Article:

Martinez-Sykora, A, So, MC, Currie, CSM et al. (2 more authors) (2020) Optimizing Pricing and Packing of Variable-Sized Cargo. IMA Journal of Management Mathematics. dpaa021. ISSN 1471-678X

https://doi.org/10.1093/imaman/dpaa021

Optimizing Pricing and Packing of Variable-Sized Cargo

A. MARTINEZ-SYKORA
CORMSIS
Department of Decision Analytics and Risk,
University of Southampton.
C.S.M. CURRIE
CORMSIS
Operational Research Group, Mathematical Sciences,
University of Southampton.
M.C. SO
CORMSIS
Department of Decision Analytics and Risk,
University of Southampton.
C. BAYLISS
Computer Science, Multimedia and Telecommunication Department,
Universitat Oberta de Catalunya
J.A. BENNELL
Leeds University Business School.

Abstract

Organizations have successfully used dynamic pricing to optimize revenues for many years, where research and practice have mainly focused on applications with independent, discrete commodities; for example, an airline ticket. In this research we consider applications where the commodity is continuous and the value of the commodity available to sell depends on the combination of previously accepted demand. We focus on vehicle ferries, where the accepted vehicle bookings are packed in lanes in the ferry to leave a useable space for future bookings. Certain combinations of vehicles may result in areas of unusable space, which will affect future revenue. While this application is the focus of the paper, there are numerous industries that face similar challenges including freight and the sale of advertising on television and radio. In this paper, we simultaneously solve the pricing and resource utilization problem to optimality for a discrete set of product types and stochastic demand. Our approach combines a dynamic pricing model with a mixed integer linear program to optimize the packing. We present results for real-world examples from the ferry industry and discuss extensions to the method to improve

the selection of vehicle configurations. dynamic pricing, revenue management, cutting and packing

1 Introduction

Many organizations have adopted variable pricing to maximize revenues. This is particularly common where the product is perishable and capacity is limited, such as for transport tickets. Our focus is on adapting dynamic pricing models to situations where the capacity used by a sale is a complex function of the individual capacity requirements of the sales accepted so far. Immediate applications arise where capacity is defined by either a continuous two (or three) dimensional constrained space, or time slots. The decision of how much to charge for a sale is then dependent on how efficiently the accepted sales can be arranged within the space/time available. The opportunity cost of fitting a large or awkwardly-shaped sale may well be greater than that associated with fitting several smaller sales which can generally be allocated more easily. Hence, for these problems we need to solve the underlying packing problem when setting prices to maximize revenue. The optimal solutions for each inventory level and each time period can be calculated well in advance of the selling season, enabling the longer time needed to calculate exact optimal solutions. We show here that incorporating sophisticated packing algorithms, in the form of mixed-integer programs into a dynamic pricing algorithm can increase revenues and improve the efficiency of the packing.

The application that we consider here is the pricing of vehicle tickets in the ferry industry, focusing on situations where the vehicles should be packed within lanes. In this case, the packing problem becomes a one-dimensional bin packing problem with heterogeneous bins. In previous work, Bayliss et al. (2018), we considered the physical space inside the ferry to be a two-dimensional space, ignoring lane configurations. This allows a more flexible placement of vehicles, and these earlier methods use heuristic algorithms to suggest the prices to charge and to address the two-dimensional packing problem. While providing efficient packing solutions, packing without consideration of lanes is slower and requires much more skill and careful placement on the part of the loaders. Consequently, splitting the physical space into lanes is a practical and widely used strategy. An additional advantage of the method we propose here is that we are aiming to solve the optimization problems exactly, trying to find the optimal prices to charge for each vehicle type at each point in time on the selling season.

The above problem description arises frequently in the transportation industry, where freight or vehicles are being transported on a scheduled service. Other applications where the remaining capacity is not a linear function of sales include selling advertising time on the radio and television (e.g., see Giallombardo et al. (2016)) or scheduling bespoke jobs on machines.

