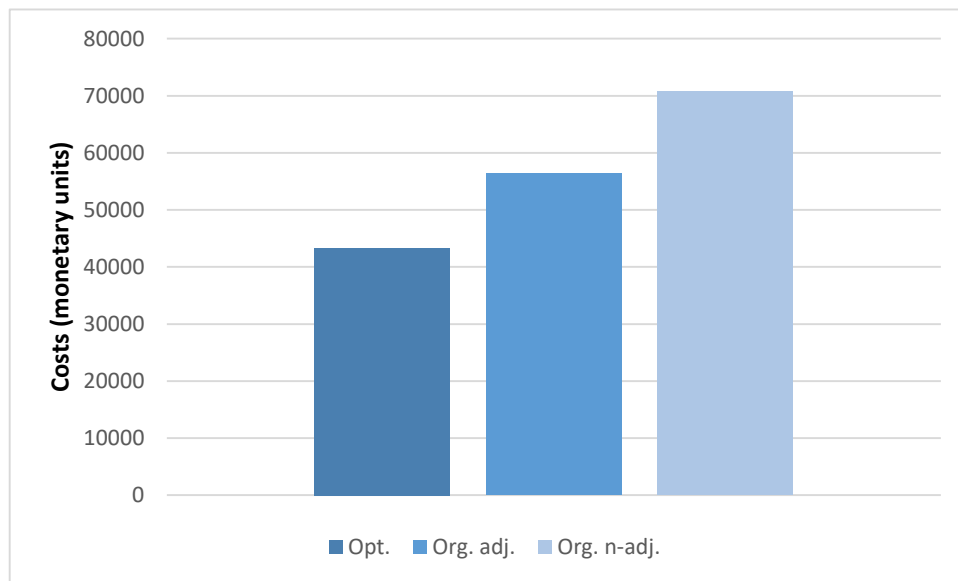


Figure 1- Comparison of the solutions obtained by the optimization model, the organization with manual adjustments and the model without allowing shared resources (Opt.= optimized with the model; Org.adj. = adjusted by the organization; Org.n-adj. = without vehicle and driver sharing)

The first column in Figure 1 shows the total cost of the optimal solution found for the model. The second one shows the total cost of the solution implemented by the organization (without optimization, but with manual adjustments), and the third column shows the total cost of an optimization model that allows neither vehicle nor driver sharing. It is interesting to note that the solution implemented by the institution is better than the one found with an optimization model that does not allow the sharing of resources. This can be explained by the fact that manual adjustments could assign shared resources to some requisitions. In this case, the use of the proposed optimization model would achieve a cost reduction of 23.18% relative to the implemented solution.

Figure 2 details part of the resources' assignment in the optimal solution found. Requisitions r34 and r36 have in total 3 passengers. They are compatible so it is possible for them to share a vehicle. As the longest trip (r36) requires two drivers, two drivers are indeed assigned. An example of slightly more complex sharing can be seen in requisitions r39, r40 and r41. These are all compatible requisitions, considering 51 passengers in total. Passengers of requisition r40 all use vehicle v15. As the vehicle

would have four seats left, one of them is used by the only passenger of r39. The three vacant seats are used by r41, which needs further vehicles to accommodate all its 38 passengers. It is possible to achieve 100% seat occupation by sharing the vehicles. Another example can be seen if we look at requisitions r33, r35 and r38. In this case, r38 and r35 share each of their vehicles separately with r33 and not with each other. However, to complete the number of available seats r33 needs one more vehicle.

Figure 2 – Examples of requisitions sharing resources in the optimal solution found for instance A

Requisition				Vehicle												Driver													
Main	Shared			P	N	v07	v08	v09	v10	v11	v12	v13	v14	v15	v16	Q	Vc	m13	m14	m15	m16	m17	m18	m19	m20	m21	m22	m23	m24
r33	r35	r38		33	1		v08					v14	v15		16	-17				m16									
r35	r33			1	1							v14			16	15	m13												
r38	r33			1	1		v08								4	3								m20					
r34	r36			2	1									v16		-2													
r36	r34			1	2									v16		3	2	m14										m23	
r39	r40	r41		1	1									v15		-1													
r40	r39	r41		12	1									v15		16	4		m15										
r41	r39	r40		38	1							v13	v14	v15	v16	35	-3	m13	m14									m24	

5.2 Instance of reduced size

For illustration purposes, a smaller instance was considered where the total number of requisitions was cut to 40. Drivers and vehicles were unchanged. This instance of the problem will be referred to as instance B. *Cplex* needed 50.39 seconds of computational time. A summary of the results is shown in Table 5.

