Operations Research Perspectives 5 (2018) 334–356

Contents lists available at ScienceDirect

Operations Research Perspectives

journal homepage: www.elsevier.com/locate/orp

Integrating pricing and capacity decisions in car rental: A matheuristic approach

Beatriz Brito Oliveira*, Maria Antónia Carravilla, José Fernando Oliveira

INESC TEC, Faculty of Engineering, University of Porto, Portugal

ARTICLE INFO

Keywords:
Car rental
Pricing
Fleet management
Matheuristic
Genetic algorithm

ABSTRACT

Pricing and capacity decisions in car rental companies are characterized by high flexibility and interdependence. When planning a selling season, tackling these two types of decisions in an integrated way has a significant impact.

This paper tackles the integration of capacity and pricing problems for car rental companies. These problems include decisions on fleet size and mix, acquisitions and removals, fleet deployment and repositioning, as well as pricing strategies for the different rental requests.

A novel mathematical model is proposed, which considers the specific dynamics of rentals on the relationship between inventory and pricing as well as realistic requirements from the flexible car rental business, such as upgrades. Moreover, a solution procedure that is able to solve real-sized instances within a reasonable time frame is developed. The solution procedure is a matheuristic based on the decomposition of the model, guided by a biased random-key genetic algorithm (BRKGA) boosted by heuristically generated initial solutions. The positive impact on profit, of integrating capacity and pricing decisions versus a hierarchical/sequential approach, is validated.

1. Introduction

Car rental companies face several decisions related with their capacity, including decisions on fleet size/mix, acquisitions and deployment, which are significantly connected with the pricing of the rentals that are fulfilled. This paper proposes a new mathematical model for the integration of these problems, as well as a solution procedure that is able to solve realistically sized instances within a reasonable time frame.

1.1. Motivation

The car rental business is a relevant sector within the current mobility systems, which has been significantly growing in the past years. In the U.S., the revenue gains have grown 4% in 2015, with an average fleet growth of 5% [1]. Moreover, the use of rental cars is also expected to grow in the future beyond the traditional corporate and leisure utilization, towards becoming an occasional alternative to owning a private car [2].

This is a business that faces interesting and challenging operations management issues, in which quantitative methods that support decision-making are becoming critical. In 2013, the CEO of Hertz, one of the main global players in the market, highlighted how technology is becoming the key competitive advantage of car rental companies and how it has been taking a central space even in the governing structure of the organizations [3]. These challenges are important for practitioners yet the literature has only recently gained momentum in structuring and studying the interesting fleet operations and revenue management problems faced by car rentals. The main differentiation of this business, when compared with more traditionally studied transportation sectors, is its flexibility. The inherent flexibility of the fleet (mobility of the vehicles), the flexibility associated with acquiring vehicles for the fleet (and removing them) and the flexibility of the decision-making processes, associated with a highly competitive, price-sensitive and efficiency-dependent market, make this a relevant and interesting sector to study.

1.2. Brief problem description and previous works

This work deals with the integration of two of the main decisions that car rentals face: determining the *capacity* of their fleet – which includes decisions on acquisition modes and timings, as well as fleet deployment between locations, in order to meet demand – and determining the *price* of the variety of rentals that are requested.

* Corresponding author.
E-mail addresses: beatriz.oliveira@fe.up.pt (B.B. Oliveira), mac@fe.up.pt (M.A. Carravilla), jfo@fe.up.pt (J.F. Oliveira).

https://doi.org/10.1016/j.orp.2018.10.002
Received 4 October 2018; Accepted 4 October 2018
Available online 05 October 2018

Currently, these problems are mainly tackled separately, often within a sequential or hierarchical framework. Pachon et al. [4] propose the primary modeling framework for fleet planning in the car rental industry: a sequential and hierarchical structure of mathematical models and solution methods to solve in different steps the problems of *pool segmentation* (where the rental stations are clustered in fleet-sharing pools), *strategic fleet planning* (where the size of the fleet of each pool is decided), and *tactical fleet planning* (where the fleet levels in each station of the pool are decided – deployment – as well as the required vehicle transfers).

Recently, some works have been developed that attempt to integrate the most linked decisions, especially fleet size and fleet deployment [5,6], although often simplifying the problem and hindering the practical application of the developed research. However, in Fink and Reiners [7], fleet sizing is studied in detail, using a realistic modeling approach that included acquisitions and removals of vehicles as well as other issues, such as partial substitution between vehicle groups, in order to turn it applicable to real-world situations.

In car rental, the issues of fleet management often intertwine issues with operations, such as the ones discussed so far, with other problems usually tackled under the revenue management framework. In fact, due to the inherent flexibility of the fleet, this sector is often studied in revenue management, especially as far as capacity allocation is concerned. For example, Guerriero and Olivito [8] derive different acceptance policies for car rental booking requests while Steinhardt and Gönsch [9] integrate these approaches with operations issues related with planned upgrades. As for pricing, it is considered as an emerging tool used by practitioners to manage demand, since it is increasingly easy and cheap for companies to dynamically and swiftly change the prices through online booking channels [10,11]. Heterogeneity of customer preferences influence most rental businesses, inclusively on the antecedence of the requests and especially in what regards pricing and revenue management. In many businesses, rental customers are divided into two main groups: customers that require the service with some antecedence and "walk-in" customers, with different willingness to pay and different service expectations. In the car rental problem tackled in this paper, the antecedence of the rental requests is considered to have several discretized levels and may significantly influence the demand, alongside price.

Oliveira et al. [12] present a thorough literature review on fleet management and revenue management issues on car rental and propose a conceptual framework for the different levels of decision. One group of decisions deals with *pool segmentation*, as proposed by Pachon et al. [4]. Then, for each pool, five interconnected decision blocks are defined. Three of these blocks are related with operations fleet management problems such as *fleet size/mix*, which broadly decides how many vehicles of each type will compose the fleet (including decisions on acquisitions and removals), *fleet deployment*, which deals with the distribution of the vehicles among locations and how they are repositioned between them, and *fleet assignment*, which assigns specific vehicles to the existing rental requests. As for the problems usually tackled by revenue management, two main blocks are defined that represent the two perspectives of the field: *pricing*, where the price of each rental is defined, and *capacity allocation*, which decides which fixed-price rental requests should be fulfilled with the existing capacity. This paper can be positioned within this conceptual framework, since it integrates three main decision blocks for a single pool of locations: *flee size/mix, fleet deployment* and *pricing*.

As previously discussed, the integration between fleet size/mix and fleet deployment has been often considered in the car rental fleet management literature. The integration of these problems (herein commonly referred to as *capacity problems*) with pricing decisions is mentioned as a research direction with considerable potential [12]. Some interesting works aim to fulfill this gap, such as Haensel et al. [13] where fleet deployment is integrated with capacity allocation decisions by simultaneously deciding on booking limits and vehicle transfers for a homogeneous fleet. To the best of our knowledge, only one paper aims to integrate pricing with fleet management decisions (in the case, fleet deployment as well) for the car rental business. Madden and Russell [14] propose an interesting formulation where the price is decided based on the discrete choice of price levels and where the direct impact of price on demand is used to balance fleet levels.

The potential of this integration, which is starting to be explored in the literature, derives from the close connections between the two problems and the overlapping decision-making time horizons. In fact, pricing decisions influence and are influenced by the availability of the fleet, which is dependent on fleet occupation and on fleet size and location.

Also in other sectors, the relationship between pricing and capacity, the ability of price to manage demand, and the potential of their integration is being explored. Zhang and Zhang [15] investigate the role of congestion tolls in an airport as a demand management tool as well as a financing source, focusing on the impact of carriers with a significant market position transposing these costs for higher price tickets. Also, in Wang et al. [16], the problem of locating a park-and-ride facility is integrated with the pricing decision. The relationship between pricing and production decisions is thoroughly explored in Bajwa et al. [17]. In [18], pricing is integrated with assortment and inventory decisions for substitutable products in a retail environment. Some interesting insights are identified regarding the lack of structure of the solutions obtained, which reflects the potential of optimization approaches, and regarding the importance of this integration to significantly improve profitability. The impact of price-driven product substitution for a company selling to different customer segments, within a context of integrated pricing and production decisions, is further studied in Kim and Bell [19], with a significant effort on demand and substitution modelling.

The solution method proposed in this work is a matheuristic, since it hybridizes a metaheuristic with mathematical models. This approach decomposes the original mathematical model in terms of its decisions. The metaheuristic guides the search over the decisions on pricing strategy, while the remaining decisions are solved using mathematical models generated by fixing the pricing strategies on the original monolithic model. In fact, approaches that combine decomposition strategies with metaheuristics are currently being used to solve difficult combinatorial problems. The decomposition takes advantage of special structures of the problem enabling these approaches to outperform "less hybridized" methodologies. Raidl [20] proposes an interesting discussion on this topic, showing promising possibilities for these approaches. The combination of genetic algorithms with decomposition strategies to solve complex problems has been used with success, for example, by Paes et al. [21] to tackle the unequal area facility location problem.

The metaheuristic used to guide the decomposition is a biased random-key genetic algorithm (BRKGA) [22]. BRKGA is a variation of the random-key genetic algorithm (RKGA) where there is a bias on the choice of one of the parents towards one with a better fitness (instead of an entirely random selection). BRKGA has been used with success in several complex problems. Moreover, this type of methodology has the ability to encompass problem-specific knowledge and to use it to boost its performance. This is demonstrated, for example, in the work of Ramos et al. [23], where BRKGA is used to tackle the container loading problem and includes procedures that take into account static stability constraints derived from mechanical equilibrium conditions.

An important part of the solution method is the generation of initial solutions for the first population of BRKGA, which is conventionally entirely random. Other works have used this type of boost for BRKGA. When tackling the two-stage stochastic Steiner tree problem, Hokama et al. [24] use a constructive heuristic to generate the entire initial population of the algorithm. It is nonetheless more common to generate only a part of the initial population, thus ensuring that there is still randomness associated with it. For example, when tackling the three-dimensional bin packing problem with heterogeneous bins, Li et al.

[25] generate four solutions using a constructive heuristic. These solutions are added to the initial population, whose remainder individuals are randomly generated. Furthermore, even one heuristically generated solution added to the initial population can have significant impact. In Stefanello et al. [26], a genetic algorithm is proposed to solve the problem of pricing network of roads, i.e. defining tolls to be applied in some arcs of the network. One solution is generated by relaxing integrality constraints and is added to the initial population, thus boosting the overall performance.

1.3. Contributions

The car rental business is characterized and seizes its natural advantage of being able to decide on capacity levels with significant flexibility. Nevertheless, other characteristic that fully differentiates this sector from other sectors mentioned above (such as retail) is the *rental*-type of transaction considered. In this context, capacity is not only affected by initial or frequent capacity/inventory decisions but also by “returning” vehicles, which are temporarily used but become available again in the future, possibly at a different location. This impacts significantly the structure of the problem.

Traditionally, car rental companies tend to separate these problems deciding on fleet size first and then managing the demand through pricing decisions that accelerate or decelerate occupation and deploying the fleet to meet demand. This work points out to the fact that integrating these decisions will allow for significant improvements due to the flexibility gained by also using fleet size as a tool to manage demand. The main disadvantage of the integration – the computational burden – is tackled by the use of an innovative solution procedure.

This work has thus three main contributions:

- A new mathematical model for the integration of capacity (including decisions on acquisitions, fleet size and mix, and deployment) and pricing decisions for car rentals.
- An innovative and high-performing solution method for the problem that is able to obtain good solutions for real-sized instances within realistic time frames.This method is based on a decomposition of the mathematical model. A genetic algorithm is used to guide the search over part of the decision variables. The value (fitness) of these *partial solutions* is evaluated by fixing them in the original model and solving it for the remaining decisions. Moreover, a structured and robust way of heuristically generating initial solutions for the genetic algorithm is proposed, showing a significant power to boost the search.
- A quantitative proof that the integration of these problems brings measurable improvements for companies, when compared with a sequential approach.

The solution method proposed was conceived according to a modular and quick-to-answer design, so that it can be easily implemented in a decision support system to help car rental companies make more profitable decisions. Nevertheless, the work developed in this paper already brings relevant managerial insights, especially regarding the potential of integrating pricing and capacity decisions for a selling season. Based on this research, a company is able to ascertain whether an integrated approach brings advantages over a sequential approach, mainly based on market size and number and type of products to price.

Moreover, a parallel can be built between the car rental business and *car sharing* systems, namely in what concerns the mobility/flexibility of the fleet and decision-making processes and the role of pricing in managing demand. Therefore, this model and solution procedure can be extended to be applied in this increasingly relevant urban mobility topic.

1.4. Paper structure

This paper is structured as follows. Firstly, the capacity-pricing problem for car rentals and the novel mathematical model will be presented (Section 2) and the proposed solution method will be explained (Section 3). Then, in Section 4, the computational tests and their results are discussed and finally, in Section 5, some conclusions and future research directions are drawn.

2. Problem definition

The work here presented was inspired by the case of a Portuguese car rental company. In this section, the problem will be introduced by providing an overall scenario of this company’s business and an overview of the scope of the problem at hand. Then, the Capacity-Pricing Model will be fully defined by its mathematical formulation as an Integer Non-Linear Programming Model (INLP).

2.1. Problem statement: The case of a Portuguese car rental company

This work aims to support the decisions of a car rental company that is planning a selling season (1–3 months) and must decide on the acquisition and fleet capacity plan, which are interconnected with the overall pricing strategy. Indeed, in this kind of business, when demand exists, very roughly companies can increase their profit either by maintaining the prices and increasing the fleet, or raising the prices and keeping the fleet size as it is.

The car rental company that inspired this work is based in Portugal, where it has approximately 40 rental stations, divided into four regions. All regions share the same fleet. The company uses these regions as “units of location” when tackling the tactical/strategic problems that will be detailed in this section. This is due to the fact that moving vehicles between stations within the same region is negligible in terms of both cost and time, unlike inter-region movements.

The fleet of the company is composed of approximately 10,000 vehicles and is divided in up to 5 vehicle groups, depending on the selling season. Besides being differentiated by group, the fleet is also divided in owned and leased fleet. The purchase of owned vehicles is planned with a certain advance. Usually, these vehicles are available in the beginning of the selling season and are sold within one year. Leased vehicles, however, are used to face peaks in demand and can be available for shorter and more flexible periods of time, with a higher cost for the company.

This fleet is used to serve the different types of rentals requested, which are characterized by start and end locations and dates, as well as required vehicle group. Depending on the selling season, this company can deal with 450–2500 different rental types, priced individually and differentiated according with the antecedence of the request. For each rental type, the number of requests the demand, which may later occupy the acquired vehicles, is highly dependent on pricing.

A common practice to help meet demand in different locations throughout the season is to perform “empty transfers”. An “empty transfer” occurs when a vehicle is moved from one location to another not as part of a rental but to meet demand, with a non-negligible cost and travel time. The company performs these transfers by either truck or using a driver.

Other practice used by this and other car rental companies to meet demand is to offer upgrades when the requested group is not available. This means that the companies offer a more-valued vehicle than what was requested for the same price. This allows them to maximize the utilization of the fleet and to meet demand. However, regular upgrades are commonly avoided as they incentivize the strategic consumer behavior of renting vehicles that do not meet expectations in hope of begin offered an upgrade. A proper fleet planning can help provide the needed vehicles where they are needed so that upgrades are only used as a last resource.