Maximizing efficiency is vital in the vehicle ferry industry to ensure the profitability of ferry services, which are often the only means of delivering freight and transporting people to and from island populations. The global ferry market was valued at over $15 billion in 2012, making it an important factor in the global transport network. Ferries often transport a wide range of vehicles types, ranging from motorcycles and private cars to large haulage trucks, which means that the packing problem is non-trivial and the reduction in capacity as a result of accepting a sale is not just dependent on the size and shape of the vehicle, but also how well it fits with the previously accepted sales. We consider ferries with multiple vehicle decks that can be configured in a variety of ways, hence adding the complication that the shape and size of the space available can be altered to provide a better fit for the mix of vehicles booked.

A dynamic program is used to find the optimal dynamic pricing policy for each vehicle type, where a vehicle type is defined by its physical dimensions. The states of the dynamic program correspond to the optimal packing solutions for each possible mix of vehicles that can fit on the ferry, where these are identified using a combination of two heuristics and a mixed-integer linear program (MILP). The heuristics, "first fit decreasing" and "minimum length", can speed up the computations involved in the MILP approach, particularly for very large examples, and are used to find packing solutions for vehicle mixes for which finding a feasible solution is relatively straightforward. We compare this combined approach with using either only the first fit decreasing or the minimum length heuristics to solve the packing problem. These produce inferior packing solutions and revenues, underlining the fact that it is important to solve the packing problem to optimality.

Using instances provided by a UK ferry operator, our results show that linking assignment and packing algorithms with dynamic pricing models can increase revenue by up to 65% compared with revenue management strategies that fix the maximum capacity of the number of vehicles of each type at the beginning of the selling period./

1.1 Problem description

For reasons of clarity we describe the problem in the context of a vehicle ferry while acknowledging its potential use in other industries. We aim to maximize the expected revenue generated by a vehicle ferry for a single origin and destination, where bookings are received during a finite selling horizon. Vehicles are classified by their dimensions (length, width and height) into a discrete set of vehicle types with known arrival rates and purchase probabilities, which can vary during the selling period, as described later in Section 6.1. This assumes that larger vehicles are more likely to pay a higher price and that willingness-to-pay increases as the time remaining to departure decreases, a common assumption based on the observation that customers with a higher willingness-to-pay tend to arrive later in the selling season.

The decision variables are the prices being charged for each vehicle type given the set of vehicles previously accepted and the time remaining until departure. Prices are chosen from pre-defined discrete sets for each vehicle type. By integrating optimal pricing and packing algorithms, the resultant dynamic pricing policies will account for the efficiency with which different combinations of vehicles can be packed. Since the number of vehicles that fit is not a straightforward function of the total area that they take up, we identify all of the vehicles mixes that would fit in the ferry by applying optimal packing algorithms, which allows us to change the vehicle mix to be considered while solving the pricing problem to maximize the total revenue.

In an extension to the problem we consider a situation in which the configuration of decks does not have to be specified until the day of departure. We allow our method to choose dynamically between several layouts, using the union of the vehicle mixes available in each configuration as the states of dynamic program for the pricing problem. This is of particular interest in the ferry industry where it is common for vehicle ferries to have the facility to lower temporary decks from the ceiling, which increases the ferry's capacity for low height vehicles whilst decreasing the ferry's capacity for high vehicles.

Our aim is therefore to maximize the value of the space available when setting prices for different vehicle types. Note that, depending on the arrival rate of each vehicle type and their willingness to pay, it might be possible that the solution which maximizes the revenue is not maximizing the utilization of the ferry. Nonetheless, through dynamic pricing the proposed approach is likely to increase the probability of achieving mixes of vehicles which increase profit.

2 Previous Research

Revenue Management (RM) has been a vibrant area of research over the past 30-40 years (Talluri and Ryzin, 2004) and one where industry has actively engaged with researchers to exploit the methodologies arising from their research. Traditionally, RM focuses on finding the optimal price to charge for discrete items of inventory, with one of the best known examples being the sale of seats on an aeroplane (e.g., Belobaba, 1989; Littlewood, 2005). The airline industry has greatly benefited from successfully employing RM models, as have other industries with perishable products such as restaurants (Bertsimas and Shioda , 2003), cruise ships (Maddah et al., 2010), car rentals (Li and Pang, 2017).