As was seen in the previous instance, many trips are sharing vehicles (77.5%). Another interesting feature of the optimal solution is the efficient use of the available seats, reaching 97.37% in the case of shared vehicles and 73.55% for non-shared vehicles. Similar figures of vehicle sharing were found in this solution to those found for Instance A.

Table 5– Results for Instance B

	Nr.	%
Objective function value	16 054.40	
Total Requests	40	
Total Passengers	311	
Total Vehicle Assignments (TVA)	35	
Total Driver Assignments	39	
Total Seats Available on Assigned Vehicles	349	
% Empty Seats on Total Assigned Vehicles	38	10.89
Trips sharing resources	31	77.50
Trips not sharing resources	9	22.50

Passengers in trips sharing resources	222	71.38
Passengers in trips not sharing resources	89	28.62
Vehicles shared by trips	22	62.86
Vehicles not shared by trips	13	37.14
Drivers Assigned to shared trips	23	58.97
Drivers Assigned to non-shared trips	16	41.03
% Empty Seats in vehicles on shared trips		2.63
% Empty Seats in vehicles on non-shared vehicles		26.45

6. Analysis of different datasets

Further computational tests were carried out on random instances obtained from Instance B in order to analyse the solutions produced by the model under different circumstances. The impact of different configurations of the available fleet (small, medium or large vehicles), reduction of the total number of seats, reduction of the number of drivers and variation of the individual costs is analysed in this section with respect to the running time and the objective function value (cost) of the optimal solution.

Instance B has 40 requisitions and considers all the vehicles and drivers of the real case (Instance A), i.e. 28 vehicles (with 376 available seats) and 24 drivers. The vehicle fleet includes 16 small vehicles (with 3 or 4 seats), 9 medium vehicles (from 16 to 26 seats) and 3 large vehicles (two with 42 seats and one with 46 seats). In all the instances generated for this analysis the number of requisitions was kept constant and equal to 40.

The first group of instances (*case 1 to case 4*) aimed at studying the effect of *reducing the total seat capacity considering different configurations of the vehicle fleet*. We have considered a fleet with an overall capacity close to 100 seats: *case 1* – only small vehicles were considered; *case 2* – only medium vehicles were considered; *case 3* – only large vehicles were considered; *case 4* – a diversified fleet was kept as in Instance B, but with fewer vehicles. The vehicles were selected at random from the respective category (small, medium or large) in Instance B, enabling repetitions (i.e., it is possible to have a fleet with several identical vehicles resulting from a multiple selection of the same original vehicle). *Case 3* has a higher number of seats (172) because only large vehicles were considered and at least 4 vehicles are needed to ensure feasibility.

The second group of instances (*case 5 to case 7*) aimed at studying the effect of *reducing the number of drivers available*: *case 5* – 20 drivers; *case 6* – 15 drivers; *case 7* – 12 drivers. For each case the

drivers were selected at random from the ones in Instance B. The original fleet of 28 vehicles was considered in this group.

The third group of instances (*case 8 to case 12*) aimed at studying the effect of *changing the costs*. These five instances consider the original fleet of vehicles and all drivers. All these cases were randomly generated from Instance B and using similar rules: the individual costs C_{ij} (cost of allocating vehicle i to requisition j) and D_{mj} (cost of allocating driver m to requisition j) suffered a positive or negative variation of up to 25% with respect to its original value. Thus, each C_{ij} was replaced with a random value in $[0.75C_{ij}, 1.25C_{ij}]$; since costs should be non-decreasing in compatible requisitions, a second rule was applied after changing all C_{ij} to ensure consistent values: if requisition j' is compatible with j , $j' > j$, then $C_{ij'} \leftarrow \max\{C_{ij}, C_{ij'}\}$. A similar procedure was applied to the generation of costs D_{mj} .

The characteristics that distinguish these instances from Instance B are summarized in Table 6. Full data and calculated solutions are available as supplementary material.

Table 6 – Characteristics of the random instances.