Due to maintenance costs, there is a high emphasis on the company's ability to maximize the occupation of vehicles. Currently, the company follows a hierarchical approach: first, it decides the capacity and afterwards makes the pricing decisions with the goal to maximize the fleet occupation. The main objective of this work is increase the company's profitability by integrating the capacity with the pricing decisions, since the latter have a strong impact on demand and, consequently, can help make better capacity decisions.

2.2. Mathematical model

In order to fully describe the problem presented above, a mathematical model was developed. The main decision variables are related with the acquisition of vehicles and with the prices of different rental types. The number of rental requests is highly dependent on the pricing decisions. These requests may later become *fulfilled rentals* and occupy the acquired vehicles. In order to fully understand and map this interaction between capacity and pricing, other decisions are considered, such as the stock of vehicles in each location and time period and the number of vehicles "empty transferred" between locations.

As mentioned above, the fleet of vehicles, besides being differentiated by group, is also divided in owned and leased fleet. It is assumed that the total number of vehicles purchased for the owned fleet are available at the beginning of the time horizon. The leased vehicles may become available for shorter periods throughout the time horizon.

The objective is to maximize the company's profit in the time horizon. The profit is the difference between the revenues obtained with the rentals fulfilled and the costs of leasing/acquiring fleet, performing the empty transfers and maintaining the owned fleet, as well as a penalization factor for upgrades.

This model has four main groups of constraints, which will be further explained in Section 2.2.4:

- Stock calculating constraints, where the stock of vehicles of each group in each time period and station is computed;
- Capacity/Demand constraints, where these are established as a limitation on the number of rentals fulfilled and empty transfers realized;
- Business-related constraints, where the limitations regarding possible upgrades and available purchase budget are established;
- Other auxiliary constraints.

The goal of introducing pricing decisions, and corresponding changeable demand levels, in the capacity planning is not to produce operational decisions that are updated on an online manner and swiftly react to changes in the system. This objective, although very important for car rental companies, is considered to be out of scope for this study. Within the tactical/strategic scope herein considered, pricing decisions and demand information are used in an offline manner to provide better quality to a model that aims to produce season-lasting decisions, such as fleet size and mix. In a real-world application, applying this model to support such decisions would not be conflicting with using a more operational model where requests appear in an online fashion and decisions like performing empty transfers or offering upgrades are revised and dynamically optimized.

2.2.1. Indices and parameters

$t, t' = \{0, \ldots, T\}$	Index for the set $\mathcal{T}$ of time periods, where $t = 0$ represents the initial conditions of the time horizon (season) and "overlaps" with $t = T$ for the previous season
$g, g1, g2 = \{1, \ldots, G\}$	Index for the set $\mathcal{G}$ of vehicle groups
$s, s1, s2, c = \{1, \ldots, S\}$	Indices for the set $\mathcal{S}$ of rental locations
$r = \{1, \ldots, R\}$	Index for the set $\mathcal{R}$ of rental types (characterized by check-out and check-in location and time period, and vehicle group requested)
$sout_r$	Check-out location of rental type r
sin_r	Check-in location of rental type r
$dout_r$	Check-out time period of rental type r
din_r	Check-in time period of rental type r
gr_r	Vehicle group requested by rental type r
$a = \{0, \ldots, A\}$	Index for the set $\mathcal{A}$ of antecedences allowed (number of time periods between the rental request and the start of the rental), where $a = 0$ represents a "walk-in" customer
$p = \{1, \ldots, P\}$	Index for the set $\mathcal{P}$ of price levels allowed
PRI_{pg}	Pecuniary value associated with price level p for vehicle group g (for example, for group $g = 2$, price level $p = 1$ has a pecuniary value of $PRI_{1,2} = 20€$)
DEM_{rap}	Demand for rental type r, at price level p, with antecedence a
COS_g	Buy cost of a vehicle of group g. The value considered is the net cost: purchase gross cost minus salvage value derived from its sale after one year (see Section 2.1)
LEA_g	Leasing cost (per time unit) of a vehicle of group g
OWN_g	Ownership cost (per time unit) of a vehicle of group g
LP_g	Leasing period for a vehicle of group g
PYU	Penalty charged for each upgrade
UPG_{g1g2}	Whether a vehicle of group $g1$ can be upgraded to a vehicle of group $g2$ (= 1) or not (= 0)
TT_{s1s2}	Transfer time from location $s1$ to location $s2$
TC_{gs1s2}	Transfer cost of a vehicle of group g from location $s1$ to location $s2$
BUD	Total budget for the purchase of vehicles
M	Big-M large enough coefficient

Other sets:

$\mathcal{R}^{g^-}$	Rental types that do not require group g
$\mathcal{R}^{in}_{st}$	Rental types whose check-in is at location s at time $\in [t-1, t[$
$\mathcal{R}^{out}_{st}$	Rental types whose check-out is at location s at time $\in [t-1, t[$
$\mathcal{R}^{use}_{t}$	Rental types that require a vehicle to be in use at t (i.e., $dout < t \wedge din \geq t$)

Inputs from previous seasons (previous decision periods):

INX^O_{gs}	Initial number of owned (O) vehicles of group g located at s, at the beginning of the season ($t = 0$)
$ONY^{L/O}_{gts}$	Number of owned (O) or leased (L) vehicles of group g on on-going empty transportation (previously decided), being transferred to location s, arriving at time t
$ONU^{L/O}_{gts}$	Number of owned (O) or leased (L) vehicles of group g on on-going rentals (previously decided), being returned to location s at time t

2.2.2. Decision variables

w^O_{gs}	Number of vehicles of group g acquired for the owned fleet available at $t = 0$ in location s
w^L_{gts}	

Number of vehicles of group g acquired by leasing to be available at time t in location s

q_{rap} = 1 if price level p is charged for rental type r with antecedence a; = 0 otherwise

$x_{gts}^{L/O}$ Number (stock) of leased (L) or owned (O) vehicles of group g located at s at time t

$y_{s1s2gt}^{L/O}$ Number of leased (L) or owned (O) vehicles of group g empty transferred at time t from location $s1$ to location $s2$

$u_{rag}^{L/O}$ Number of fulfilled rentals requested as rental type r with antecedence a that are served by a leased (L) or owned (O) vehicle of group g

$f_{gt}^{L/O}$ Auxiliary variable: total leased (L) or owned (O) fleet of group g at time t

2.2.3. Objective function

Eq. (1) represents the objective function of the model, which aims to maximize the profit of the company, comprising the activities of renting, purchasing and leasing vehicles and managing the fleet. The first element of the objective function represents the revenue earned from the fulfilled rentals, which is given by the price charged (dependent on the group requested) times the number of rentals served using leased and owned fleet. This term of the objective function renders the model non-linear, since two decision variables are multiplied.

The second term represents the cost of purchasing the owned fleet – a one-time cost. The following terms are related with the costs of leasing the vehicles (recurrent throughout the leasing period) and the ownership costs (also recurrent). The latter are significantly smaller than the former and aim to represent the regular costs of maintaining the owned fleet. Then, the empty transfer costs are represented, which depend on the group of each vehicle transferred and the origin-destination pair. Finally, an artificial marginal cost to offer upgrades is included in order to ensure that this practice only exists if there are no available cars from the required group.

$$\begin{aligned}&\max \ \text{Profit from fulfilled rentals} - \text{Buying cost} - \text{Leasing cost} \\ &-\text{Ownership cost} \\ &- \text{Empty transfer cost} - \text{Penalty for upgrading} = \\ &\max \sum_{r=1}^{R}\sum_{a=1}^{A}\left(\left(\sum_{g=1}^{G} u_{rag}^{L} + u_{rag}^{O}\right)\sum_{p=1}^{P} q_{rap} PRI_{p,gr_r}\right) - \sum_{g=1}^{G}\left(\sum_{s=1}^{S} w_{gs}^{O}\right) COS_g \\ &- \sum_{g=1}^{G}\left(\sum_{t=1}^{T} f_{gt}^{L}\right) LEA_g - \sum_{g=1}^{G}\left(\sum_{t=1}^{T} f_{gt}^{O}\right) OWN_g \\ &- \sum_{s1=1}^{S}\sum_{s2=1}^{S}\sum_{g=1}^{G}\left(\sum_{t=1}^{T}\left(y_{s1s2gt}^{L} + y_{s1s2gt}^{O}\right)\right) TC_{gs1s2} - \sum_{g=1}^{G}\sum_{r\in\mathcal{R}^{g^-}}\sum_{a=1}^{A}\left(u_{rag}^{L} + u_{rag}^{O}\right) PYU\end{aligned} \quad (1)$$

2.2.4. Constraints

Stock calculating constraints. Eqs. (2)–(6) represent the calculation of the "stock" of available vehicles of a certain group, in a specific location, at a specific time. These constraints also link the problem for the different time periods and locations.

Eq. (2) aim to characterize the stock of owned vehicles of each group, in each location, for each time period except the initial one. The stock is equal to the one of the previous period, increased by expected arrivals from rentals and empty transfers that started on previous seasons (parameters) and by the arrival of vehicles that were being employed in rentals that started this season and were meanwhile returned to this specific location, decreased by the vehicles that were meanwhile occupied by rentals that started in this location, increased also by the vehicles that were being empty-transferred from other locations and have meanwhile arrived, and finally decreased by the vehicles that were transferred to other locations.

$$\begin{aligned}\mathbf{s.\,t}\, x_{gts}^{O} &= x_{g,t-1,s}^{O} + ONY_{gts}^{O} + ONU_{gts}^{O} \\ &+ \sum_{r\in\mathcal{R}_{s,t}^{in}}\sum_{a=1}^{A} u_{r,a,g}^{O} - \sum_{r\in\mathcal{R}_{s,t}^{out}}\sum_{a=1}^{A} u_{r,a,g}^{O} \\ &+ \sum_{c=1}^{S} y_{c,s,g,t-TT_{cs}-1}^{O} - \sum_{c=1}^{S} y_{s,c,g,t-1}^{O} \qquad \forall\, g,\, t>0,\, s\end{aligned} \quad (2)$$

Eqs. (3) and (4) represent a similar situation yet applied to the leased fleet. One of the main differences of this type of fleet is that acquisitions may occur throughout the season. Therefore, a similar structure can be seen when confronting with Eq. (2), but with the addition of some terms related with the acquisition of leased vehicles. In Eqs. (3) and (4), the stock is increased with the corresponding leasing acquisitions. Then, since leased vehicles must be removed from the fleet after the leasing period (LP) is over, Eq. (4), valid for all time periods greater than the leasing period, also decrease the stock by the number of returned leased vehicles.

$$\begin{aligned}x_{gts}^{L} &= x_{g,t-1,s}^{L} + ONY_{gts}^{L} + ONU_{gts}^{L} \\ &+ \sum_{r\in\mathcal{R}_{s,t}^{in}}\sum_{a=1}^{A} u_{r,a,g}^{L} - \sum_{r\in\mathcal{R}_{s,t}^{out}}\sum_{a=1}^{A} u_{r,a,g}^{L} \\ &+ \sum_{c=1}^{S} y_{c,s,g,t-TT_{cs}-1}^{L} - \sum_{c=1}^{S} y_{s,c,g,t-1}^{L} \\ &+ w_{g,t-1,s}^{L} \qquad \forall\, g,\, 0<t\le LP_g,\, s\end{aligned} \quad (3)$$

$$\begin{aligned}x_{gts}^{L} &= x_{g,t-1,s}^{L} + ONY_{gts}^{L} + ONU_{gts}^{L} \\ &+ \sum_{r\in\mathcal{R}_{s,t}^{in}}\sum_{a=1}^{A} u_{r,a,g}^{L} - \sum_{r\in\mathcal{R}_{s,t}^{out}}\sum_{a=1}^{A} u_{r,a,g}^{L} \\ &+ \sum_{c=1}^{S} y_{c,s,g,t-TT_{cs}-1}^{L} - \sum_{c=1}^{S} y_{s,c,g,t-1}^{L} \\ &+ w_{g,t-1,s}^{L} - w_{g,t-LP_g-1,s}^{L} \qquad \forall\, g,\, t>LP_g,\, s\end{aligned} \quad (4)$$

Eqs. (5) and (6) calculate this stock for the beginning of the season ($t = 0$). As for the owned fleet, Eq. (5), the initial stock will be equal to the stock existent in the previous season (parameter) and the number of purchased vehicles. The leased fleet, Eq. (6), is considered to be initially null.

$$x_{g0s}^{O} = INX_{gs}^{O} + w_{gs}^{O} \qquad \forall\, g,\, s \quad (5)$$

$$x_{g0s}^{L} = 0 \qquad \forall\, g,\, s \quad (6)$$

Capacity / Demand constraints:. At a given location and time period, the number of rentals fulfilled and the empty transfers that start at that location and time are limited by the stock of available cars - Eq. (7). Eq. (8) ensure that the number of rentals fulfilled is also limited by the demand for the rental type, at the chosen price level.

$$\sum_{r\in\mathcal{R}_{st}^{out}}\sum_{a=1}^{A} u_{rag}^{L/O} + \sum_{c=1}^{S} y_{scgt}^{L/O} \le x_{gts}^{L/O} \qquad \forall\, g,\, t,\, s \quad (7)$$

$$\sum_{g=1}^{G}\left(u_{rag}^{L} + u_{rag}^{O}\right) \le DEM_{rap} + M(1 - q_{rap}) \qquad \forall\, r,\, a,\, p \quad (8)$$

Business-related constraints:. The upgrading policies (i.e., which groups can be upgraded to which groups) are translated into Eq. (9).

$$\sum_{a=1}^{A}\left(u_{rag}^{L} + u_{rag}^{O}\right) \le UPG_{gr_r,g} \times M \qquad \forall\, r,\, g \quad (9)$$

Also, the number of purchased vehicles in each time period is limited by the total available budget - Eq. (10).

$$\sum_{s=1}^{S}\sum_{g=1}^{G} w_{gs}^{O} COS_g \leq BUD \tag{10}$$

Other constraints:. Eq. (11)ensure that only one price level is chosen per rental type and antecedence.

$$\sum_{p=1}^{P} q_{rap} = 1 \qquad \forall\, r,\, a \tag{11}$$

In order to facilitate the construction of the objective function, an auxiliary decision variable was created that represents the totality of the leased (L) and owned (O) fleet of a certain group in each time period. Eq. (12) define it as the sum of the stock of vehicles at the rental locations, the vehicles that are currently being used in rentals and the cars currently being transferred between locations.

$$f_{gt}^{L/O} = \sum_{s=1}^{S} x_{gts}^{L/O} + \sum_{r\in\mathcal{R}_t^{use}}\sum_{a=1}^{A} u_{r,a,g}^{L/O} + \sum_{s1=1}^{S}\sum_{s2=1}^{S}\sum_{t'=\max\{0,t-TT_{s1s2}\}}^{t-1} y_{s1,s2,g,t'}^{L/O} \qquad \forall\, g,\, t \tag{12}$$

Finally, Eq. (13) represent the domain of the decision variables. Except for the binary variable that selects the price level to be charged, all variables are integer and non-negative.

$$\begin{aligned} q_{rap} &\in \{0, 1\} && \forall\, r, a, p \\ w_{gts}^{L} &\in \mathbb{Z}_0^+ && \forall\, g, t, s \\ w_{gs}^{O} &\in \mathbb{Z}_0^+ && \forall\, g, s \\ y_{s1s2gt}^{L/O} &\in \mathbb{Z}_0^+ && \forall\, s1, s2, g, t \\ x_{gts}^{L/O} &\in \mathbb{Z}_0^+ && \forall\, g, t, s \\ u_{rag}^{L/O} &\in \mathbb{Z}_0^+ && \forall\, r, a, g \\ f_{gt}^{L/O} &\in \mathbb{Z}_0^+ && \forall\, g, t \end{aligned} \tag{13}$$

A brief discussion on some insights regarding model structure are presented on Appendix A.1. This discussion is based on an analogy between the formulation proposed and the transportation problem model and helps further understand the inherent structure of this problem.