Our objective is to maximize the value of the available space, which we achieve by encouraging (through price) the most profitable vehicle mixes. There are analogies to multi-product dynamic pricing (see Akcay et al., 2010), but where each product has different dimensions, and the optimal numbers of products to sell are unknown a priori. The majority of the research in this area focuses on air freight (see Slager and Kapteijns, 2004; Kleywegt and Papastavrou, 2001, 1998), where the key issue in these articles is that the dimensions of the packages are not known exactly at the time of the booking (e.g., Kasilingam, 1996; J.S.Billings, 2003; Amaruchkul et al., 2007). For this reason the objective of the optimization routines implemented is often to minimize overbooking rather than to optimize revenue. Air cargo RM has some additional complexities over the situation that we consider here, namely a weight constraint and a constraint on the number of container positions. These complexities force the model solutions to be somewhat simplified, e.g., by using standard weight-volume relationships or density values provided by historical data. For example, Kasilingam (1996) combines the capacity dimensions with some flexibility in the schedule, so that a specified delivery time per product is satisfied.

The methods used in air cargo RM include Markov Decision Processes (Han et al., 2010) and dynamic stochastic knapsack (Kleywegt and Papastavrou, 1998). Newsvendor models are also popular and Wong et al. (2009) propose a model based on a variant of the multi-item newsvendor model, which accounts for both the weight and the volume capacity constraints but does not consider the packing of the items, while

Zou et al. (2013) use a two-location newsvendor model to optimize overbooking for an airline operating multi-segment flights. A continuous-time stochastic control model to solve a RM problem with a general two-dimensional capacity with two types of demand can be found in Xiao and Yang (2010).

To the best of our knowledge, none of the existing algorithms designed to maximize revenues solve the implicit packing problem using the best approaches available in the literature in terms of computational time and solution quality. We consider situations where product sizes are known at the time of booking, which means that the packing problem can be considered when setting prices. During the selling season the packing problem is a feasibility problem which tells us which vehicle types can still fit onto the ferry. The packing solution becomes fixed once all of the capacity has been sold and the selling season has finished.

In this paper we show that this problem can be addressed *off-line* in some problems with realistic ferry sizes and with a comprehensive discretisation of vehicle types; in such cases it is possible to list all of the vehicle mixes that fit onto the ferry. These constitute the feasible states of the dynamic program. This allows us to determine the optimal price that should be offered to any type of customer arriving at any time period in any state. The two key aspects to take into account in order to apply the proposed methodology is to use some greedy heuristics to speed up the derived packing problem, which is needed to compute the states, and the use of a sensible discretisation of vehicles into several types depending on their dimensions in order to reduce the curse of dimensionality derived when solving the pricing problem.

We base our computational results on problems derived from real data from a ferry company that operates with ferries that can accommodate around 214 standard-sized cars. It is worth mentioning that the largest ferries used in the industry have a capacity over 800 vehicles, and in this case the derived pricing problem will be much bigger and the number of vehicles type should be reduced to two or three in order to apply this method. With large numbers of vehicle types, the problem will become intractable due to the number of states and only heuristics or approximation algorithms would work. For these larger problem sizes, the heuristic method we describe in Bayliss et al. (2018) is generally a good option, although note that it differs from the method we describe here in that it is not an exact optimization so is not guaranteed to find the optimal solution.

Our work generalizes the problem presented in Kleywegt and Papastavrou (2001), who solve to optimality the dynamic and stochastic knapsack problem, in which each demand requests the same amount of a given resource and the decision to be made is to accept or reject the request. The more general case in which demands require different amounts of a resource is studied in Kleywegt and Papastavrou (1998). The method we describe here can be viewed as an extension of the dynamic stochastic knapsack approach, where accept/reject decisions are replaced by pricing decisions for regulating demand.

We do not consider the situation where multiple drop offs and pick ups are needed. When several origins and destinations are involved, then the problem becomes one of transportation planning, where freight is being transported between several origins and destinations across a network (e.g. Bartodziej et al. (2007)). An example with multiple drop offs can be found in the rail industry, where Crevier et al. (2012) proposed a model combining an integer programming model which address the logistics operations and maximizes the revenues. The authors propose a bilevel mathematical formulation in which railroad operations and RM are considered.