Group	Case	Characteristics
1	case 1	26 small vehicles → 103 seats
	case 2	5 medium vehicles → 106 seats
	case 3	4 large vehicles → 172 seats
	case 4	6 vehicles: 3 small, 2 medium, 1 large → 96 seats
2	case 5	20 drivers
	case 6	15 drivers
	case 7	12 drivers
3	case 8	
	... case 12	Random variations in costs of up to 25%

A summary of the results obtained using *cplex* is shown in Table 7. For sake of clarity, the results of Instance B are repeated in this table.

Table 7 – Results of the random instances.

Group	Case	Objective value	<i>Cplex</i> GAP	Computational time (sec)	% Empty seats in shared vehicles	% Empty seats in non-shared vehicles
	Instance B	16,054.40	0%	50.39	0.44	26.45
1	case 1	24,699.45	0%	6,405.63	1.54	6.06
	case 2	26,712.65	0%	2.45	16.28	62.16
	case 3	56,113.90	0%	0.52	43.81	83.33
	case 4	23,628.17	0%	3.42	9.17	37.18
2	case 5	16,269.13	0%	43.75	2.16	26.45
	case 6	16,269.13	0%	26	2.16	26.45
	case 7	16,269.13	0%	23	2.16	26.45
	case 8	14,840.77	0%	55.01	2.45	10.47

3	case 9	14,933.50	0%	52.44	3.24	24.51
	case 10	15,110.39	0%	51.98	3.82	25.58
	case 11	15,316.97	0%	54.2	2.28	21.54
	case 12	14,682.40	0%	51.34	2.58	21.93

The analysis of the results leads to the following conclusions:

- In the first group of instances, *case 1* (which considers only small vehicles) turned out to be much more difficult to solve than the other cases. This finding is in accordance with what might be expected since more vehicles and drivers have to be assigned in *case 1*, so there are many more combinations of possible assignments.
- It seems that the fleet composition has a higher impact than the total number of seats available on the computational time.
- In the second group, all instances have similar solutions, and these solutions are similar to the solution to Instance B. Therefore, cutting the number of drivers in these instances did not change the solution structure. There are some variations in the computational times, but no pattern can be established: the computational time is really problem dependent.
- In the third group, in which all costs were changed, there are no substantial differences in the outcomes of the five cases, or in the computational times. Therefore, costs do not seem to have much influence on the computational time..
- Concerning occupancy rates in shared and non-shared vehicles, all the solutions to these instances involve a percentage of empty seats that is smaller in shared vehicles than in non-shared vehicles (i.e. vehicles not shared by several requisitions). This finding is in accordance with the results for Instances A and B, supporting the assertion of efficiency in the use of seats, mainly in shared vehicles.

As a main conclusion, we can say that these results suggest that the fleet composition is the issue that most influences the computational time needed to solve the problem when the number of requisitions is kept constant and various changes are tried in the fleet configuration, in the number of drivers or in the structure of costs.

7. Genetic algorithm combined with *Cplex*

The general solver was able to calculate optimal solutions in reasonable computational times for all instances tested. However, given the combinatorial nature of the problem, it is expected that, for larger instances of the problem, this might not happen. This observation has motivated the development of a dedicated heuristic procedure.

Developing a metaheuristic to deal with this problem is a highly complex task. The main problem faced has to do with solution feasibility. These difficulties are felt in the construction of initial solutions and, even more, in the design of operators that do not "damage" too much the solutions in terms of feasibility

because it is very difficult to recover it. It is difficult to keep feasibility due to several issues. Listing just two, among many other constraints: trips that occur in periods of time with intersection cannot be assigned to the same vehicles and drivers; the vehicle/set of vehicles assigned to a trip/set of trips sharing resources must have enough capacity to attend the requisition/s. It is very hard to guarantee feasibility considering the usual operators used in metaheuristics, since they all imply making and breaking the sharing of vehicles and drivers. These difficulties are far beyond what has to be considered in designing a metaheuristic for GAP, for instance, mainly due to the existence of two interdependent resources. The representation of a solution is also nontrivial (contrarily to the GAP). We have tested several representations and all of them suffered from important drawbacks regarding solution feasibility. When most of the generated solutions are infeasible, the metaheuristic does not work and does not converge to an interesting solution.