3. Proposed solution method

Since the Capacity-Pricing Model is significantly complex and hard to solve for real-sized instances, inclusively due to the non-linearity of the objective function, a solution method was proposed to obtain good quality solutions within a reasonable time-frame.

The overall idea of the method is based on the decomposition of the original model in *pricing decisions* and the *remaining decisions*, exploring the structure of the mathematical model. A metaheuristic – in this case, a genetic algorithm – is used to search for good pricing strategies. Here, for simplicity, the term *pricing strategy* will be used to represent a set of feasible values for the pricing decision variables $q_{rap}, \forall r,\ a,\ p$ (see Section 2.2.2). To assess how good a pricing strategy is, the values corresponding to the pricing decisions are fixed and the mathematical model is solved for the remaining variables. The resulting objective value quantifies the profit that can be obtained with the pricing strategy.

Fig. 1 shows the overview of the proposed solution method. Biased Random-Key Genetic Algorithm (BRKGA) is the metaheuristic used to search good pricing decisions and is represented by the central diamond shape. In this section, the BRKGA framework will be detailed, including the structure of the chromosomes and population. Fitness calculation is the process within the genetic algorithm that assesses how good each pricing strategy is and, as mentioned above, comprehends solving the mathematical model with the pricing decisions fixed. In fact if the prices are fixed inputs and not decision variables, the problem becomes an integer linear problem and hence easier to solve. Such problems are special cases of Mixed Integer Programs (MIP) and due to ubiquity of this acronym, it will be used to represent this problem. In order to speed up the process, the linear program (LP) that results from relaxing the integrality constraints of this MIP is considered as a substitute approximation for the fitness evaluation.

From this search procedure, the best pricing strategy is retrieved and the final value to the remaining variables is calculated by fixing the pricing strategy and solving to optimality the resulting MIP model (bottom rectangle in Fig. 1).

BRKGA's "generation zero" is conventionally entirely random. In this solution method, specific knowledge about this problem, such as the natural decomposition scheme that arises from forbidding upgrades, was used to improve the performance of the BRKGA's search. Initial heuristic pricing strategies were generated and fed into the "generation zero". These solutions are achieved by decomposition, relaxation and construction, and are represented by the rectangles on the left that are shown as inputs to the BRKGA procedure. This will also be detailed in this section, with a full discussion on the modeling choices made, which are also represented in the bottom of Fig. 1.

3.1. BRKGA framework

A Biased Random-Key Genetic Algorithm (BRKGA) was used to guide the search over different pricing strategies. In genetic algorithms, a solution is considered as an *individual* belonging to a *population* and encoded in a *chromosome*. The objective function value of the solution is translated into the chromosome's fitness. A population, composed of a set of individuals, is evolved over some generations. Each generation involves the creation of a new population through the combination of pairs of individuals of the previous generation (the parents), as well as random mutation. The fitness value is herein critical for the selection of elements to combine and produce the following generation. Genetic algorithms with random-keys use random real numbers between 0 and 1 as genes. A deterministic procedure, the *decoder*, translates each chromosome into a solution of the original problem and evaluates it in terms of its fitness [22].

In this case, the solutions that compose a population and that were translated into chromosomes are the *pricing strategies*. The value of each pricing strategy would be the result of solving the MIP model with the price as a fixed input. However, in order to accelerate the procedure, an approximation was used to evaluate the fitness of each chromosome: the linear program (LP) resulting from relaxing all integrality constraints. To obtain the final solution, the MIP model is run with the integrality constraints considering as price input the best pricing strategy found by the BRKGA.

3.1.1. General idea and motivation

The general idea of the proposed solution method is to use BRKGA to generate and evolve pricing strategies. Each pricing strategy is evaluated in terms of the optimum outcome for all the decisions, by solving the Capacity-Pricing Model to optimality with prices as inputs. This allows to decompose the main problem in easier sub-problems. At the same time, the fact that this decomposition and the search within the consequent sub-problems are guided by a metaheuristic gives the solution method, at least theoretically, a certain validity and consistency. Moreover, by using a population-based method, it is expected that local optima will be avoided.

3.1.2. Chromosome structure

A chromosome represents a pricing strategy, i.e., the price levels chosen for each rental type, requested with a certain antecedence. A chromosome is a vector of genes, which can take on a value – an allele – between 0 a 1. In this structure, each gene in the chromosome relates to

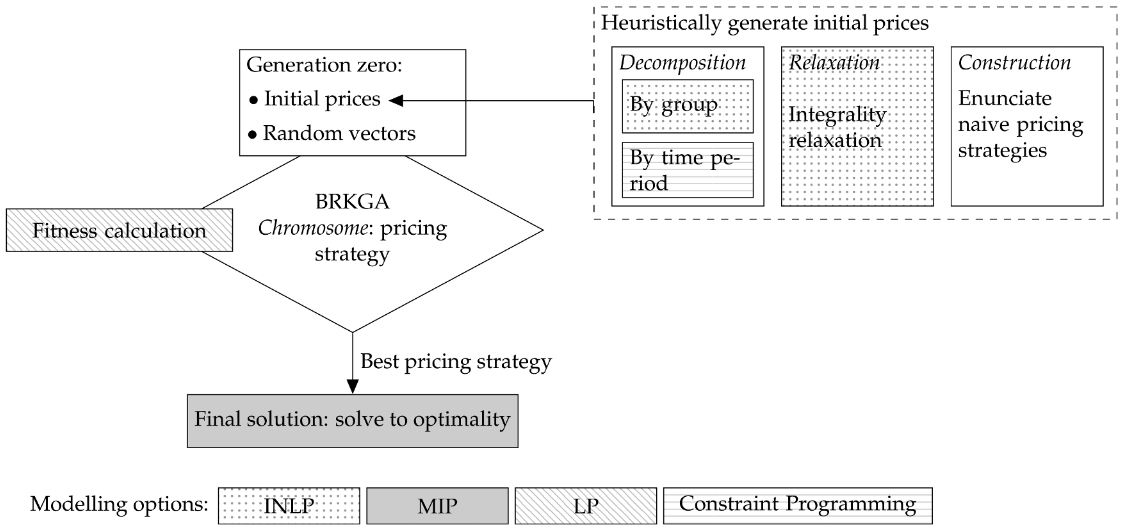

Fig. 1. Overview of the proposed solution method.

the combination of a rental type with an antecedence level, therefore each chromosome has $|\mathcal{R}| \times |\mathcal{A}|$ genes, where $|\mathcal{R}|$ is the number of rental types and $|\mathcal{A}|$ the number of antecedence levels. The allele of the gene, i.e. the random number (n) associated with it, is then compared with the threshold that comes from dividing the range [0,1] in $|\mathcal{P}|$ equal partitions, where $|\mathcal{P}|$ is the number of possible price levels allowed:

$$\text{price level} = \left\lfloor \frac{n}{1/|\mathcal{P}|} \right\rfloor + 1 \tag{14}$$

Fig. 2 illustrates this translation process for a simple example.

3.1.3. Fitness evaluation

In order to understand the value of each pricing strategy, the fitness of the chromosome is evaluated. As mentioned above, the objective is to understand what the optimum result of using each pricing strategy is, considering the impact it has in all other decisions. To achieve this, one should solve the MIP that results from fixing on the Capacity-Pricing Model the pricing strategy given by the chromosome. Preliminary tests showed that, although the MIP model is fairly quick to solve, the solution times (around a few minutes) were not adequate when considering a population of considerable size that should evolve for some generations within a reasonable time frame. Therefore, to significantly speed up the process, the linear relaxation of the MIP (LP) was used as an approximation.

For this approximation to be valid, it is important to guarantee that not only the LP obtains an objective value similar to the MIP but also that the fitness ranking by which the chromosomes are sorted in a population is similar. In fact, in BRKGA, the evolution of a population consists, on a simplified view, in three main steps: (1) the best elements of the population (the elite) are directly copied to the next generation, (2) new chromosomes are generated from the cross-over of two elements of the current generation (elite or not), and (3) new chromosomes are randomly generated and inserted (mutant chromosomes). The fitness is used to sort the elements of a population so that the top (elite) and bottom elements are identified and steps (1) and (2) take place.

Therefore, to validate this approximation, 100 chromosomes were randomly generated and evaluated using the MIP model and its LP relaxation, based on Instance 1 (see Section 4.1). As expected, the LP was always solved in a few seconds, while the MIP was given a time limit of 2 min and could prove optimality in approximately one third of the cases. Fig. 3 shows the boxplot for each situation. In fact, the values of the objective function were very similar, even in the cases where the MIP could not prove optimality. As expected, the LP approximation obtained results more similar to the MIP when the latter was able to prove optimality. However, in both cases, the differences are very small (always less than 0.14%).

As for the order by which these 100 chromosomes are sorted, there are some differences when using the objective function value of the LP or of the MIP. Nevertheless, these differences do not appear to be significant. Fig. 4 shows this by plotting the chromosomes in the order sorted by *LP approximation* against the *MIP objective function value*. With this, it is possible to conclude that, although the ranking order is not exactly the same using the two approaches (if it were the graph would show a monotonically decreasing plot), where there are differences in ranking position there are no major differences in the objective function value. For example, the main difference is between positions 33 and 35 and here the difference of the MIP objective function value between the three chromosomes in these positions is only of 0.03%. Therefore, solving the LP relaxation provides a valid approximation for the MIP objective value in the context of this solution method.

3.2. Generation zero: Heuristically generated initial pricing strategies

In order to boost the performance of the BRKGA, some specific pricing strategies were added to the (conventionally entirely random) initial generation, or “generation zero”. The goal was to use specific knowledge of the problem to provide solutions that could have a good performance and could otherwise be missed. This specific knowledge is especially related with practical-driven simplifications or relaxations of the original problem. For example, if upgrades are not allowed, the problem becomes separable by vehicle group and hence easier to solve and the resulting pricing strategy may show significant potential to improve and evolve in this framework.

	r_1			r_2			r_3			r_4		
	a_1	a_2	a_3	a_1	a_2	a_3	a_1	a_2	a_3	a_1	a_2	a_3
chromosome:	0.86	0.22	0.74	0.15	0.78	0.19	0.52	0.66	0.98	0.41	0.12	0.05
price level 1 (> 0.50)	X		X		X		X	X	X			
price level 2 (< 0.50)		X		X		X				X	X	X

Fig. 2. Chromosome structure and translation into a pricing strategy. Example of choice among 2 possible price levels for 4 rental types (r) and 3 antecedences (a).

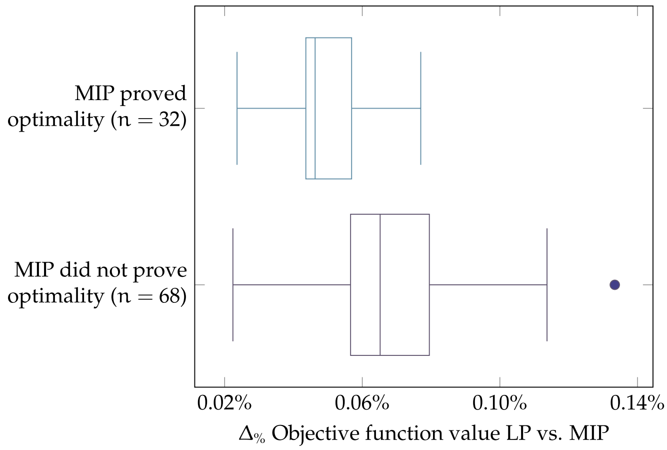

Fig. 3. Box plot for the percent variation between objective value of LP relaxation vs. MIP for 100 random chromosomes.

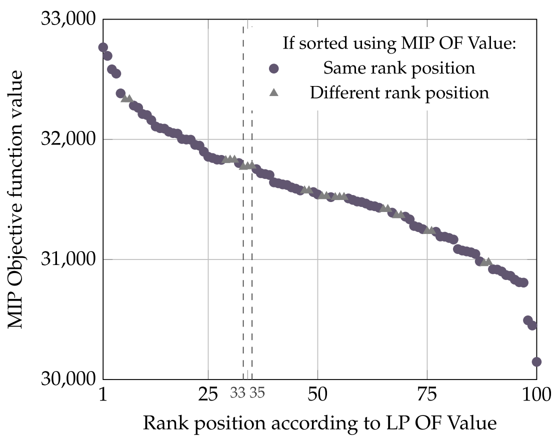

Fig. 4. MIP objective function (OF) value of 100 random chromosomes sorted using the LP approximation.

In this work, the addition of initial pricing strategies was structured according to their sources. The initial prices were thus obtained by three types of procedures:

- *Decomposition* of the main problem in separable sub-problems;
- *Relaxation* of integrality constraints;
- *Constructions* of naive strategies.

Decomposition:. One of the "natural decompositions" of the Capacity-Pricing Model was previously mentioned and consists in solving the problem for each vehicle group individually. Although the resulting sub-problems are still INLPs, they are smaller and easier to solve, and provide significant information in a practical context. Another decomposition approach often used in multi-period problems is to separate the problem by time period. In this case, it corresponds to solving the problem with a "myopic" perspective, considering one week at a time (if the week is used as time unit) and using the decisions of the previous week as inputs of the following one. Two approaches were used, with different "myopia degrees" that were materialized in how the leasing costs were accounted for. In the most myopic approach, only the leasing costs for that specific week were considered whether in the other approach if a leasing was decided in that week the leasing costs for the entire leasing period were imputed to the decision week. Other "myopic" aspect of both these approaches is that purchases for the owned fleet are only considered on the first week. In conclusion, three initial pricing strategies are generated by decomposition: one by group decomposition and two by time period decomposition.

Relaxation:. The initial pricing strategies generated by this method do not necessarily arise from specific knowledge about the problem, but from the behavior of the Capacity-Pricing Model. Some preliminary experiments were conducted in order to understand if relaxing the integrality constraints of specific (integer) decisions would have a significant impact on both solving speed and solution quality. From these experiments, four different relaxation approaches were selected. The first consisted in relaxing the integrality of all decision variables, except the binary price selecting variables. The remaining three consisted on relaxing the integrality of all decision variables, except the binary price selecting variables and one of the three main decisions: acquisitions (w), stock (x) and rentals fulfilled (u). Each of these four approaches are still based on non-linear models, yet are easier to solve than the original one.