Besides the heuristic approach presented in Bayliss et al. (2018) and Bayliss et al. (2016), we have not found a previous research in the Operations Research literature on pricing for vehicle ferries. Maddah et al. (2010) consider dynamic pricing for passenger cruise ships, and focus on the problem of having multiple capacity constraints (cabins and lifeboats). Technically, a similar double capacity constraint exists in the vehicle ferry industry in that each ferry will have a limit on the number of passengers that can travel due to lifeboat capacity, which is separate from the space available on its vehicle deck. However, in our experience, the maximum number of passengers is rarely reached and so the packing of the vehicle deck is almost always the binding constraint.

We describe a dynamic pricing algorithm as opposed to a dynamic allocation algorithm. Hence, the decision is one of selecting a price from a finite set for each of the available products rather than whether to accept or reject an order. Dynamic pricing algorithms were introduced by Kincaid and Darling in the 1960s to describe the sale of goods in a shop (Kincaid and Darling (1963)), and their use in the RM literature has increased steadily since the publication of Gallego and van Ryzin (1997). Phillips (2005) provides a useful discussion of price-response models. A key component of a dynamic pricing algorithm is an understanding of the price sensitivity of the customers. We assume that customers are segmented by product types and use a logistic distribution to describe the willingness-to-pay. Bitran and Mondschein (1997) use a similar shape for the willingness-to-pay distribution for a retail pricing problem, but they use the Weibull distribution. Assuming a logistic distribution for the willingness-to-pay distribution means that a customer's price sensitivity is increased at prices close to the accepted market price; a good description of customer behavior in a competitive market. Other authors have used an exponential distribution to model acceptance probabilities (Gallego and van Ryzin (1994) and Zhao and Zheng (2000)). As is common in the transportation industry, we assume that willingness-to-pay increases as departure time approaches. Much of the early, pioneering research in RM exploited this feature of the airline industry by protecting seats for higher paying passengers (Littlewood (2005) and Belobaba (1989)).

In the problems we consider here we define the capacity as the number of vehicles of each type that can fit onto the ferry. Therefore, the capacity available depends on the layouts obtained by the packing algorithms used. Given that the capacity is a fixed two-dimensional area with height constraints, the combinatorial problem is that of finding a feasible arrangement of the items. In this case, the combinatorial optimization problem given by finding a feasible layout, under the typology of cutting and packing problems presented by Wäscher et al. (2007), is a Two-Dimensional Multiple Bin-Size Bin Packing Problem (2D-MBSBPP). A survey can be found in Lodi et al. (2002). Regarding exact procedures, de Carvalho (2002) defined an MILP formulation and Pisinger and Sigurd (2005) proposed a branch and price algorithm. Approximation algorithms can be found in Kang and Park (2003). Most recently, the metaheuristic algorithm proposed by Wei et al. (2013) obtains the best published results for this problem.

3 Model Formulation

3.1 Approach Overview

We solve the pricing problem to optimality using a dynamic program, where the states are defined by the numbers of vehicles of each type rather than the area used in the ferry. Note that the packing solutions for two different vehicle mixes with the same total area of vehicles can be very different in terms of the remaining useable space and consequently the ability to include additional vehicles. The price offered to a vehicle thus depends on how it fits in the ferry with the current set of booked vehicles and the expected demand in the remainder of the selling season, rather than just its dimensions.

In order to obtain the states of the dynamic program we enumerate all of the possibilities for the numbers of vehicles of each type which are able to fit on the ferry. This is achieved by formulating a multi-objective problem in which the objectives correspond to maximizing the number of vehicles of a particular type that can be fitted onto the ferry. This allows us to obtain a "Pareto set" of vehicle mixes, defined as the set of vehicle mixes for which adding one more vehicle of any type would result in the capacity being exceeded. Let m be the number of vehicles types, then any vehicle mix $s = \{s_1, \dots, s_m\}$, where s_i denotes the number of vehicles of type i, belongs to the "Pareto set" if and only if the following two conditions are satisfied.

1. All the vehicles in s fit onto the ferry.
2. Any vehicle mix $\overline{s}$ that satisfies the conditions: (i) $\overline{s} = \{\overline{s}_1, \dots, \overline{s}_m\}$ for which $\overline{s}_i \geq s_i \; \forall i \in \{1, \dots, m\}$ and (ii) there exists $i' \in \{1, \dots, m\}$ such that $\overline{s}_{i'} > s_{i'}$; does not fit onto the ferry.