Considering all the described difficulties, a hybrid metaheuristic (*matheuristic*) was developed for this problem, which combines a genetic algorithm with a mathematical programming solver (*cplex*). The chromosomes do not codify the complete solution, but only the assignment of vehicles to requisitions. Furthermore, chromosomes do not represent explicitly this part of the solution, but they rather consist of patterns of 1's and 0's, where 1 means free variables and 0 means variables fixed to 0 regarding the assignment of vehicles to requisitions. Therefore, a chromosome s is a binary matrix $s = [s_{ij}], i=1, \dots, V$ (number of vehicles), $j=1, \dots, J$ (number of requisitions), which represents a mask for obtaining a solution: $s_{ij} = 0$ means that $x_{ij} = 0$, i.e., the vehicle i cannot be assigned to requisition j , and $s_{ij} = 1$ means that x_{ij} is free to be 0 or 1 (the vehicle can be assigned or not). For each chromosome s , the generalized assignment model (1)-(13) with x_{ij} fixed to 0 for i, j such that $s_{ij} = 0$ is solved using *cplex*, considering a time limit. Setting a time limit means that *cplex* can or cannot reach the optimal solution for the subproblem represented by the chromosome. The more degrees of freedom it has (number of variables with a value that can take either 0 or 1) the more time it will need to find the optimal solution. These chromosomes can be interpreted as representing search spaces that are smaller than the entire feasible region, for the assignment of vehicles to requisitions. The feasibility of the solution is guaranteed by the use of a general solver. It is possible that no feasible solution exists in the solution space codified by the chromosome, or *cplex* is not able to find one because of the imposed time limit. If this is the case, that chromosome will have a very poor fitness (large cost value).

The evolution of these solutions will make them progress towards more interesting search spaces, where it will be easier for *cplex* to find the optimal solution. Table 8 shows an example of a chromosome.

Table 8 – Example of a chromosome

	req.1	req.2	req.3	req.4	req.5
v1	1	0	0	1	1
v2	0	0	1	0	1
v3	1	1	0	1	0
v4	0	1	1	0	0

Chromosome \mathbf{s}' represents a mask for which v1 cannot be assigned to req.2 nor to req.3; v2 cannot be assigned to req.1, req.2 nor to req.4; etc. This means that $x_{12} = 0, x_{13} = 0, x_{21} = 0, x_{22} = 0, x_{24} = 0$, etc., are fixed a priori. Therefore, the problem is solved just considering the x_{ij} variables corresponding to $s_{ij} = 1$.

After generating an initial population \mathbf{P} of N chromosomes, $\mathbf{P} = \{\mathbf{s}^n, n = 1, \dots, N\}$, the MILP sub-problem for each mask \mathbf{s}^n is solved using the *cplex* (with a low time limit). The objective function value (cost) of the solution is the fitness of \mathbf{s}^n . The reproduction and mutation operators of the genetic algorithm are then applied considering the following scheme:

- Two parents \mathbf{s}' and \mathbf{s}'' are selected from \mathbf{P} by binary tournament. A crossover row $k \in \{1, \dots, V - 1\}$ is randomly drawn, separating the vehicles of masks \mathbf{s}' and \mathbf{s}'' into two parts. Two offspring are generated, \mathbf{o}' and \mathbf{o}'' such that $\mathbf{o}' = \begin{bmatrix} s'_{ij}, i = 1, \dots, k \\ s''_{ij}, i = k + 1, \dots, V \end{bmatrix} j = 1, \dots, J$ and $\mathbf{o}'' = \begin{bmatrix} s''_{ij}, i = 1, \dots, k \\ s'_{ij}, i = k + 1, \dots, V \end{bmatrix} j = 1, \dots, J$. An example is shown in Figure 3.
- The mutation operator is applied to \mathbf{o}' and \mathbf{o}'' with probability P_{mut} of changing each gene o_{ij} (we have used $P_{mut} = 0.1/V$).

		req.1	req.2	req.3	req.4	req.5			req.1	req.2	req.3	req.4	req.5	
$\mathbf{s}' =$	v1	1	0	0	1	1		$\mathbf{s}'' =$	v1	0	0	1	0	0
	v2	0	0	1	0	1			v2	1	1	0	0	1
	v3	1	1	0	1	0			v3	0	0	1	1	1
	v4	0	1	1	0	0			v4	1	0	1	0	1
		req.1	req.2	req.3	req.4	req.5			req.1	req.2	req.3	req.4	req.5	
$\mathbf{o}' =$	v1	1	0	0	1	1		$\mathbf{o}'' =$	v1	0	0	1	0	0
	v2	1	1	0	0	1			v2	0	0	1	0	1
	v3	0	0	1	1	1			v3	1	1	0	1	0
	v4	1	0	1	0	1			v4	0	1	1	0	0

Figure 3 – Example of crossover between \mathbf{s}' and \mathbf{s}'' with crossover row $k = 1$.