Construction:. These initial pricing strategies are generated not based on the Capacity-Pricing Model but on construction heuristics and aim to represent the naive or obvious solutions that could otherwise be missed. It is not expected that these strategies allow for a significant improvement boost, yet, since the processing time of enunciating these strategies is negligible, it is worth considering them, as they are often the strategies "at hand" to be used by companies. There are two ways of constructing naive strategies: one is to price every rental type requested at every antecedence level with the same price ($|\mathcal{P}|$ pricing strategies are thus generated, where $|\mathcal{P}|$ is the number of possible price levels), and the other is to apply always increasing or decreasing price levels to a rental type, depending on the antecedence (which leads to two other naive pricing strategies).

Modeling options

The *decomposition* and *relaxation* methods to obtain initial pricing solutions are based on non-linear models. Although these models are simpler and easier than the original Capacity-Pricing Model, preliminary experiments revealed difficulties in tackling some of the bigger real-sized instances (see Section 4.1). To face this, it was necessary to define and compare basilar modeling options.

Table 1 compares four types of models that can be used to generate initial prices: multi- and single-period (M)INLP and Constraint Programming (CP) models. Single-period models are introduced to tackle the *decomposition by time period*. The *construction* of naive strategies is not represented in the table since it consists in enunciating pricing strategies and not in solving mathematical models.

The most immediate option would be to use the Capacity-Pricing Model (INLP multi-period) and a single-period corresponding version. The multi-period model suits the *decomposition by group*, with the addition of a group of constraints to ensure that no upgrades are allowed, and the four types of integrality *relaxation* considered. In both cases, preliminary tests showed that there is a practical limitation on the size of the instance tackled, especially due to the compile-time of the non-linear segment of the objective function. However, this time can be considered a "fixed cost", since the objective function is common to these five initial prices to be generated (one by *group decomposition* and

Table 1
Modelling options comparison.

	Decomposition		Relaxation
	By group	By time period	
(M)INLP multi-period	✓[a]	–	✓[a]
(M)INLP single-period	–	✓[a]	✓[a]
CP multi-period	✓[a]	–	✗
CP single-period	–	✓	✗

[a] Subject to size limitations.

four by integrality *relaxation*) and thus only has to be compiled once.

The single-period model could be used to generate the two initial pricing strategies based on *time-period decomposition*, with similar instance size limitations. Nevertheless, since the objective function is different, a new "fixed cost" should be considered. By definition, it could also be used while *relaxing* the integrality of different decision variables.

Summarizing, each of the first two lines of Table 1 encompass a fixed resolution time to compile the non-linear model (which leads to limitations in the instance size) and a variable time to solve per initial price.

As the instance size limitations can hinder the generation of *decomposition* and *relaxation* initial pricing strategies for bigger instances and are mainly caused by the non-linearity of the objective function, a different modeling (and, consequently, solving) approach was considered: the adaptation of the multi-period and the single-period (M)INLP models to a multi-period and a single-period Constraint Programming (CP) models. Constraint Programming was considered due to its ability to deal with non-linearity issues, which were consuming the most time in the previously considered models.

First and foremost, Constraint Programming is a "paradigm for solving combinatorial search problems" [27] and a modeling approach suitable for integer decisions. The basic idea of CP is that variables have finite integer domains, related by a set of constraints that must be satisfied and that define the finite solution space. Therefore, it cannot be used to tackle the generation by *relaxation*. Preliminary tests showed that there were practical limitations on the instance size for the multi-period CP model for *group decomposition*. Moreover, these limitations were more significant than the ones found for the INLP model (i.e., some instances that could be tackled by the INLP could not be tackled by the CP multi-period model). However, for the single-period CP model, no significant size limitations were found.

Concluding, for the *decomposition by group* and *relaxation* the original Capacity-Pricing Model with additional constraints was selected as preferred modeling approach, while the CP single-period model was used to heuristically generate initial prices based on *time decomposition*. The CP single-period model is presented as an Appendix (Section A.2) and was developed with two alternative objective functions, depending on the degree of myopia, as previously discussed.

In order to ensure a reasonable run time, a practical limit was set to establish for which instances *relaxation* and *group decomposition* initial prices could be generated. Nonetheless, few instances are expected to surpass this limit. The limit was calculated based on computational tests that showed that there is an exponential relationship between the size of the instances (measured by the number of rental types $|\mathcal{R}|$ times the number of vehicle groups $|\mathcal{G}|$) and the time to generate *relaxation* and *group decomposition* pricing strategies (see Fig. 5). In order to keep run time for these procedures under the limit of 100 min, for instances with $|\mathcal{R}| \times |\mathcal{G}| > 8,250$ it will not be advantageous to generate prices by *relaxation* and *group decomposition*. As observed in Fig. 5, only the two biggest instances, which are considerably bigger than the remaining ones, would be included in this set. The impact of the generation of these initial pricing strategies on the overall run time will be further discussed on Section 4.

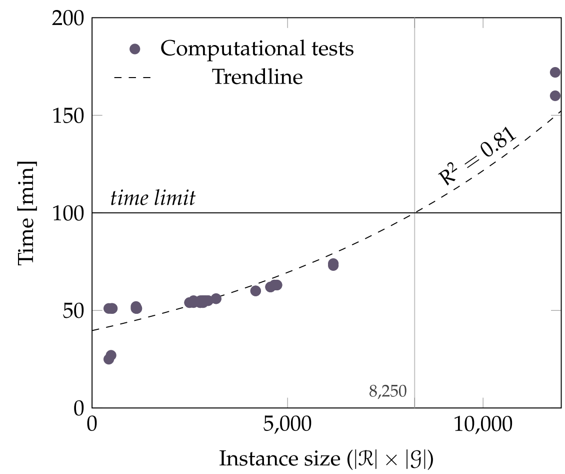

Fig. 5. Time to generate group decomposition and relaxation initial prices, using (M)INLP models.

4. Computational tests, results and discussion

This section aims to present and discuss the computational tests performed and results obtained, as a means to validate the different components of the solution method, as well as its overall relevance. This section describes the real-sized instances generated, based on the ones available in the literature, to perform these tests. Then, a baseline of comparison is established, in order to understand the impact of the capacity and pricing integration versus the (typical) sequential/hierarchical approach. Finally, the most relevant results will be presented and discussed, including a comparison with an exact approach to the original INLP using a non-linear solver.

4.1. Instances

In [28], twenty instances for the vehicle-reservation assignment problem in car rentals are presented. These instances are based on real data retrieved from a Portuguese car rental company and contain real information regarding detailed reservation requests and vehicles. The data regarding reservation requests was used to generate fourty realistic instances. This section explains how these instances were generated, with special focus on types of rentals and demand data.

Instances for the vehicle-reservation assignment problem

The vehicle-reservation assignment problem presented in [28] consists in assigning specific vehicles to fulfill reservation requests in order to maximize the profit of the car rental company. The instances made available [29] provided, among other parameters and information, full lists of reservation requests that the company had received up to specific dates. These requests were characterized by start and end date (and hour), start and end rental station, vehicle group requested, the profit expected from fulfilling the request, a priority status (related with customer confirmation) and an indication whether the customer would, if needed, accept a downgrade, which are used by the company as a last resource (upgrades were assumed to be always accepted).

As expected, in these instances the density of requests is higher for closer dates. Fig. 6 exemplifies for a specific instance how the reservations are distributed in time according to their start date and rental length.

Adaptation towards realistic rental types

In order to build significant and realistic rental types for the capacity-pricing integrating model, the reservation requests listed on the above-mentioned original instances were aggregated by rental types. All listed reservation requests that shared the following characteristics were aggregated by rental types: group of the vehicle required, starting week, ending week, starting zone, and ending z one.

As for the start and end time, rentals were aggregated in a weekly basis due to the strategic level of the decisions considered in this model, as discussed in Section 2. Moreover, only reservations that start within the time horizon of twelve weeks were considered.

As for the start and end location, the original list of reservations detailed the specific rental station. Once more, due to the type and impact of decisions considered in these models, the start rental stations mentioned in the original instances were aggregated in four zones. As for the end zone, since in the original instances it was almost always

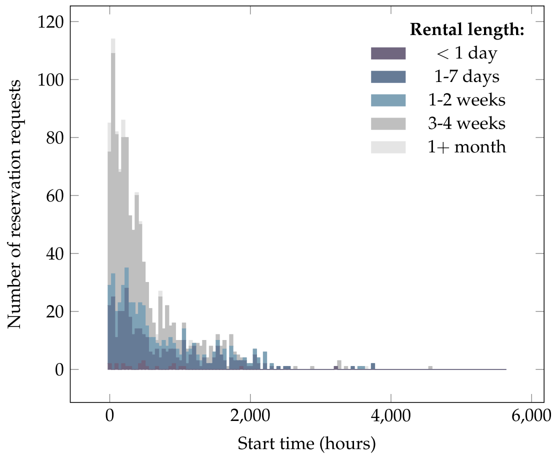

Fig. 6. Example of original instance for the vehicle-reservation assignment problem – distribution throughout the time horizon according to start date, by rental length.

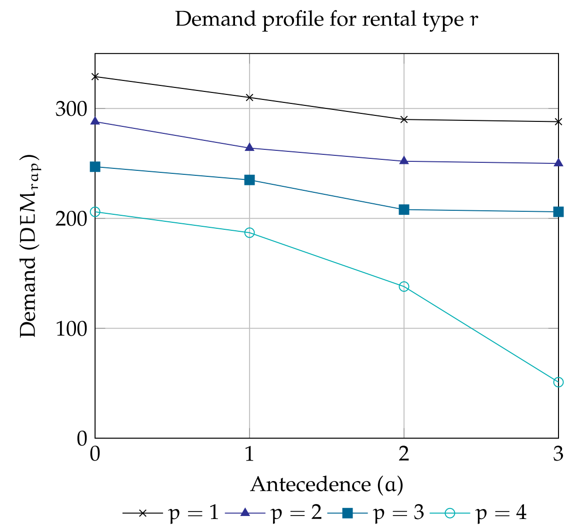

Fig. 7. Example of a generated demand profile for a specific rental type.

coincident with the start zone, it was also randomly determined.

From this aggregation, other parameters were also defined: the number of rental zones and the number of required groups are dependent on the rental types for each instance.

Demand inputs

The aggregation of the reservations described previously in this section provided a measure of the actual demand for the different types of rentals, based on the number of listed reservations that fell into each aggregated bin (rental type).

Nevertheless, in this model, the demand input DEM_{rap} for each rental type r depends on the price level $p = \{1, ..., P\}$ and on the antecedence $a = \{0, ..., A\}$ with which the rental request was made. Therefore, there was a need to generate different levels of demand for each rental type, related with the variation of these two indices. The demand given by the aggregation of reservations on the original instances (OD) provides a realistic reference for each rental type, and sets the reference demand for the first price level ($p = 1$). The reference demand (RD) for the following prices levels ($p > 1$) is strictly decreasing and is obtained by the following equation, where $|\mathcal{P}|$ stands for the number of price levels and α is a parameter that controls the gap between the levels.

$$RD_p = \begin{cases} OD, & p = 1 \\ RD_{p-1} - \frac{OD}{\alpha\,|\mathcal{P}|}, & p > 1 \end{cases} \quad (15)$$

Note that $\alpha \geq 1$ ensures that the reference demands are strictly decreasing and never null. However, if needed, it is also possible to model the demand-price relationship of a luxury product, where the demand increases as the price increases, by setting $\alpha < 0$. In this specific case, based on preliminary results and type of business, the value $\alpha = 2$ was chosen.

After setting the reference demand for each price level, one needs to generate the demand per antecedence as is detailed in the following equations, where β represents a randomly generated number such that $\beta \in]0, 1[$:

$$DEM_{rap} = \begin{cases} RD_p, & a = 0 \\ RD_{p+1} + \lfloor \beta(DEM_{r,p,a-1} - RD_{p+1} + 1) \rfloor, & a > 0 \wedge p < P \\ \lfloor \beta \times DEM_{r,p,a-1} \rfloor, & a > 0 \wedge p = P \end{cases} \quad (16)$$

The reference demand for each price level is associated with the first antecedence level. For the following antecedence levels, the demand value will be built from the reference demand of the next price level, to which will be added a fraction (β) of the gap between this and the demand value of the previous antecedence. On the last price level, a similar reasoning is applied, where the reference demand upon which the value is built is zero. This calculation ensures that all values that the demand takes across price levels are greater than the reference demand of the next price level. Fig. 7 represents a possible demand profile for a rental type which had an original demand $OD = 329$.

The generated demand profiles are based on realistic data from a car rental company that operates in Portugal, which is a relatively small market. In order to validate the results for bigger markets, a scale factor was also considered when generating the demand profiles. More specifically, two different instances were generated from each of the original listings of requests: the first is directly derived from the original instances ("scale factor" of 1) and represents a small/medium market such as Portugal, while the second has a "scale factor" of 100, which is multiplied to the former demand profile, thus representing the challenges faced by a company operating on a significantly bigger market.

Remaining inputs

Some parameters were unknown in the original instances or not fully adaptable to this model and were thus randomly generated, based on previously defined minimum and maximum values and respecting the relationship and hierarchy between vehicle groups, when applicable – for example, for the monetary value associated with each price level and group. The cost parameters were also generated in a similar fashion, yet maintaining a reasonable comparison between them when needed, e.g. the daily leasing costs are always significantly higher than the daily ownership/maintenance costs.

The upgrades were allowed in a fully nested way, i.e. the vehicle groups follow an hierarchy and rentals that require a least valued group can be upgraded to all groups that are more valued.

The budget was also randomly set yet for all instances it was proportional to the number of rental types and the scale factor of demand.

Table A.1, on Appendix A.4, details the main characteristics of each of the forty instances generated following the methodology described in this section. The instances are available at Oliveira et al. [30].

4.2. Baseline: Hierarchical resolution strategy

In order to justify the advantages of integrating capacity and pricing problems, a baseline resolution strategy was developed for comparison, based on a more traditional sequential or hierarchical decision-making process. The goal of this comparison is to determine the potential of integrating these problems. Fig. 8 depicts the overall hierarchical approach. In this framework, the first decisions made are the ones related

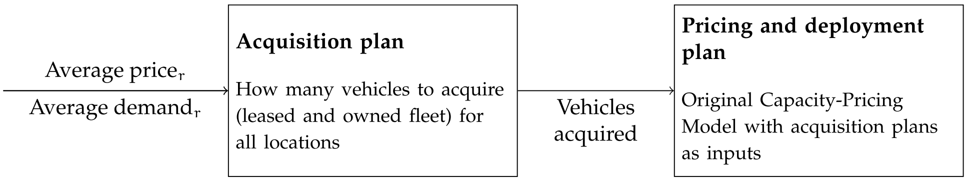

Fig. 8. Structure of the sequential baseline resolution strategy.

with the acquisitions (capacity), based on average prices and demand for each rental type. These decisions are made for the fleet as a whole, without considering the deployment between locations. The aggregated number of fulfilled rentals is also decided in order to account for upgrading decisions. On a second phase, the deployment, empty transfers and consequent stock decisions are made, as well as the pricing decisions, also implying the decision on the number of rentals fulfilled. The second phase thus consists on solving the original Capacity-Pricing Model where the acquisitions (for the overall pool of locations) are fixed inputs. The mathematical programming models used for the first and second phase are adaptations from the model presented in Section 2.2 and are detailed in Appendix A.3.