We denote by Q the set of all feasible vehicle mixes, i.e, any vehicle mix $s = \{s_1, \dots, s_m\}$ satisfying condition 1. If we denote by R the "Pareto set" of vehicle mixes then $R \Subset Q$. In practice, $|Q|$ is much bigger than $|R|$.

Once the Pareto set has been identified, it is then straightforward to list all of the feasible vehicle mixes, which correspond to the states of the dynamic program.

Figure 1 gives an overview of the methodology. We describe the dynamic pricing formulation in Section 4, setting out the dynamic program, and describing the MILP and the construction heuristics used for assigning vehicles to lanes. There are two ways of approaching the pricing problem. In the first, a fixed allocation problem, we place a limit on the number of vehicles of each type that we accept and in the second, a dynamic allocation problem, we allow this to vary during the selling season based on realized demand. We present the fixed and the dynamic allocation problems in Section 4.2. The input to the pricing problem from the packing algorithms is the full set of vehicle mix states, which is described in Section 4.3 and the heuristic packing rules in Section 4.4.

Two versions of the bin packing formulation are considered: the first assumes products must be placed in lines (or lanes) and a second, more relaxed packing model, allows wider vehicles to straddle two adjacent lanes. The latter increases the ferry's capacity for larger vehicle types at the expense of smaller ones, which can be beneficial in some demand scenarios. As an extension to the basic method, in Section 5 we explain how these algorithms can be adapted to decide between alternative deck and lane

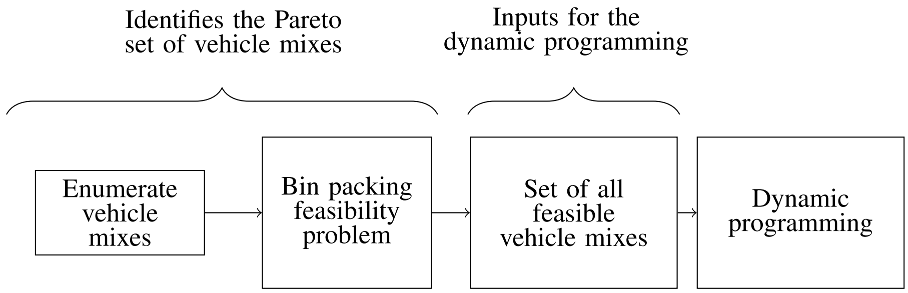

Figure 1: *Approach overview*

configurations on the ferry. Finally, in Section 6 we present the computational results for real and simulated instances coming from the vehicle ferry industry. We can solve to optimality some real-world instances with up to 5 vehicle types and ferries that can fit up to 214 standard-sized cars and our results show the benefits of keeping the allocation of space to each product type flexible compared with fixing these allocations in advance. We also demonstrate the value of incorporating optimal packing algorithms when solving the dynamic pricing problem.

It is worth highlighting that the computational time to solve these problems in a standard desktop computer could go up to 33 hours (see Section 6) and the memory needed to run the algorithm is around 15 GigaBytes. Since we are solving the off-line problem this computational effort is totally affordable by the companies and the ferry operator could let the algorithm run for days or even weeks. Note that the output of the run will be the complete pricing plan and there is no extra calculation to be made during the selling season.

4 Model Formulation

We assume initially that the available space is split into lanes and each vehicle can be assigned to a lane. This reflects standard practice, where lanes are used in order to reduce the loading time. The choice of how to configure the lanes is considered later in the article (Section 5). We begin with the most basic example, where lanes are fixed and no overlap is allowed, but we do go on to consider a situation in which products are allowed to overlap lanes in Section 4.4.4.

We begin by defining notation and presenting the dynamic programming formulation, which is solved to optimality. Following the dynamic program, we present two heuristics and one MILP model used to identify all of the feasible states. In the computational experiments, we implement the two constructive packing heuristics, first fit and minimum length, on their own to generate the states of the dynamic program. This allows us to evaluate the benefit of the computationally intensive optimal MILP approach. As part of our exact method, these heuristics are computationally useful for quickly identifying the feasibility of many vehicle mixes, which helps to compute the Pareto set with a much lower computational effort by reducing the number of times the MILP is solved. The MILP model is required to solve the packing problem to optimality when it is more difficult, i.e, the ferry is closer to capacity, and the heuristics fail to provide a feasible solution. We also extend this model in the subsequent sections in order to consider situations where wide vehicles are allowed to overlap adjacent lanes.