An offspring population \mathbf{O} with N children is obtained after applying steps 1 and 2 during $N/2$ times (assuming N an even number).

A selection procedure is then applied to select the N individuals of \mathbf{P} and \mathbf{O} that will integrate the population for the next generation. The fittest chromosome will always move to the next generation. The other $N-1$ individuals are selected by binary tournaments, each one made between an element of \mathbf{P} and an element of \mathbf{O} . The chromosome with lower cost (best fitness) is selected and deleted for further selection.

The genetic algorithm stops after a predefined number of G generations and the best solution is returned.

The main parameters that must be set for this *matheuristic* are: the number of solutions in the population (N); the number of generations (G); the time limit imposed to *cplex* for each chromosome; the number of free variables that each chromosome codifies in the initial population.

The first three parameters are directly linked to the computational time: more individuals, more generations and a greater time limit for *cplex* means increased computational times. It is important to find a compromise between allowed computational time and solution quality.

The number of 1's and 0's allowed in a chromosome is related with the degrees of freedom *cplex* will have, and may also affect the computational time. In the extreme cases, if all s_{ij} are set to 1, then there is no reduction in the number of variables and the corresponding sub-problem has the same dimension as the original problem. On the opposite case, if $s_{ij} = 0$ for many i, j , the corresponding sub-problem may be infeasible. In the initial population, each chromosome is randomly generated considering a predefined probability $Prob_1$ of generating 1 and $1-Prob_1$ the probability of generating 0. If $Prob_1$ is small, the number of 0's in each chromosome is high and the corresponding sub-problem is either infeasible or its resolution is fast (because it has only a few number of variables). If $Prob_1$ is large, there is a greater chance that a lower cost solution will be obtained, but the resolution of the sub-problem may require a longer computational time (*cplex* will probably use all the allowed time). However, it has been observed that including a sub-optimal solution (obtained with a very short computational time) to the original problem may improve the search process of the genetic algorithm. Thus, the initial population includes a chromosome mask of only 1's and the others are generated with lower $Prob_1$.

Several different combinations of parameters were tested considering instance *A*. The best compromise results were obtained considering $Prob_1$ equal to 10%, populations of 10 solutions, a maximum of 10 generations and a maximum time limit for *cplex* equal to 60 seconds. The *matheuristic* was run 10 times. The average total computational time was 1 hour. The optimal solution was found in 5 out of 10 runs. The average gap to the optimal cost was 0.24%, and the worst gap was 3.64%. Figure 4 presents the evolution of the objective function value in one of the runs that led to the calculation of the optimal solution. It is possible to observe the ability of the algorithm to improve the best-known value of the objective function in almost all generations.

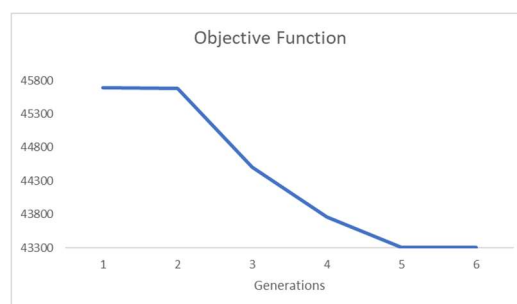


Figure 3 – Objective function values using the *matheuristic*

Increasing $Prob_1$ without increasing the maximum time limit for *cplex* diminishes the diversity of the population, and the algorithm converges prematurely. Increasing the number of individuals or the number of generations has a strong impact on the total computational time. We also tested increasing

the number of individuals, decreasing the *cplex* time limit, but the results were worse, especially in terms of the worst gap achieved. Comparing the computational times and the quality of the solutions, it can be stated that *cplex* is a better option regarding this instance.

We have tested the *matheuristic* in another instance, randomly generated from instance A, where the number of requisitions was increased to 200, keeping all the resources unchanged. This is a much more challenging problem to *cplex*. Within a time limit of 4 hours, it was not possible to find the optimal solution. The best solution found had a gap of 1.14%. Considering that this problem has a higher dimension, the algorithm was tested considering a *cplex* time limit of 90 seconds for calculating the fitness of each individual, 20 individuals in the population and a total number of generations equal to 8. This means that the total time is similar to the time limit imposed to *cplex* to solve the original problem. The average gap was equal to 2.09% (since the optimal solution is not known, the gap was calculated with respect to the best limit given by *cplex*).