4.3. Structure of the tests

Proposed solution method:. To assess the performance of the solution method proposed in Section 3, each of the forty instances (see Section 4.1) was run twice. Firstly, the BRKGA was run with a fully random generation zero. Secondly, the heuristically generated initial prices were added to this generation when running the BRKGA. This makes it possible to measure the impact of using these initial prices.

To implement the BRKGA, the “brkgaAPI” [31] was adapted. Apart from the size of the chromosomes that is strictly dependent on the number of rental types and vehicle groups of the instance (see Section 3.1), the remaining main parameters were kept constant for all instances. The default values suggested in Toso and Resende [31] were adequate for the problem herein considered and thus used (Table 2). No parallel decoding was applied and one independent population was considered. This was due to the fact that the decoding procedures are based in mathematical models and are therefore significantly more complex and time-consuming than the ones usually used with this metaheuristic. Finally, the stopping criterion chosen for the BRKGA procedure was the solving time (1 h).

As for the heuristic generation of initial prices, a time limit was set for the group and time period decomposition and relaxation. The construction-generated prices are virtually immediate to generate. The group decomposition and the relaxation involve solving INLP models, as discussed in Section 3.2. For each of these models, a time limit of 10 min was set. As for the time period decomposition, two series of single-period Constraint Programming models were solved (for the two types of objective functions), with a time limit of also 10 min.

As for the final MIP run used to solve to optimality the best pricing strategy, a time limit of 10 min was applied, although it was never reached.

Comparison baseline:. Besides the proposed solution method, a baseline was developed based on the sequential resolution of the same problem (see Section 4.2). Each instance was solved as well using this method. In order to be conservative when assessing the performance of the proposed method, the time limit for the baseline was set to be slightly bigger than the actual maximum total time to solve when using the integrated method – 2 h and 40 min (9600 s).

Table 2
Main BRKGA parameters that are similar for all instances.

Parameter		Value
p	Size of population	50
pe	Fraction of population to be the elite-set	0.20
pm	Fraction of population to be replaced by mutants	0.10
ρ	Probability that offspring inherits an allele from elite parent	0.70

Exact approach to the Capacity-Pricing Model:. In order to validate the need for a non-exact solution method, the mathematical model presented in Section 2.2 was solved using a non-linear solver for each instance, with the same time limit set for the proposed method.

Technical details:. The algorithms, Mathematical Programming models and Constraint Programming models were developed in C++/IBM ILOG Concert Technology and were run on a HP Z820 Workstation computer with 128GB of RAM memory, and with 2 CPUs (Xeon E5-2687W 0 @ 3.10 GHz). The MIP and MINLP Solver used was CPLEX 12.6.3 and the CP solver used was CPLEX CP Optimizer 12.6.3.

4.4. Results and discussion

In the remainder of this section, the results will be presented and discussed. Four main issues will be discussed in more detail: the advantages and disadvantages of feeding heuristically generated initial prices to BRKGA's “generation zero”, which initial solutions perform the best, the overall performance of the integrating approach versus the sequential baseline, and the advantages and disadvantages of using a non-exact approach to this problem.

Impact of using initial prices

One of the novelties of the proposed approach was the use of heuristically generated initial pricing strategies to compose part of the initial generation of the BRKGA framework. It is thus important to understand how this part of the methodology impacts the overall performance in terms of solution quality and solving time.

First and foremost, using heuristically generated initial prices is significantly beneficial for the solution quality. Fig. 9 presents the improvement on the final solution obtained by the full methodology (including heuristically generated initial prices) versus a similar BRKGA procedure but with a *fully random* generation zero. The detailed values

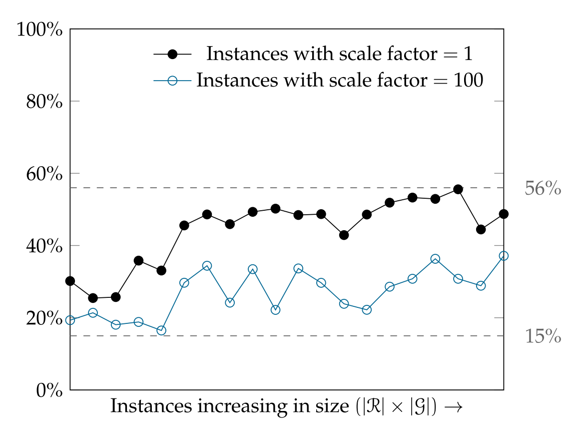

Fig. 9. Improvement on final solution MIP objective function value: heuristically generated initial prices added to generation zero versus fully random generation zero.

that support this figure can be found in Table A.2, on Appendix A.4. As observed, adding heuristically generated initial prices to the initial generation improves significantly the final solution, with results at least 15% better and, for some instances, more than 50%.

This level of improvement is due to the fact that the heuristically generated initial prices represent significantly good solutions. This statement is supported by two facts: **(1)** the fitness of the initial pricing strategies is consistently the best fitness of the initial generation, and **(2)** BRKGA has less room to evolve when these good solutions are directly added to the initial generation.

In fact, the best of these initial pricing strategies (evaluated in terms of their fitness – see Section 3.1.3) is for all instances the best of the initial generation **(1)**. Tables A.3 and A.4, in Appendix A.4, present the detailed results for the fully random BRKGA run and the run with additional heuristic initial prices. These tables include the best fitness obtained for the initial and last generations, the number of generations that the method was able to evolve within the time limit, and the final MIP run objective function value and gap. Moreover, for the run with additional heuristic initial prices (Table A.7), the best fitness obtained by these added initial prices is presented and it is possible to observe that it always matches the best fitness of the initial generation.

Moreover, there are significant differences when comparing the evolution that BRKGA is able to achieve (i.e., how much the best fitness of each generation increases from the initial to the last). Fig. 10 depicts this evolution (in percentage of increase from initial to last generation), discriminating the scale factor of the instances (i.e., the size of the market considered) and whether the “fully random generation zero” version of BRKGA or the “initial prices added to generation zero” version of BRKGA was used. Although the size of the market did not influence significantly this evolution, the type of methodology used did. When these (good) initial pricing strategies were added to the initial generation, the evolution was significantly smaller **(2)**.

Other interesting conclusion of Fig. 9 is that there is a difference on the improvement achieved by heuristically generating initial prices when comparing the size of the markets considered (scale factor). The improvement is bigger for instances that represent markets of the size as the one in study (scale factor = 1), for the same number of rental types and vehicle groups. This means that the procedures to generate initial pricing strategies are especially efficient for smaller markets. This is an expected result since generating initial prices often involves solving complex models with a limit on solution time: the smaller the instance, the better solutions are obtained.

Furthermore, there is a difference when comparing sizes of instances (indicated by the number of rental types and vehicle groups): the improvement achieved by adding initial prices to the initial generation tends to be more significant for bigger instances. This might be explained by the fully-random BRKGA loosing impetus when the solution space increases significantly.

Nevertheless, the boost on solution quality obtained by inserting heuristically generated initial prices on generation zero of the BRKGA procedure comes with a price to pay: the additional time to solve.

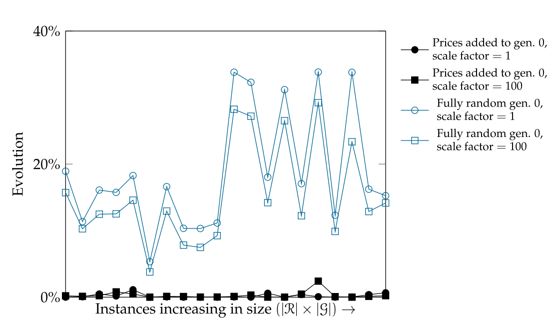

Fig. 10. Evolution of BRKGA: last generation best fitness versus initial generation best fitness.

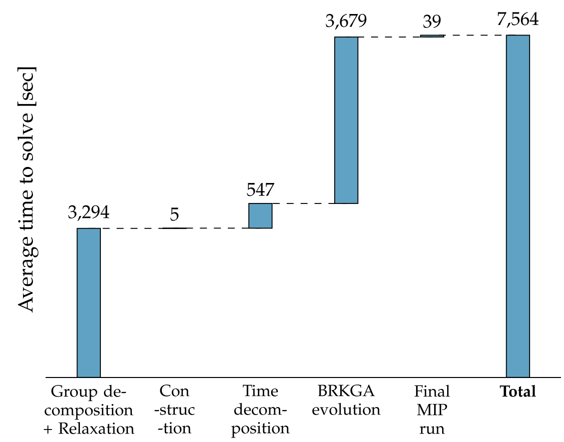

Fig. 11. Average time to solve each each component of the proposed method.

Fig. 11 shows the average time to solve each component for all instances. In Appendix A.4, the discriminated values for each instance can be found on Table A.5. These results reflect the time limits discussed on Section 4.3. Group decomposition and relaxation, which imply solving (M)INLP models, take more time than the other procedures to generate initial prices. However, as it will be discussed in the following paragraphs, they have a good performance. Also as expected, the MIP run to generate the final complete solution is fairly quick.

Overall, the generation of initial prices increases significantly the solving time. Nevertheless, this increase seems to be more than justified by the boost on solution quality obtained. Moreover, although the BRKGA is not able in average to evolve much of these already very good initial solutions, it is important to ensure that local optima are being avoided and that the variability of the solutions are improved as much as possible within a realistic time-frame. In fact, it is important to bear in mind that the ultimate goal of such a methodology is to be applied in a decision support system to help companies make better plans. Considering the time horizon of this problem, which aims to plan for a full selling season, 2 or 3 h, even if multiplied by a finite number of different runs to test different scenarios or strategic options (see Section 5 for a more detailed discussion on this topic), seem to be a small price to pay.

Initial heuristic prices – Comparing sources

Since the different sources of the initial prices added to the initial generation of BRKGA take a significantly different amount of time to solve, it is important to analyze their performance in terms of solution quality. The performance is evaluated by the following measures: (1) the number of times the initial price with the best fitness for some instance was generated by this specific source, and (2) how close were in average the fitness values of the initial prices generated by this source to the best fitness of the instance (translating each fitness value into a percentage of the instance's best fitness), as well as the stability (or variability) of this closeness (measured by its standard deviation). Fig. 12 presents these measures for the four sources of heuristically generated initial prices. Also, Table A.6 in Appendix A.4 details for each instance the best fitness obtained by each source.

The standard deviation of the measure “percentage of best price” is included to clarify the variability of the quality of the initial prices generated by each source. Nevertheless, this variability could be ascertained by the other two measures. In fact, if a certain source provides a significant number of times the *best price* of the instance but has a relatively low average *percentage of best price* – as happens for *construction* – it is due to a high variability of the quality of the generated prices.

Except for *time decomposition*, all sources provide more than once the initial price with the best fitness of the instance. *Relaxation* achieves

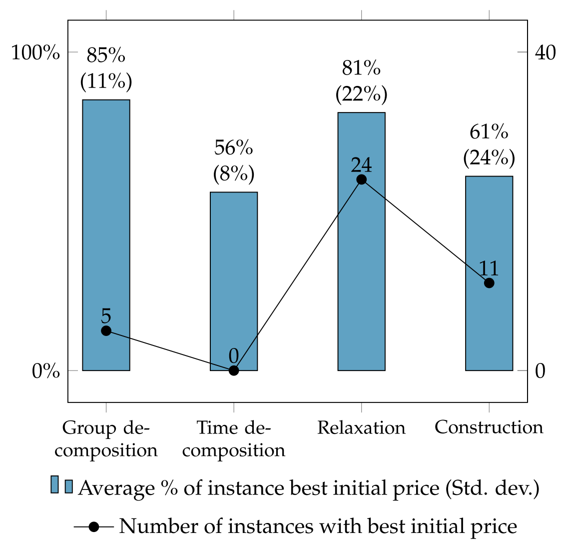

Fig. 12. Performance of the sources of initial prices: frequency of generation and closeness to the best initial price generated for each instance.

this for more than half of the instances. *Construction* of naive/obvious solutions seems to be quite powerful too. Nevertheless, as discussed above, it is not very stable. As expected, for the same instance if some obvious pricing strategy is very good some other is bound to be quite bad. *Group decomposition* is not often the best initial pricing strategy source yet it is very stable and, in average, significantly close to the best initial price. As for *time decomposition*, it is never the source of the best price and the average percentage of the best price is also penalized for that. Nevertheless, it shows the less variability of all sources and also consumes less time than *group decomposition* and *relaxation*.

In conclusion, all sources show some advantages, with results that seem to be highly dependent on the instance, and it thus seems reasonable to keep all of them in the procedure. Once again keeping the "big picture" of the ultimate application in mind, the modular structure of this part of the methodology also renders it easy to be translated into a decision support system, where these modules or sources can be turned on and off depending on the time/performance trade-off of the decision-maker.

Performance versus baseline

Comparing the proposed methodology with the sequential baseline presented in Section 4.2 establishes a quantitative proof of the value of integrating capacity and pricing decisions in car rental. This is, in fact, one of the main contributions of this work, as the integration of these decisions, although conceptually supported by other works (see Section 1), is now quantitatively justified versus a more traditional sequential or hierarchical approach.

The sequential baseline defined in Section 4.2 gives an upper bound on the value that can be actually obtained by companies, since it has (slightly) more processing time and is already using the novel model proposed, which is a detailed and enhanced representation of the problem. That is to say that it was designed to make the comparison between approaches fair and the relative performance of the integration measured in a conservative way.

Fig. 13 shows for each instance the improvement in terms of the final solution objective value obtained by the integrating versus the baseline (sequential) approach. It is possible to observe that the improvement is extremely significant, growing exponentially with the size of the instances. For every instance, the proposed approach is better than the sequential approach. Moreover, it was able to solve the two biggest instances, which could not be solved by the sequential method and are thus not represented. Considering the instances that could be solved, the average of improvement in objective value is 139%, yet it

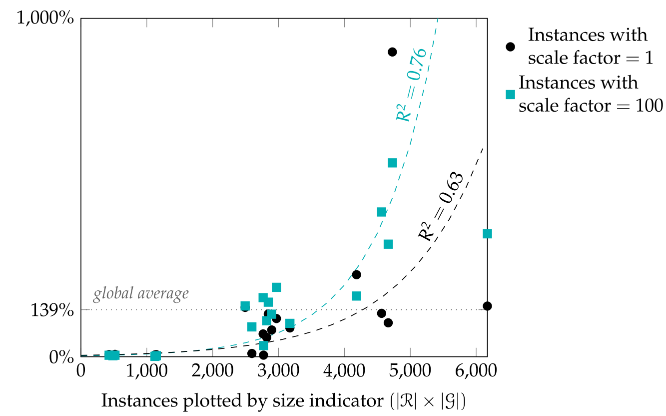

Fig. 13. Improvement of proposed integrating method versus baseline sequential approach, in terms of best objective value.

can go up to approximately 900%. Table A.7, in Appendix A.4, details the improvement achieved for each instance.

Exact vs. non-exact approach to the Capacity-Pricing Model

The mathematical model presented in Section 2.2 was implemented in a non-linear solver in order to evaluate the extent to what a straightforward exact approach could perform well and thus assess the need for non-exact solution methods for this problem.

Table 3 shows the overall results and Table A.8, on Appendix A.4, details these results. For 13 out of 40 instances, using this exact approach would lead to slightly better or similar results than using the proposed method. It is interesting to notice that the solver was always stopped by the time limit, even when running these instances.