4.1 Notation

The selling season is divided into T time periods, where $t \in \{1, \dots, T\}$ is the number of time periods remaining until the end of the selling season. The selling season starts at $t = T$ and ends at $t = 1$, which corresponds to the last time period in which a customer might arrive. We set the length of each time period to be sufficiently small such that the probability of more than one arrival occurring in each period is negligible. Through the rest of the article, we assume that customers arrive following a time-homogeneous Poisson process, where rates differ between vehicle types.

We assume that the space can be split into lanes where $J = \{1, \dots, n\}$ denotes the set of lane types available, in which each lane type j, $j \in \{1, \dots, n\}$ is described by its maximum available height, $\hat{h}_j$, and width, $\hat{w}_j$, and its total length, $\hat{l}_j$. The number of lanes available for each type $j \in J$ is denoted by n_j. For the sake of simplicity, in this section we assume that the number of lanes of each type is known and fixed. In Section 5 we extend this formulation in order to consider different configurations of the space.

Let $I = \{1, \dots, m\}$ be the set of vehicle types. Each vehicle type $i \in I$ has width w_i, length l_i and height h_i and a known present demand, Z_i, which represents the number of customers already booked onto the ferry. In addition, there is a probability λ_i of a vehicle of type $i \in I$ arriving during a given time period.

4.2 Dynamic Programming Formulation

For each time period $t \in 1, \dots, T$, there is a price vector $p_t = (p_t^1, \dots, p_t^m)$ that gives the price on offer for each vehicle type $i \in I$. The values of p_t^i are discrete and limited to a predefined set of prices, $p_t^i \in \{Y_1^i, \dots, Y_{q_i}^i\}$ $\forall t \in \{1, \dots, T\}$, where q_i is the number of prices to be considered for vehicle type i. Note that it is possible to find the continuous optimal prices, but in a real world context discrete sets of allowable prices are more realistic. We assume that $Y_1^i < Y_2^i < \dots < Y_{q_i}^i$.

We define $\alpha_t(i, p_t^i)$, $t \in \{1, \dots, T\}$, $i \in I$, to be the probability that a customer with a vehicle type i accepts price p_t^i. It is worth highlighting that these $\alpha_t(i, p_t^i)$ only depend on the price of the current vehicle type (i), and not on the prices of other vehicle types. We are not assuming any condition on the acceptance probability but in the computational results we use a sigmoidal distribution, as defined in Section 6.1. We assume that $\alpha_t(i, Y_{q_i}^i) = 0$.

Let $Z = (Z_1, \dots, Z_m)$ be a utilization vector, where each entry Z_i denotes the number of bookings received for vehicle type i. We need to keep track of which vehicle types have been booked instead of simply the remaining available space because the mix of vehicles will impact on the ease with which additional vehicles can be fitted in. We consider two formulations: the fixed allocation problem and the dynamic allocation problem.

For the *fixed allocation problem* we assume a fixed capacity for each vehicle type at the start of the selling season and the state variable is bounded by $\overline{Z} = (\overline{Z}_1, \ldots, \overline{Z}_m)$, where the $\overline{Z}_i$ denotes the number of spaces available for vehicles of type i. Given a time period $t \in \{1, \ldots, T\}$, a vehicle type $i \in I$ and a state variable Z, we define a state-dependent action space as

$$P^1_{Zti} = \{p^i_t \in \{Y^i_1, \ldots, Y^i_{q_i}\};\ p^i_t = Y^i_{q_i} \ \text{ if } Z_i = \overline{Z_i}, i = 1, \ldots, m\}.$$

For the *dynamic allocation problem* we assume the state variable Z has to be within the *capacity envelope* of vehicle mixes, where the capacity envelope defines the set of feasible vehicle mixes. This formulation allows for dynamic allocation of capacities to vehicle types during the selling period and consequently is able to react to arrivals. For example, in the case where the number of bookings of one vehicle type is much higher than expected, we could accept all of them if there is space in the ferry. In the dynamic allocation problem we define a state-dependent action space as

$$P^2_{ZTi} = \{p^i_t \in \{Y^i_1, \ldots, Y^i_{q_i}\};\ \alpha_t(i, p^i_t) = 0\ \forall p^i_t \in \{Y^i_1, \ldots, Y^i_{q_i}\} \text{ if } Z + e_i \notin Q\}.$$

Note that for any vehicle mix in the Pareto front (R), it will not be possible to fit a new arrival, regardless of its vehicle type. The action space P^2_{ZTi} will then set the prices in such a way that the probability of acceptance is 0 for all the vehicle types.