Among the 12 instances that were tested in section 6, *case 1* was particularly difficult to solve by *cplex*. This is the instance characterized by having small vehicles only. *Cplex* needed 107 minutes to find the optimal solution. The *matheuristic* was run 10 times, considering a *cplex* time limit of 60 seconds for calculating the fitness of each individual, 10 individuals in the population and a total number of generations equal to 10. $Prob_1$ was kept equal to 10%. It was always able to find the optimal solution. The average computational time was 50 minutes. The optimal solution was found, on average, in the second generation and, in the worst case, in the third generation. One of the reasons that can justify the better performance of the *matheuristic* in this case is the fact that, with small vehicles only, the number of possible vehicle combinations that can be assigned to requisitions is much larger than with vehicles with bigger capacity. This forces the general solver to build a very large branch&bound tree.. In the *matheuristic*, each individual codifies a much smaller search space in which *cplex* is able to find an optimal solution in a very short time.

Although the computational results were not totally favorable to the heuristic, it is possible to disguise future developments based on the main idea of this heuristic, i.e., chromosomes codifying search spaces instead of complete solutions:

- One parameter that greatly influences the heuristic behavior is the *cplex* time limit; this parameter can be set in a dynamic way, with lower values in the beginning, to promote diversification, and higher values as the generations progress, at least for the most promising individuals.
- The parameter $Prob_1$ can also be set differently for different pairs of requisitions and vehicles. It could be beneficial to define $Prob_1$ as a function of the ratio between the total number of passengers of the requisition and the capacity of the vehicle.
- It is also possible to consider the concept of islands, where each island has individuals codifying different search spaces associated with the two different resources. The evolution will be

independent in each island, but connections can be made using, for instance, the best individuals.

- Information about sharing can also be considered, making it possible to rapidly discard individuals that do not represent interesting search spaces (exploring the information gathered by the *cplex* branch&bound trees).

This will be the focus of future research.

8. Conclusions

In this paper, a model has been presented that aims at optimizing the assignment of two distinct resources to a set of tasks, where the assignment of one of the resources depends on the assignment of the other resource. Moreover, tasks can share resources if they are compatible. The presented model can thus be interpreted as an extension of GAP. As far as the authors know, this is the first time that such a problem has been tackled. It was motivated by a real case of assigning both vehicles and drivers to planned trips. Although the model has been applied to a concrete situation of vehicle and driver assignment, it can be easily adapted to many other situations where two distinct types of resources have to be assigned to a set of tasks, with constraints that link the assignment of the two resources.

From the computational tests performed, it is possible to conclude that allowing the use of shared resources is very important in terms of cost reduction. The use of the proposed optimization model for the real case, which is able to accommodate vehicle sharing between requisitions, contributed to a significantly lower cost than that provided by the non-computer-assisted solution determined by the institution. The optimal solution for the complete dataset is significantly better than the one implemented by the institution. A further analysis with random instances suggest that the composition of the vehicle fleet (i.e. number of vehicles within each category of number of seats) has a significant influence on the computational time required to solve the problem.

A *matheuristic* was also developed, based on a genetic algorithm, where each chromosome codifies a search space, instead of an individual solution. The computational results demonstrate that this heuristic is competitive when the fleet is mainly composed by vehicles of small capacity (problems that are challenging for *cplex*).

The main concern of the institution regarding the assignment of resources to planned trips is cost minimization. There are, however, other aspects of the solutions that should be analysed and taken into account in the optimization, namely equity in trips assigned to drivers. With the proposed model, it is possible that some drivers are assigned to many more trips than others. This could be seen in the optimal solution found for the complete dataset (Instance A), where the driver with the largest number of assigned requisitions will be driving on 16 trips while some drivers are not assigned to any trip. Equity measures that take into account not only the number of trips but also distances travelled and other ordinary work assignments for drivers could be considered. Even regarding vehicles, some

concerns could be included in the model. Indeed, the frequency of the use of the vehicles and miles travelled can imply very different wear and tear conditions and may therefore have an impact on the total costs associated with the fleet (particularly in the case of own vehicles). These concerns are motivating future developments of multi-objective models where objective functions other than cost minimization will be explicitly considered.