In 27 out of 40 instances the proposed heuristic method outperforms the non-linear solver. In fact, for 7 of these instances, the exact solver is not even able to find a feasible solution different from the "trivial solution" (where all decision variables are set to 0 and the best objective value is also 0). For the 20 instances where the solver is able to find other feasible solutions, the proposed heuristic method achieves in average 204% of improvement on the objective value. It is interesting to notice that the size of the instances (measured by the number of rental types $|\mathcal{R}|$ times the number of vehicle groups $|\mathcal{G}|$) is likely not the only factor influencing the ability of the exact solver to find good solutions.

Overall, these results support the importance of developing heuristic solution methods as the one presented in this work to tackle the Capacity-Pricing Model.

5. Conclusions

This paper tackled the integration of capacity and pricing problems in car rentals, which are significantly relevant – both academically and for practitioners. A new integrating mathematical model was proposed, as well as a solution procedure based on its decomposition, guided by a biased random-key genetic algorithm. The value of integrating these problems was established and empirically measured by successfully comparing the results of the proposed solution method with the ones obtained by a hierarchical and sequential approach.

The solution methodology herein proposed may be used by companies to support their decisions, since it was built on a realistic model and is relatively fast to produce good solutions. In fact, as previously discussed, its average solving times and modular structure allows for it to be used as part of a decision support system where the final user could run the procedure several times for different scenarios, such as different levels of investment on the fleet or different demand forecasts. At the same time, it would be possible to select different sources of heuristically generated initial prices to better control time to solve and

Table 3
Overall comparison of the best values obtained by the proposed solution method (here denoted *BRKGA*) and by the non-linear solver (here denoted *INLP*).

	# instances	Average size indicator ($\lvert\mathcal{R}\rvert \times \lvert\mathcal{G}\rvert$)	Average improvement on best objective value: BRKGA vs. INLP
BRKGA **worst**performance than INLP	**4**	2339	−2%
BRKGA **similar**performance to INLP	**9**	695	0%
BRKGA **better** performance than INLP	**27**		
− *INLP: 1+ feasible solutions*	20	3693	204%
− *INLP: trivial solution*	5	3441	–
− *INLP: no feasible solutions*	2	11845	–

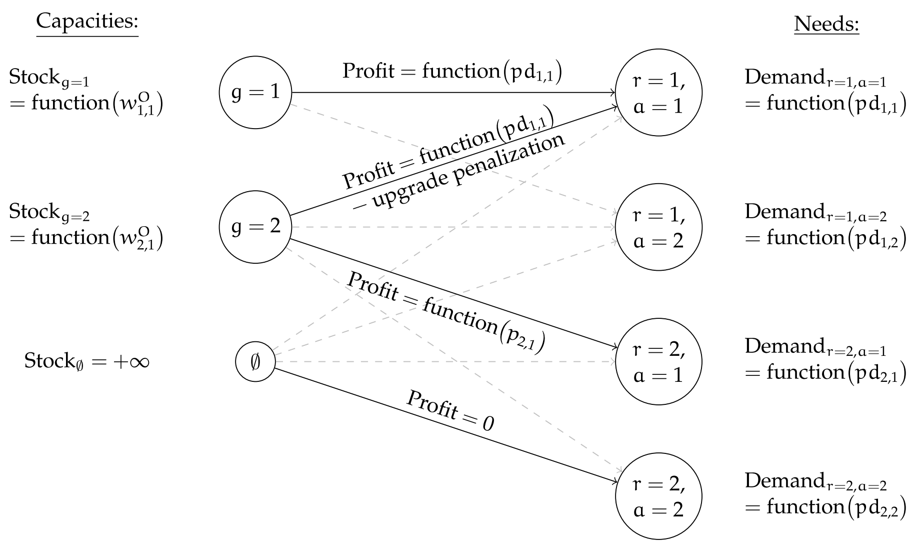

Fig. 14. Representation as a "transportation" problem, with capacities on the origin nodes (stock of vehicle groups) and needs on the destination nodes (demand for related rental types): zoom in a specific location (s = 1) and time (t = 1). The notation follows the mathematical model notation presented in Section 2.2.1 and, for

variability of solutions.

As for future work, the better development, modeling and integration of the demand-price relationship in the model could be developed. In order to obtain more realistic and robust solutions, the stochasticity of demand could be considered. Moreover, further economic studies could help develop a more precise demand input for the model, thus leading to more accurate results.

Finally, as it was previously mentioned, this sector shares important characteristics with emerging mobility systems, such as carsharing, namely fleet mobility and flexibility, heavy dependency on efficiency and high occupation rates, and the ability to use prices to manage demand. This work can thus be extended, e.g. by allowing the free floating of the fleet (dropping-off vehicles in any location, not only previously established locations), to help carsharing companies better manage their fleet and pricing schemes. In Wagner et al. [32], the challenges posed by the spatial flexibility of free float are identified and the authors propose a model to explain the variation of activity through the proximity of certain points of interest. Building on this type of demand-modelling techniques, the work developed in this paper can be further extended to the rapidly expanding market of carsharing.

Acknowledgments

This work is partially financed by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project "POCI-01-0145-FEDER-029279".

The authors would like to thank the company that inspired this work for the support and realistic perspective on the challenges tackled.

Appendix A

A.1. Insights on model and problem structure

The mathematical model developed on Section 2.2 brings some structural insights to the problem at hand. An analogy can be made between specific sections of this formulation and transportation problems modeled with linear programming. One of the main differences resides in the fact that most of the conventional parameters of transportation problems (capacities, needs, unit costs,...) are, in this formulation, decision variables. Nevertheless, interesting insights can help understand the problem structure and model behavior.

Fig. 14 depicts this analogy for a section of the problem consisting of a specific *location* and *time*. For clarity, the problem is here simplified: it considers only owned fleet (O), two vehicle groups $g = \{1, 2\}$, two types of rentals $r = \{1, 2\}$ and two different antecedence levels $a = \{1, 2\}$, and it is focused on a specific location $s = 1$ and time period $t = 1$. In this small example, both rental types $r = \{1, 2\}$ refer to rentals that start in the same location and period of time, yet $r = 1$ requests a vehicle of group $g = 1$ and $r = 2$ of group $g = 2$. Also, the upgrading policy of the company states that a rental type requesting group $g = 1$ can be upgraded to a vehicle of group $g = 2$, but not the other way around.

The origins of the transportation problem (nodes on the left) are the different vehicle groups plus a virtual node that represents all rental requests

Table A.1
Main characteristics of the generated instances.

Instance	Base instance [28]	Scale factor	# rental types ($\|\mathcal{R}\|$)	# vehicle groups($\|\mathcal{G}\|$)	Size indicator ($\|\mathcal{R}\| \times \|\mathcal{G}\|$)
1	8	1	428	1	428
2	8	100	428	1	428
3	18	1	486	1	486
4	18	100	486	1	486
5	3	1	517	1	517
6	3	100	517	1	517
7	5	1	562	2	1124
8	5	100	562	2	1124
9	12	1	572	2	1144
10	12	100	572	2	1144
11	20	1	831	3	2493
12	20	100	831	3	2493
13	11	1	865	3	2595
14	11	100	865	3	2595
15	19	1	922	3	2766
16	19	100	922	3	2766
17	13	1	924	3	2772
18	13	100	924	3	2772
19	1	1	564	5	2820
20	1	100	564	5	2820
21	4	1	948	3	2844
22	4	100	948	3	2844
23	7	1	724	4	2896
24	7	100	724	4	2896
25	6	1	742	4	2968
26	6	100	742	4	2968
27	9	1	793	4	3172
28	9	100	793	4	3172
29	14	1	1046	4	4184
30	14	100	1046	4	4184
31	16	1	1141	4	4564
32	16	100	1141	4	4564
33	17	1	933	5	4665
34	17	100	933	5	4665
35	15	1	1182	4	4728
36	15	100	1182	4	4728
37	2	1	1234	5	6170
38	2	100	1234	5	6170
39	10	1	2369	5	11,845
40	10	100	2369	5	11,845

Table A.2
Improvement on final solution MIP objective function value: heuristically generated initial prices added to generation zero versus fully random generation zero.

Instance of scale factor = 1	Instance of scale factor = 100	Size indicator ($\|\mathcal{R}\| \times \|\mathcal{G}\|$)	Improvement for scale factor = 1	Improvement for scale factor = 100
1	2	428	30.2%	19.3%
3	4	486	25.5%	21.4%
5	6	517	25.7%	18.1%
7	8	1124	35.8%	18.8%
9	10	1144	33.1%	16.5%
11	12	2493	45.6%	29.7%
13	14	2595	48.6%	34.4%
15	16	2766	45.9%	24.2%
17	18	2772	49.3%	33.5%
19	20	2820	50.2%	22.2%
21	22	2844	48.4%	33.7%
23	24	2896	48.7%	29.7%
25	26	2968	42.9%	23.9%
27	28	3172	48.6%	22.2%
29	30	4184	51.9%	28.6%
31	32	4564	53.3%	30.8%
33	34	4665	52.9%	36.3%
35	36	4728	55.6%	30.8%
37	38	6170	44.5%	28.9%
39	40	11,845	48.7%	37.2%
		Average	44.3%	27.0%

Table A.3
Proposed solution method – results for fully random BRKGA (without heuristically generated initial prices).

Instance	Best fitness generation zero	Best fitness last generation	# generations	Final MIP OF value	Final MIP gap
1	42,804	56,153	182	56,147	0%
2	3,879,500	4,908,510	245	4,908,490	0%
3	51,435	68,815	301	68,814	0%
4	4,862,010	5,997,660	184	5,997,660	0%
5	65,714	87,933	293	87,933	0%
6	6,142,810	7,877,400	289	7,877,400	0%
7	34,446	45,570	221	45,559	0%
8	3,084,800	3,924,010	310	3,923,860	0%
9	36,584	48,961	236	48,951	0%
10	3,177,430	4,106,230	280	4,106,070	0%
11	56,201	64,770	63	64,750	0%
12	4,945,060	5,645,490	59	5,645,440	0%
13	59,889	70,108	49	70,096	0%
14	5,306,280	5,955,530	54	5,955,480	0%
15	61,698	71,712	53	71,699	0%
16	5,470,600	6,174,840	53	6,174,680	0%
17	65,041	75,846	53	75,821	0%
18	5,755,090	6,499,770	49	6,499,680	0%
19	32,557	38,720	45	38,692	0%
20	2,852,850	3,301,400	41	3,301,250	0%
21	63,118	73,281	49	73,271	0%
22	5,592,730	6,290,130	50	6,290,060	0%
23	38,718	44,815	41	44,798	0%
24	3,334,590	3,752,240	43	3,752,090	0%
25	46,298	54,649	47	54,625	0%
26	4,097,910	4,680,220	43	4,680,120	0%
27	41,062	48,569	61	48,555	0%
28	3,593,020	4,116,970	77	4,116,700	0%
29	64,546	72,491	34	72,462	0%
30	5,629,730	6,185,670	36	6,185,630	0%
31	77,776	85,802	28	85,780	0%
32	6,795,650	7,303,810	28	7,303,520	0%
33	51,107	56,805	31	56,783	0%
34	4,417,890	4,826,680	33	4,826,640	0%
35	78,408	86,524	30	86,503	0%
36	6,777,220	7,307,780	31	7,307,720	0%
37	75,955	84,580	32	84,558	0%
38	6,614,440	7,294,210	39	7,294,080	0%
39	161,692	170,322	16	170,274	0%
40	14,656,500	15,212,500	15	15,212,500	0%

that will not be fulfilled, i.e. not assigned to a group. The capacity in the origins represents the number of vehicles of the specific group available at the specific location and time and is a function of the acquisition decision for this group. For the virtual “no-group” origin node, the capacity is unlimited.

The destinations (nodes on the right) are the requests for the rental types, with a certain antecedence. The need of each destination is the demand of each rental type, for each antecedence, which is a function of the corresponding pricing decision (here the chosen price level is represented as $pd_{ra} = \sum_{p \in \mathcal{P}} p \times q_{rap}$).

The unitary link “parameters” are here also highly dependent on the pricing decision. They represent the profit obtained from fulling a rental, which is a function of the price charged. If an upgrade is offered, there is a penalization to account for. All links with origin in “no-group-node” have a null profit.

In this problem, for a specific time and location, the flow between origins – available fleet – and destinations – rental requests – represents the actual number of rentals fulfilled. The limitations on the number of rentals by the stock of available vehicles and by the demand for each type of request reflect the main constraints of the problem, presented before. This type of analogy allows to better understand the structure of the problem and the relationship between the different decisions.

A.2. Constraint programming single-period model

Considering the indices and parameters and based on the Capacity-Pricing Model presented in Section 2.2, the following single-period Constraint Programming model was developed.

Note that each single-period model tackles a different set of rental types $\mathcal{R}_t$: $dout = t$, consisting of the ones whose starting date (*dout*) falls within the considered time period *t*.

Decision variables

The following table presents the decision variables, as well as their domains.

$q_{ra} = \{1, ..., P\}$	price level charged for rental type r with antecedence a
$w_{gs}^{L} = \{0, ..., ubw^{L}\}$	

Table A.4
Proposed solution method – results for BRKGA with heuristically generated initial prices.

Instance	Best fitness initial solutions	Best fitness generation zero	Best fitness last generation	# generations	Final MIP OF value	Final MIP gap
1	73,087	73,087	73,087	137	73,087	0%
2	5,856,040	5,856,040	4,908,510	150	5,856,790	0%
3	86,301	86,301	68,815	139	86,329	0%
4	7,279,250	7,279,250	5,997,660	110	7,279,330	0%
5	110,504	110,504	87,933	129	110,548	0%
6	9,291,210	9,291,210	7,877,400	118	9,301,070	0%
7	61,889	61,889	45,570	72	61,878	0%
8	4,647,930	4,647,930	3,924,010	113	4,662,610	0%
9	65,101	65,101	48,961	66	65,139	0%
10	4,669,800	4,669,800	4,106,230	94	4,782,210	0%
11	93,649	93,649	64,770	23	94,248	0%
12	7,305,510	7,305,510	5,645,490	31	7,321,720	0%
13	103,769	103,769	70,108	18	104,172	0%
14	7,969,430	7,969,430	5,955,530	25	8,003,950	0%
15		104,247	71,712	19	104,624	0%
16	7,658,140	7,658,140	6,174,840	26	7,667,490	0%
17	113,191	113,191	75,846	18	113,213	0%
18	8,664,240	8,664,240	6,499,770	26	8,674,780	0%
19	58,140	58,140	38,720	19	58,121	0%
20	4,024,910	4,024,910	3,301,400	18	4,033,130	0%
21	108,267	108,267	73,281	19	108,768	0%
22	8,388,140	8,388,140	6,290,130	32	8,406,700	0%
23	66,509	66,509	44,815	18	66,613	0%
24	4,826,660	4,826,660	3,752,240	18	4,865,230	0%
25	77,619	77,619	54,649	20	78,055	0%
26	5,797,090	5,797,090	4,680,220	19	5,796,600	0%
27	71,358	71,358	48,569	28	72,141	0%
28	5,006,880	5,006,880	4,116,970	34	5,030,720	0%
29	110,077	110,077	72,491	13	110,046	0%
30	7,949,970	7,949,970	6,185,670	21	7,954,810	0%
31	131,509	131,509	85,802	16	131,486	0%
32	9,550,530	9,550,530	7,303,810	21	9,552,340	0%
33	86,853	86,853	56,805	15	86,827	0%
34	6,577,990	6,577,990	4,826,680	20	6,579,660	0%
35	134,592	134,592	86,524	17	134,573	0%
36	9,548,500	9,548,500	7,307,780	22	9,557,750	0%
37	122,101	122,101	84,580	20	122,145	0%
38	9,387,120	9,387,120	7,294,210	26	9,399,680	0%
39	253,300	253,300	170,322	9	253,255	0%
40	20,864,700	20,864,700	15,212,500	12	20,864,600	0%

Number of vehicles of group g acquired by leasing available at location s. The domain upper bound is based on the maximum demand for all rental types, considering the sum of all antecedence levels:
$ubw^L = \sum_{r\in\mathcal{R}_t}\sum_a \max_p\{DEM_{rap}\}$

$w_{gs}^O = \{0,\dots,ubw^O\}$ Number of vehicles of group g acquired for the owned fleet available in location s (only for $t = 0$). The domain upper bound is based on the available budget:
$ubw^O = \left\lfloor \frac{B}{C_g} \right\rfloor$

$x_{gs}^{L/O} = \{0,\dots,2ubw^{L/O}\}$ Number of leased (L) or owned (O) vehicles of group g located at s. The domain upper bound is based on purchases, yet not limited to them (interconnected time periods).