The dynamic programming equations can be written as follows, where $V(Z, t)$ denotes the optimal expected revenue from period t to the end of the selling season (revenue-to-go).

$$\begin{aligned} V(Z, t) = \textstyle\sum_{i=1}^m \max_{p^i_t \in P_{ZTi}} &\left\{\lambda^i \left[\alpha_t(i, p^i_t)\left(p^i_t + V(Z + e_i, t-1)\right) + \left(1 - \alpha_t(i, p^i_t)\right) V(Z, t-1)\right]\right\} \\ &+\lambda^0 V(Z, t-1), \quad \forall t = 1 \ldots T,\ \forall Z \in Q, \end{aligned} \tag{4.1}$$

where $\lambda^0 = 1 - \sum_{i=1}^m \lambda^i$ is the probability that no arrival occurs. The utilization vector moves to $Z + e_i$ if vehicle type i purchases a ticket, where e_i is the unit vector of length m with zeros in all of the entries except a one at position i. The action space P_{ZT} can be either P^1_{ZT} or P^2_{ZT} dependent on whether the fixed or dynamic allocation problem is being solved.

The boundary condition, which captures the effect that the value of remaining ferry capacity perishes at the end of the selling season, is given by

$$V(Z, 0) = 0 \quad \forall Z \in Q. \tag{4.2}$$

A customer arrives into the system to buy space for a vehicle of size i with probability λ^i and will purchase with probability $\alpha_t(i, Y^i_{r_i})$. Purchases yield an immediate benefit of p^i_t but increase the utilization vector to $Z + e_i$

4.3 Optimizing the Fixed Allocation Problem

The first case described above considers a fixed booking limit for each vehicle type, where the maximum number of vehicles of type $i \in I$ that can be accepted onto the ferry is $\overline{Z}_i$. The way the values $\overline{Z}_i$ are selected has a strong effect on the final expected revenues. It is easy to check that if two vectors $\overline{Z}^1 = (\overline{Z}^1_1, \ldots, \overline{Z}^1_m)$ and $\overline{Z}^2 = (\overline{Z}^2_1, \ldots, \overline{Z}^2_m)$ satisfy $\overline{Z}^1_i \geq \overline{Z}^2_i$, $\forall i \in I$ then $V(\overline{Z}^1, T) \geq V(\overline{Z}^2, T)$, i.e, the expected revenues obtained by solving (4.1) using $\overline{Z}^1$ are greater than or equal to the expected revenues when using $\overline{Z}^2$. Therefore, at least one capacity vector $\overline{Z}$ belonging to the Pareto set will lead to the highest expected revenues from all of the feasible vectors. In the computational experiments presented in Section 6 we focus only on solutions in the Pareto set, where we report the best, average and worse expected revenues, allowing us to report the upper bound and the worst case revenues for the Fixed Allocation Problem.

In Section 4.4.3 we present a method to compute all of the vehicle mixes in the Pareto set. For each vehicle mix we then solve equations (4.1) to compute the expected revenues and choose the vehicle mix with the highest expected revenue to be the optimal fixed allocation policy.

4.4 Optimizing the Dynamic Allocation Problem

In the dynamic allocation problem we no longer fix the number of each vehicle type independently but instead ensure that the accepted vehicle mix lies within the capacity envelope. This requires us to define the capacity envelope, or equivalently the Pareto set that forms its boundary, in advance. This can be computationally expensive and so we begin by defining two heuristics than can speed up the process.