Another aspect that is not being addressed in the model proposed is the uncertainty associated with the requisitions. Although only planned trips are considered, it can happen that requisitions appear unexpectedly, for urgent trips. Moreover, trips can take more or less time to be completed than expected (several factors may be involved in travel delays, such as traffic problems and weather conditions), with implications not only in terms of costs but also in terms of resources assignment compatibility. This will also be considered in future work.

Acknowledgments

This work has been supported by the Fundação para a Ciência e a Tecnologia (FCT) under project grant UID/Multi/00308/2019. The first author has been supported by CAPES Proc. 009345/2013-02 BEX.

References

- Baldacci, R., Battarra, M., & Vigo, D. (2008). The Vehicle Routing Problem: Latest Advances and New Challenges, *43*.
- Bielli, M., Bielli, A., & Rossi, R. (2011). Trends in models and algorithms for fleet management. *Procedia - Social and Behavioral Sciences*, *20*, 4–18.
- Billhardt, H., Fernandez, A., Lemus, L., Lujak, M., Osman, N., Ossowski, S., & Sierra, C. (2014). Dynamic coordination in fleet management systems: Toward smart cyber fleets. *IEEE Intelligent Systems*, *29*, 70–76.
- Braekers, K., & Kovacs, A. A. (2016). A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B*, *94*, 355–377.
- Cheung, R., Hang, D., & N Shi. (2005). A labeling method for dynamic driver-task assignment with uncertain task durations. *Operations Research Letters*, *33*, 411–420.
- Cheung, R. K., & Hang, D. D. (2003). A time-window sliding procedure for driver-task assignment with random service times. *IIE Transactions*, *35*(5), 433–444.
- Coelho, L., Cordeau, J., & Laporte, G. (2012). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C*, *24*, 270–287.
- Coslovich, L., Pesenti, R., & Ukovich, W. (2006). Minimizing fleet operating costs for a container transportation company. *European Journal of Operational Research*, *171*, 776–786.
- Deti, P., Papalini, F., & Lara, G. de. (2017). A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega*, *70*, 1–14.
- Ertogral, K., Akbalik, A., & González, S. (2017). Modelling and analysis of a strategic fleet sizing problem for a furniture distributor “Modelling and analysis of a strategic fleet sizing problem for a furniture distributor.” *European J. Industrial Engineering*, *11*(1), 49–77.

- Fink, A., & Reiners, T. (2006). Modeling and solving the short-term car rental logistics problem. *Transportation Research Part E: Logistics and Transportation Review*, 42(4), 272–292.
- Freling, R., Huisman, D., & Wagelmans, A. P. M. (2003). Models and Algorithms for Integration of Vehicle and Crew Scheduling. Pdf. *Journal of Scheduling*, 6, 63–85.
- Gavish, B., & Pirkul, H. (1991). Algorithms for the Multi-Resource Generalized Assignment Problem. *Management Science*, 37(6), 695–713.
- Groër, C., Golden, B., & Wasil, E. (2009). The Consistent Vehicle Routing Problem. *Manufacturing & Service Operations Management*, 11(4), 630–643.
- Hoff, A., Andersson, H., Christiansen, M., Hasle, G., & Lokketangen, A. (2010). Industrial aspects and literature survey: Fleet composition and routing. *Computers and Operations Research*, 37(12), 2041–2061.
- Karsu, Ö., & Azizoğlu, M. (2014). Bicriteria multiresource generalized assignment problem. *Naval Research Logistics (NRL)*, 61(8), 621–636.
- Kiraz, B., & Topcuoglu, H. (2010). Hyper-heuristic approaches for the dynamic generalized assignment problem. In *10th IEEE International Conference on Intelligent Systems Design and Applications (ISDA)* (pp. 1487–192).
- Kogan, K., Khmel'nitsky, E., & Ibaraki, T. (2005). Dynamic Generalized Assignment Problems with Stochastic Demands and Multiple Agent--Task Relationships. *Journal of Global Optimization*, 31(1), 17–43.
- Kovacs, A. A., Golden, B. L., Hartl, R. F., & Parragh, S. N. (2014). Vehicle routing problems in which consistency considerations are important: A survey. *Networks*, 64(3), 192–213.
- Laurent, B., & Hao, J. (2007). Simultaneous vehicle and driver scheduling: A case study in a limousine rental company. *Computers & Industrial Engineering*, 53, 542–558.
- Li, L., & Fu, Z. (2002). The school bus routing problem: a case study. *Journal of the Operational Research Society*, 53, 552–558.
- Lima, F., Pereira, D., Conceição, S., & Nunes, N. (2016). A mixed load capacitated rural school bus routing problem with heterogeneous fleet: Algorithms for the Brazilian context. *Expert Systems with Applications*, 56, 320–334.
- Liu, Y., & Wang, S. (2015). A scalable parallel genetic algorithm for the generalized assignment problem. *Parallel Computing*, 46, 98–119.
- Markov, I., Varone, S., & Bierlaire, M. (2016). Integrating a heterogeneous fixed fleet and a flexible assignment of destination depots in the waste collection VRP with intermediate facilities. *Transportation Research Part B*, 84, 256–273.
- Mazzola, J. B., & Neebe, A. W. (2012). A generalized assignment model for dynamic supply chain capacity planning. *Naval Research Logistics (NRL)*, 59(6), 470–485.
- Mazzola, J. B., & Wilcox, S. P. (2001). Heuristics for the multi-resource generalized assignment problem. *Naval Research Logistics*, 48(6), 468–483.
- Mitrović-Minić, S., & Punnen, A. (2009). Local search intensified: Very large-scale variable neighborhood search for the multi-resource generalized assignment problem. *Discrete Optimization*, 6, 370–377.
- Moccia, L., Cordeau, J., Monaco, M. F., & Sammarra, M. (2009). A column generation heuristic for a dynamic generalized assignment problem. *Computers & Operations Research*, 36, 2670–2681.
- Morales, D. R., & Romeijn, H. E. (2004). The Generalized Assignment Problem and Extensions. In D.-Z. Du & P. M. Pardalos (Eds.), *Handbook of Combinatorial Optimization* (pp. 259–311). Boston: KluwerAcademic Publishers.
- Oliveira, B. B., Carravilla, M. A., Oliveira, J. F., & Toledo, F. M. B. (2014). A relax-and-fix-based