$y_{s1s2g}^{L/O} = \{0,\dots,2ubw^{L/O}\}$ Number of leased (L) or owned (O) vehicles of group g transferred from location $s1$ to location $s2$ (starting on this time period). The domain is limited by the stock available.

$u_{rag}^{L/O} = \{0,\dots,ubu\}$ Number of fulfilled rentals requested as rental type r with antecedence a that are served by a leased (L) or owned (O) vehicle of group g. The domain upper bound is based on the maximum demand for each specific rental type and antecedence:
$ubu = \max_p\{DEM_{rap}\}$

$f_g^{L/O} = \{0,\dots,prevW^{L/O} + ubu\}$ Auxiliary variable: total leased (L) or owned (O) fleet of group g. The domain upper bound is given by the purchases of previous time periods plus the upper bound of the current rentals:
$prevW^{L/O} = \sum_{t'<t}\sum_g [w_{gs}^{L/O}]_{t'}$

Also, since the single-period model was developed to be solved sequentially for all time periods, all inputs (parameters) that are given by the result of the decision variables from the previous time period will be noted with the prefix P. Note that the variable that represents the decision on the number of rentals to fulfill that start in the specific t, $u_{rag}^{L/O}$ has $|\mathcal{R}_t|$ elements on the first index while the input $Pu_{rag}^{L/O}$ that stores these decisions for past time periods, has a corresponding number of $|\mathcal{R}|$ elements.

Table A.5
Time to solve each component of the proposed method..

Instance	Time [s]					Total
	Group decomposition + Relaxation	Construc-tion	Time period decomposition	BRKGA	Final MIP run	
1	1512	5	605	3612	1	5735
2	3037	4	511	3624	0	7177
3	1643	4	468	3601	1	5717
4	3047	4	422	3626	0	7100
5	3040	5	510	3612	1	7167
6	3034	5	469	3607	0	7114
7	3057	5	516	3626	9	7213
8	3109	5	473	3624	2	7212
9	3057	5	518	3652	3	7234
10	3069	0	565	3641	3	7279
11	3217	5	573	3750	69	7615
12	3215	5	579	3609	5	7412
13	3242	6	481	3723	43	7496
14	3291	5	625	3699	3	7623
15	3270	5	578	3643	38	7534
16	3265	5	535	3756	4	7566
17	3265	5	480	3646	21	7417
18	3271	5	531	3651	4	7463
19	3279	5	525	3725	41	7575
20	3269	0	632	3635	7	7544
21	3283	5	530	3726	46	7589
22	3267	5	582	3648	4	7506
23	3289	5	578	3792	109	7772
24	3317	5	684	3744	5	7754
25	3296	5	531	3648	51	7530
26	3300	5	585	3705	5	7600
27	3342	0	534	3679	46	7601
28	3338	5	489	3655	14	7501
29	3613	5	584	3615	92	7910
30	3584	5	596	3643	6	7834
31	3729	5	593	3811	85	8223
32	3703	5	596	3738	6	8048
33	3775	4	546	3681	185	8191
34	3759	5	549	3738	4	8054
35	3784	5	491	3671	117	8069
36	3773	5	597	3611	25	8011
37	4364	5	599	3751	195	8913
38	4455	5	510	3729	29	8728
39	10,319*	6	530	3764	224	14,843
40	9,592*	6	585	3756	58	13,997

* Run despite size limitation discussed on Section 3.2. Not included in average.

CP model

Absolutely myopic objective function.

max Profit from fulfilled rentals − Buying cost−Leasing cost (for the time period) − Ownership cost − Empty transfer cost − Penalty for upgrading=

$$\max \sum_{r=1}^{|R_t|}\sum_{a=1}^{A}\left(\left(\sum_{g=1}^{G} u^L_{rag} + u^O_{rag}\right)PRI_{q_{ra}g_r}\right) - \sum_{g=1}^{G}\left(\sum_{s=1}^{S} w^O_{gs}\right)COS_g - \sum_{g=1}^{G} f^L_g LEA_g - \sum_{g=1}^{G} f^O_g OWN_g - \sum_{s1=1}^{S}\sum_{s2=1}^{S}\sum_{g=1}^{G}\left(y^L_{s1s2gt} + y^O_{s1s2gt}\right)TC_{gs1s2} - \sum_{g=1}^{G}\sum_{r\in\mathcal{R}^{g}_{-}}\sum_{a=1}^{A}\left(u^L_{rag} + u^O_{rag}\right)PYU \tag{17}$$

Alternative: less myopic objective function (different leasing cost term).

Table A.6
Comparison of the best fitness obtained by each approach to generate initial prices (best value for each instance highlighted).

Instance	Group decomposition	Time decomposition	Relaxation	Construction
1	73,040	43,363	**73,087**	66,141
2	5,762,000	3,057,180	**5,856,040**	5,380,860
3	86,238	38,051	**86,301**	78,033
4	6,945,850	3,260,770	**7,279,250**	6,649,860
5	**110,504**	46,330	110,341	100,056
6	8,911,330	4,788,130	**9,291,210**	8,546,980
7	**61,889**	35,081	61,107	55,109
8	4,070,060	2,531,770	**4,647,930**	4,305,980
9	**65,101**	37,676	64,189	58,167
10	**4,669,800**	2,836,990	4,632,280	4,493,570
11	83,492	55,080	**93,649**	88,474
12	5,115,850	4,694,010	**7,305,510**	6,871,040
13	83,727	57,776	**103,769**	97,267
14	5,382,840	5,137,320	**7,969,430**	7,538,400
15	83,720	62,568	**104,247**	97,992
16	5,700,040	4,808,470	5,717,840	**7,658,140**
17	92,592	52,242	**113,191**	104,677
18	5,909,810	4,851,550	**8,664,240**	8,235,850
19	**58,140**	33,132	54,507	52,179
20	3,676,720	2,717,660	3,308,830	**4,024,910**
21	92,313	62,265	**108,267**	101,476
22	5,782,670	5,087,590	**8,388,140**	7,997,890
23	63,790	41,376	**66,509**	61,862
24	3,903,230	3,395,280	**4,826,660**	4,654,030
25	73,544	45,364	**77,619**	74,272
26	4,767,000	3,895,570	4,724,890	**5,797,090**
27	64,423	42,227	**71,358**	66,208
28	3,891,500	3,136,800	3,661,660	**5,006,880**
29	86,325	63,046	**110,077**	102,936
30	5,766,210	5,217,100	5,781,400	**7,949,970**
31	102,272	77,070	**131,509**	123,660
32	7,005,540	6,272,680	7,005,540	**9,550,530**
33	77,371	54,172	**86,853**	81,755
34	4,521,370	4,108,220	**6,577,990**	6,232,290
35	105,184	73,569	**134,592**	125,208
36	6,945,900	6,363,110	6,945,900	**9,548,500**
37	104,925	73,008	117,303	**122,101**
38	6,748,610	5,572,490	6,748,610	**9,387,120**
39	224,797	130,562	159,451	**253,300**
40	17,314,300	11,503,500	15,803,700	**20,864,700**

Table A.7
Improvement of proposed integrating method versus baseline sequential approach.

Instance of scale factor = 1	Instance of scale factor = 100	Size indicator ($R \times G$)	Improvement for scale factor = 1	Improvement for scale factor = 100
1	2	428	6.4%	4.4%
3	4	486	4.2%	3.0%
5	6	517	6.9%	4.4%
7	8	1124	5.3%	1.0%
9	10	1144	6.0%	3.1%
11	12	2493	145.9%	150.0%
13	14	2595	9.6%	88.2%
15	16	2766	67.5%	174.6%
17	18	2772	4.7%	32.9%
19	20	2820	57.1%	106.3%
21	22	2844	126.1%	161.2%
23	24	2896	79.1%	125.8%
25	26	2968	112.8%	205.1%
27	28	3172	85.2%	98.6%
29	30	4184	242.4%	179.4%
31	32	4564	128.2%	427.8%
33	34	4665	100.3%	332.7%
35	36	4728	900.6%	572.9%
37	38	6170	149.7%	363.3%
39	40	11,845	-	-
		Average	117.8%	158.7%

Table A.8
Comparison of the best values obtained by the non-linear solver (INLP) and BRKGA with heuristically generated initial prices.

Instance	Size indicator ($\|\mathcal{R}\| \times \|\mathcal{G}\|$)	BRKGA best value	INLP best value	Improvement BRKGA vs. INLP
1	428	73,082	73,059	0%
2	428	5,854,440	5,852,920	0%
3	486	86,349	86,306	0%
4	486	7,276,740	7,275,560	0%
5	517	110,562	110,482	0%
6	517	9,299,180	9,309,570	0%
7	1124	61,709	61,790	0%
8	1124	4,661,830	4,652,980	0%
9	1144	64,730	65,042	0%
10	1144	4,814,190	4,855,770	−1%
11	2493	96,716	83,401	16%
12	2493	7,346,430	0	
13	2595	105,274	106,594	−1%
14	2595	7,954,920	5,742,290	39%
15	2766	105,954	104,982	1%
16	2766	8,108,450	5,192,100	56%
17	2772	113,830	115,564	−2%
18	2772	8,663,270	7,672,260	13%
19	2820	56,125	44,669	26%
20	2820	4,042,860	2,921,580	38%
21	2844	108,628	110,285	−2%
22	2844	8,378,750	6,100,250	37%
23	2896	66,729	39,086	71%
24	2896	4,675,110	0	
25	2968	78,988	44,493	78%
26	2968	5,827,720	0	
27	3172	71,909	66,283	8%
28	3172	5,400,970	3,901,760	38%
29	4184	102,913	75,652	36%
30	4184	7,949,700	0	
31	4564	130,884	71,048	84%
32	4564	9,555,990	667,525	1332%
33	4665	82,036	58,443	40%
34	4665	6,234,340	0	
35	4728	134,346	64,661	108%
36	4728	9,549,240	1,688,550	466%
37	6170	122,200	28,783	325%
38	6170	9,400,800	689,945	1263%
39	11,845	253,245		
40	11,845	20,864,600		

$$\begin{aligned}\max\ & \text{Profit from fulfilled rentals} - \text{Buying cost} - \text{Leasing cost (for the entire leasing period)} \\ & - \text{Ownership cost} - \text{Empty transfer cost} - \text{Penalty for upgrading} = \\ & \max \sum_{r=1}^{|R_t|}\sum_{a=1}^{A}\left(\left(\sum_{g=1}^{G} u_{rag}^{L} + u_{rag}^{O}\right)PRI_{q_{ra}gr_r}\right) - \sum_{g=1}^{G}\left(\sum_{s=1}^{S} w_{gs}^{O}\right)COS_g \\ & - \sum_{g=1}^{G} f_g^{L} LEA_g LP_g - \sum_{g=1}^{G} f_g^{O} OWN_g - \sum_{s1=1}^{S}\sum_{s2=1}^{S}\sum_{g=1}^{G}\left(y_{s1s2gt}^{L} + y_{s1s2gt}^{O}\right)TC_{gs1s2} \\ & - \sum_{g=1}^{G}\sum_{r\in\mathcal{R}^{g}_{-}}\sum_{a=1}^{A}\left(u_{rag}^{L} + u_{rag}^{O}\right)PYU\end{aligned} \tag{18}$$

Stock calculating constraintsOwned fleet.

$$\mathbf{s.\,t.}\, x_{gs}^{O} = \begin{cases} INX_{gts}^{O} + w_{gs}^{O}, & t = 0 \\ [Px_{gs}^{O}]_{t-1} + ONY_{gts}^{O} + ONU_{gts}^{O} + \sum_{r\in\mathcal{R}_{s,t}^{in}}\sum_{a=1}^{A} Pu_{r,a,g}^{O} - \sum_{r\in\mathcal{R}_{s,t}^{out}}\sum_{a=1}^{A} Pu_{r,a,g}^{O} + \sum_{c=1}^{S} y_{c,s,g,t-TT_{cs}-1}^{O} - \sum_{c=1}^{S} y_{s,c,g,t-1}^{O}, & t > 0 \end{cases} \quad \forall\, g, s \tag{19}$$

Leased fleet.

$$x_{gs}^{L}=\begin{cases}0, & t=0\\ [Px_{gs}^{L}]_{t-1}+ONY_{gts}^{L}+ONU_{gts}^{L}\\ +\sum_{r\in\mathcal{R}_{s,t}^{in}}\sum_{a=1}^{A}Pu_{r,a,g}^{L}-\sum_{r\in\mathcal{R}_{s,t}^{out}}\sum_{a=1}^{A}Pu_{r,a,g}^{L} & 0<t\le LP_g\\ +\sum_{c=1}^{S}y_{c,s,g,t-TT_{cs}-1}^{L}-\sum_{c=1}^{S}y_{s,c,g,t-1}^{L}\\ +wL_{gs}, \\ [Px_{gs}^{L}]_{t-1}+ONY_{gts}^{L}+ONU_{gts}^{L}\\ +\sum_{r\in\mathcal{R}_{s,t}^{in}}\sum_{a=1}^{A}Pu_{r,a,g}^{L}-\sum_{r\in\mathcal{R}_{s,t}^{out}}\sum_{a=1}^{A}Pu_{r,a,g}^{L} & t>LP_g\\ +\sum_{c=1}^{S}y_{c,s,g,t-TT_{cs}-1}^{L}-\sum_{c=1}^{S}y_{s,c,g,t-1}^{L}\\ +w_{gs}^{L}-[Pw_{g}^{L}s]_{t-LP_g-1}, \end{cases}\qquad \forall\, g,\, s \tag{20}$$

Capacity on origins / needs on destinations constraints.

$$\sum_{g=1}^{G}\left(u_{rag}^{L}+u_{rag}^{O}\right)\le DEM_{raq_{ra}}\qquad \forall\, r\in\mathcal{R}_t,\, a \tag{21}$$

$$\sum_{r\in\mathcal{R}_{st}^{out}}\sum_{a=1}^{A}u_{rag}^{L/O}+\sum_{c=1}^{S}y_{scg}^{L/O}\le x_{gs}^{L/O}\qquad \forall\, g,\, s \tag{22}$$

Business-related constraints.

$$UPG_{gr_r,g}=0\Rightarrow\sum_{a=1}^{A}\left(u_{rag}^{L}+u_{rag}^{O}\right)=0\qquad \forall\, r\in\mathcal{R}_t,\, g \tag{23}$$

$$(\text{only for } t=0)\qquad \sum_{s=1}^{S}\sum_{g=1}^{G}w_{gs}^{O}COS_g\le BUD \tag{24}$$

Other constraints.

$$f_{g}^{L/O}=\sum_{s=1}^{S}x_{gs}^{L/O}+\sum_{r\in\mathcal{R}_t^{use}}\sum_{a=1}^{A}Pu_{r,a,g}^{L/O}+\sum_{s1=1}^{S}\sum_{s2=1}^{S}\sum_{t'=t-TT_{s1s2}}^{t-1}\left[Py_{s1,s2,g}^{L/O}\right]_{t'}\qquad \forall\, g \tag{25}$$

A.3. Sequential resolution strategy

The sequential resolution strategy presented in Section 4.2 was developed as a baseline to assess the performance of the integration strategy and consists on solving two models sequentially: an acquisition plan model and a pricing and deployment plan model. The mathematical formulation of these models will be presented in this appendix.

A3.1. Acquisition plan model

Considering the indices and parameters and based on the Capacity-Pricing Model presented in Section 2.2, the following MIP adaptation was developed for the acquisition plan. Moreover, this model also requires as inputs the average price level ($avgq_g$) and average demand ($avgDEM_g$) per rental type, which were linearly derived from each instance (see Section 4.1).

Decision variables:

w_{gt}^{L}	Number of vehicles of group g acquired by leasing at time $t=\{0,\ \ldots, T-1\}$
w_{g}^{O}	Number of vehicles of group g acquired for the owned fleet available at time $t=0$
$x_{gt}^{L/O}$	Number of leased (L) or owned (O) vehicles of group g at time t
$u_{rg}^{L/O}$	Number of fulfilled rentals requested as rental type r that are served by a leased (L) or owned (O) vehicle of group g
$f_{gt}^{L/O}$	Auxiliary variable: total leased (L) or owned (O) fleet of group g at time t

Mathematical Integer Program (MIP)

$$\max \sum_{r=1}^{R}\left(\sum_{g=1}^{G} u_{rg}^{L}+u_{rg}^{O}\right)PRI_{avgq_g,gr_r}-\sum_{g=1}^{G}\left(\sum_{s=1}^{S} w_{gs}^{O}\right)COS_g$$
$$-\sum_{g=1}^{G}\left(\sum_{t=1}^{T} f_{gt}^{L}\right)LEA_g-\sum_{g=1}^{G}\left(\sum_{t=1}^{T} f_{gt}^{O}\right)OWN_g-\sum_{g=1}^{G}\sum_{r\in\mathcal{R}^{g}_{-}}\left(u_{rg}^{L}+u_{rg}^{O}\right)PYU \tag{26}$$

Stock calculating constraints.

$$\mathbf{s.\ t.} x_{gt}^{O}=x_{g,t-1}^{O}+\sum_{s=1}^{S}\left(ONY_{gts}^{O}+ONU_{gts}^{O}\right)$$
$$+\sum_{s=1}^{S}\sum_{r\in\mathcal{R}_{s,t}^{in}} u_{rg}^{O}-\sum_{s=1}^{S}\sum_{r\in\mathcal{R}_{s,t}^{out}} u_{rg}^{O} \qquad \forall\, g,\, t>0 \tag{27}$$

$$[0.5cm]x_{gt}^{L}=x_{g,t-1}^{L}+\sum_{s=1}^{S}\left(ONY_{gts}^{L}+ONU_{gts}^{L}\right)$$
$$+\sum_{s=1}^{S}\sum_{r\in\mathcal{R}_{s,t}^{in}} u_{rg}^{L}-\sum_{s=1}^{S}\sum_{r\in\mathcal{R}_{s,t}^{out}} u_{rg}^{L}$$
$$+w_{g,t-1}^{L} \qquad \forall\, g,\, 0<t\le LP_g \tag{28}$$

$$x_{gt}^{L}=x_{g,t-1}^{L}+\sum_{s=1}^{S}\left(ONY_{gts}^{L}+ONU_{gts}^{L}\right)$$
$$+\sum_{s=1}^{S}\sum_{r\in\mathcal{R}_{s,t}^{in}} u_{rg}^{L}-\sum_{s=1}^{S}\sum_{r\in\mathcal{R}_{s,t}^{out}} u_{rg}^{L}$$
$$+w_{g,t-1}^{L}-w_{g,t-LP_g-1}^{L} \qquad \forall\, g,\, t>LP_g \tag{29}$$

$$[0.2cm]x_{g0}^{O}=\sum_{s=1}^{S} INX_{gs}^{O}+w_{g}^{O}, \qquad \forall\, g \tag{30}$$

$$x_{g0}^{L}=0 \qquad \forall\, g \tag{31}$$

Capacity on origins / needs on destinations constraints.

$$\sum_{g=1}^{G}\left(u_{rg}^{L}+u_{rg}^{O}\right)\le avgDEM_r \qquad \forall\, r \tag{32}$$

$$u_{rg}^{L/O}\le x_{g,dout_r,sout_r}^{L/O} \qquad \forall\, r,\, g \tag{33}$$

Business-related constraints.

$$u_{rg}^{L}+u_{rg}^{O}\le UPG_{gr_r,g}\times M \qquad \forall\, r,\, g \tag{34}$$

$$\sum_{g=1}^{G} w_{g}^{O}COS_g\le BUD \tag{35}$$

Other constraints.

$$f_{gt}^{L/O}=x_{gt}^{L/O}+\sum_{r\in\mathcal{R}_{t}^{use}} u_{rg}^{L/O} \qquad \forall\, g,\, t \tag{36}$$

$$\begin{aligned} w_{gt}^{L}&\in\mathbb{Z}_0^+ && \forall\, g,\, t\in\{0,\ldots,T-1\}\\ w_{g}^{O}&\in\mathbb{Z}_0^+ && \forall\, g\\ x_{gt}^{L/O}&\in\mathbb{Z}_0^+ && \forall\, g,\, t\\ u_{rg}^{L/O}&\in\mathbb{Z}_0^+ && \forall\, r,\, g\\ f_{gt}^{L/O}&\in\mathbb{Z}_0^+ && \forall\, g,\, t \end{aligned} \tag{37}$$

A3.2. Pricing and deployment plan model

Considering the indices and parameters, decision variables and the Capacity-Pricing Model presented in Section 2.2, the following adaptation was developed. The adaptation consists on an extension of the model (with the same decision variables), with the addition of two constraint groups. The main difference resides on the overall acquisition plan (aggregated for all locations), which is an input that comes from the MIP model presented above. This input will be mentioned as *Pw*. The additional constraints are:

$$\sum_{s=1}^{S} w_{gts}^{L} = Pw_{gt}^{L} \qquad \forall\, g,\, t \tag{38}$$

$$\sum_{s=1}^{S} w_{gs}^{O} = Pw_{g}^{O} \qquad \forall\, g \tag{39}$$

A.4. Complete tables of results

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.orp.2018.10.002.

References

[1] Auto Rental News. U.S. car rental revenue and fleet size comparisons 2005–20152015;Retrieved June 14, 2016; URL http://www.autorentalnews.com/fileviewer/2230.aspx.

[2] Gao P, Sha S, Zipser D, Baan W. Finding the fast lane: emerging trends in China's auto market. Tech. Rep.. McKinsey & Company; 2016. Retrieved November 3, 2016.

[3] McKinsey & Company. Leading in the 21st century: an interview with Hertz CEO Mark Frissora. 2013. Retrieved November 3, 2016; URL http://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/leading-in-the-21st-century-an-interview-with-hertz-ceo-mark-frissora.

[4] Pachon J, Iakovou E, Ip C. Vehicle fleet planning in the car rental industry. J Revenue Pricing Manage 2006;5(3):221–36. https://doi.org/10.1057/palgrave.rpm.5160041.

[5] You P-S, Hsieh Y-C. A study on the vehicle size and transfer policy for car rental problems. Transp Res Part E 2014;64:110–21. https://doi.org/10.1016/j.tre.2014.01.007. https://doi.org/10.1016/j.tre.2014.01.007

[6] Li Z, Tao F. On determining optimal fleet size and vehicle transfer policy for a car rental company. Comput Oper Res 2010;37(2):341–50. https://doi.org/10.1016/j.cor.2009.05.010. https://doi.org/10.1016/j.cor.2009.05.010

[7] Fink A, Reiners T. Modeling and solving the short-term car rental logistics problem. Transp Res Part E 2006;42(4):272–92. https://doi.org/10.1016/j.tre.2004.10.003.

[8] Guerriero F, Olivito F. Revenue models and policies for the car rental industry. J Math Modell Algorithms Oper Res 2014;13(3):247–82. https://doi.org/10.1007/s10852-013-9234-8. http://link.springer.com/10.1007/s10852-013-9234-8

[9] Steinhardt C, Gönsch J. Integrated revenue management approaches for capacity control with planned upgrades. Eur J Oper Res 2012;223(2):380–91. https://doi.org/10.1016/j.ejor.2012.05.047. http://linkinghub.elsevier.com/retrieve/pii/S0377221712004365

[10] McAfee PR, te Velde V. Dynamic pricing in the airline industry. In: Hendershott T, editor. Handbook on economics and information systems. 1. Elsevier; 2006. p. 527–67. http://vita.mcafee.cc/PDF/DynamicPriceDiscrimination.pdf

[11] Oliveira BB, Carravilla MA, Oliveira JF, Raicar P, Acácio D, Ferreira J, et al. Pricing for internet sales channels in car rentals. In: Barbosa-Póvoa A, de Miranda JL, editors. Operations research and big data: IO2015-XVII Congress of Portuguese Association of Operational Research (APDIO). Springer International Publishing; 2015. p. 139–47.

[12] Oliveira BB, Carravilla MA, Oliveira JF. Fleet and revenue management in car rental companies: a literature review and an integrated conceptual framework. Omega 2017;71:11–26. https://doi.org/10.1016/j.omega.2016.08.011. https://doi.org/10.1016/j.omega.2016.08.011

[13] Haensel A, Mederer M, Schmidt H. Revenue management in the car rental industry: a stochastic programming approach. J Revenue Pricing Manage 2012;11(1):99–108. https://doi.org/10.1057/rpm.2011.2. https://doi.org/10.1057/rpm.2010.52

[14] Madden T, Russell RA. Modelling the rental car pricing and relocation problem. Int J Revenue Manage 2012;6(3–4):246–57.

[15] Zhang A, Zhang Y. Airport capacity and congestion pricing with both aeronautical and commercial operations. Transp Res Part B 2010;44(3):404–13. https://doi.org/10.1016/j.trb.2009.09.001. https://doi.org/10.1016/j.trb.2009.09.001

[16] Wang JYT, Yang H, Lindsey R. Locating and pricing park-and-ride facilities in a linear monocentric city with deterministic mode choice. Transp Res Part B 2004;38(8):709–31. https://doi.org/10.1016/j.trb.2003.10.002.

[17] Bajwa N, Sox CR, Ishfaq R. Coordinating pricing and production decisions for multiple products. Omega 2016;64:86–101. https://doi.org/10.1016/j.omega.2015.11.006. https://doi.org/10.1016/j.omega.2015.11.006

[18] Ghoniem A, Maddah B. Integrated retail decisions with multiple selling periods and customer segments: optimization and insights. Omega 2015;55:38–52. https://doi.org/10.1016/j.omega.2015.02.002. https://doi.org/10.1016/j.omega.2015.02.002

[19] Kim SW, Bell PC. Optimal pricing and production decisions in the presence of symmetrical and asymmetrical substitution. Omega 2011;39(5):528–38. https://doi.org/10.1016/j.omega.2010.11.002. https://doi.org/10.1016/j.omega.2010.11.002

[20] Raidl GR. Decomposition based hybrid metaheuristics. Eur J Oper Res 2015;244(1):66–76. https://doi.org/10.1016/j.ejor.2014.12.005.

[21] Paes FG, Pessoa AA, Vidal T. A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. Eur J Oper Res 2016;256(3):742–56. https://doi.org/10.1016/j.ejor.2016.07.022. http://linkinghub.elsevier.com/retrieve/pii/S0377221716305598

[22] Gonçalves JF, Resende MGC. Biased random-key genetic algorithms for combinatorial optimization. J Heuristics 2011;17(5):487–525. https://doi.org/10.1007/s10732-010-9143-1.

[23] Ramos AG, Oliveira JF, Gonçalves JF, Lopes MP. A container loading algorithm with static mechanical equilibrium stability constraints. Transp Res Part B 2016;91:565–81. https://doi.org/10.1016/j.trb.2016.06.003.

[24] Hokama P, San Felice MC, Bracht EC, Usberti FL. A heuristic approach for the stochastic steiner tree problem. 11th DIMACS implementation challenge. 2014.

[25] Li X, Zhao Z, Zhang K. A Parallel Genetic Algorithm for Three Dimensional Bin Packing with Heterogeneous Bins. Proceedings of the 2014 industrial and systems engineering research conference. 20149780983762430. https://doi.org/10.14445/22312803/IJCTT-V17P108. arXiv:1411.4565

[26] Stefanello F, Buriol LS, Resende MGC. A biased random-key genetic algorithm for a network pricing problem. XLV simpósio Brasileiro de pesquisa operacional. 2013. p. 3346–657. http://www-usr.inf.ufsm.br/~stefanello/publications/SBPO_2013.pdf

[27] Rossi F, Beek PV, Walsh T. Handbook of constraint programming. Elsevier0444527265; 2006.

[28] Oliveira BB, Carravilla MA, Oliveira JF, Toledo FMB. A relax-and-fix-based algorithm for the vehicle-reservation assignment problem in a car rental company. Eur J Oper Res 2014;237(2):729–37. https://doi.org/10.1016/j.ejor.2014.02.018. https://doi.org/10.1016/j.ejor.2014.02.018

[29] Oliveira B.B., Carravilla M.A., Oliveira J.F.. Vehicle reservation assignment in car rental data. EUDAT B2share, v1. http://hdl.handle.net/11304/027d022c-2ff4-4ff3-b308-8d7ded27ced1. 2016.

[30] Oliveira B.B., Carravilla M.A., Oliveira J.F.. Capacity-Pricing Model: car rental instances. Mendeley Data, v1. https://doi.org/10.17632/g49smv7nh8.1. 2017b. 10.17632/g49smv7nh8.1.

[31] Toso RF, Resende MGC. A c++ application programming interface for biased random-key genetic algorithms. Optim Methods Software 2015;30(1):1–16. https://doi.org/10.1080/10556788.2014.890197.

[32] Wagner S, Brandt T, Neumann D. In free float: developing business analytics support for carsharing providers. Omega 2016;59:4–14. https://doi.org/10.1016/j.omega.2015.02.011. https://doi.org/10.1016/j.omega.2015.02.011