- algorithm for the vehicle-reservation assignment problem in a car rental company. *European Journal of Operational Research*, 237(2), 729–737.
- Öncan, T. (2007). A Survey of the Generalized Assignment Problem and Its Applications. *INFOR: Information Systems and Operational Research*, 45(3), 123–141.
- Park, J., & Kim, B. (2010). The school bus routing problem: A review. *European Journal of Operational Research*, 202, 311–319.
- Park, J., Tae, H., & Kim, B. (2012). A post-improvement procedure for the mixed load school bus routing problem. *European Journal of Operational Research*, 217, 204–213.
- Pentico, D. W. (2007). Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176(2), 774–793.
- Peters, E., Matta, R. de, & Boe, W. (2007). Short-term work scheduling with job assignment flexibility for a multi-fleet transport system. *European Journal of Operational Research*, 180, 82–98.
- Riera-Ledesma, J., & Salazar-González, J. (2012). Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. *Computers & Operations Research*, 39, 391–404.
- Sethanan, K., & Pitakaso, R. (2016). Improved differential evolution algorithms for solving generalized assignment problem. *Expert Systems with Applications*, 45, 450–459.
- Shtub, A., & Kogan, K. (1998). Capacity planning by the dynamic multi-resource generalized assignment problem (DMRGAP). *European Journal of Operational Research*, 105, 91–99.
- Simão, H. P., Day, J., George, A. P., Gifford, T., Nienow, J., & Powell, W. B. (2009). An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application. *Transportation Science*, 43(2), 178–197.
- Souza, D. S., Santos, H. G., & Coelho, I. M. (2017). A Hybrid Heuristic in GPU-CPU Based on Scatter Search for the Generalized Assignment Problem. *Procedia Computer Science*, 108C, 1404–1413.
- Spliet, R., & Dekker, R. (2016). The driver assignment vehicle routing problem. *Networks*, 68(3), 212–223.
- Woodcock, A. J., & Wilson, J. M. (2010). A hybrid tabu search/branch & bound approach to solving the generalized assignment problem. *European Journal of Operational Research*, 207, 566–578.
- Xu, Y., Wang, X., & Sun, T. (2018). Heuristic routing algorithm toward scalable distributed generalized assignment problem. *Soft Computing*, 22, 845–859.
- Yagiura, M., Iwasaki, S., Ibaraki, T., & Glover, F. (2004). A very large-scale neighborhood search algorithm for the multi-resource generalized assignment problem. *Discrete Optimization*, 1(1), 87–98.
- Zäpfel, G., & Bögl, M. (2008). Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics*, 113, 980–996.