4.4.1 First Fit Heuristic.

The FF heuristic applied to the one dimensional bin packing problem consists of placing one item at a time in the first bin (lane) that it will fit. The complexity of this algorithm is $O(n \log n)$ if a search balanced binary tree is used to find the bin to insert the next piece, see D. Knuth (1998), where n is the number of vehicles. Note that different initial permutations of the vehicles lead to different solutions and it is well documented that in order to obtain high quality results using the FF heuristic it is better to sort the vehicles by non-increasing length. We assume that there are different lane types, each with different constraints on height and width, and so the order in which the lanes are considered will also impact the quality of the fit. As there are two different capacity constraints to be satisfied, the width and the height, the size of the lane cannot be used as a dominance criteria between lanes. We instead sort the lanes in such a way that the first lane considered has the lowest total arrival rate of the vehicle types that fit in the lane. That is, for each lane type $j \in J$, let J^j be the set of lanes of type j and let $I^j \subset I$ be the set of vehicle types such that any vehicle $i \in I^j$ fits in lane j. We then calculate the sum of the arrival rates of all the vehicles types that can fit in that lane,

$$a_j := \sum_{i \in I^j} \lambda_i,$$

and then sort the lanes by non increasing a_j. In order to break ties, we place vehicles in the lane with the shortest available length.

4.4.2 Minimum Length Placement Heuristic.

Using the ML heuristic, the vehicles are placed with the objective of maximizing the minimum length available within a given type of lane. We first assign vehicles to the lane type as in the FF algorithm, where we select the lane type with lower a_j. The particular lane is then chosen to be that with the greatest length remaining.

The ML heuristic is presented in Algorithm 1.

Algorithm 1 ML placement heuristic. Returns true if the vehicles fit in the ferry.

```
Input: Number of vehicles of each type (m_i, i ∈ I, set of lanes in the ferry J)
Sort vehicles by non increasing length
Sort the lanes by non-increasing a_j;
for each i = 1, ..., m do
    for each i' = 1, ..., m_i do
        Let j be the lane type with lowest a_j such that the vehicle type i fits
        Let k be the lane with the most space over all lanes of type j, k = 1, ..., |J_j|.
        Place vehicle i' of type i in lane k of type j
        if vehicle fits then
            Update space available in k
        else
            Return false
Return true
```

4.4.3 Exact Model.

We introduce a Mixed Integer Linear Program, MILP1, which is used to check the feasibility of adding an additional vehicle to the previously-booked vehicle mix. Although the objective of MILP1 is to check feasibility, we use an objective function that maximizes the unused space in each lane type. This allows us to reduce the number of times we need to solve the model. If a vehicle mix is found to be feasible, we will know the optimal lane allocation for each vehicle and the space available in each lane; therefore, in the next iteration, we can determine if a given vehicle fits in a lane without solving the MILP again. This can be achieved by applying either the FF or ML heuristics, using the previous MILP1 solution as an initial solution. The objective function is chosen to set priorities on which lanes to use, to avoid issues with symmetries, thus eliminating the potential for unnecessary computation, and guaranteeing that the space remaining will decrease each time a new vehicle is considered.

Let x_{ijk} be an integer variable representing the number of vehicles of type i assigned to lane k of type j. Decision variables are only generated for feasible vehicle allocation decisions, which is when $w_i < \hat{w}_j$ and $h_i < \hat{h}_j$. Let y_{jk}, $j \in 1, \ldots, n$, $k \in J_j$ be a binary variable which takes the value 1 if lane k of type j is used and 0 otherwise. Coefficients c_{jk}, $j \in 1, \ldots, n$, $k \in J_j$ satisfy the constraint that $c_{jk} < c_{jk'}$, $k < k'$ and $k' \in J_j$. These coefficients c_{jk} are used in the objective function in order to set the priorities on which lanes should be used first to help avoid symmetries.

In this model we assume that d_i is known, $\forall i \in I$, which matches with what is already packed. We define a real variable u, which represents the remaining usable length in the used lane with the most available space. We force coefficients $c_{jk} > \hat{l}_j$, $\forall j \in J, \forall k \in J_j$ giving more priority to determine which lanes we use first and then, as a secondary objective, to minimize u. The first term of the objective function ensures that we reduce the number of lanes used in the solution and that we use the most restrictive lanes first. The idea behind this objective is to reduce the number of times that we need to solve the MILP as this is time-consuming for practical-sized problems. For example, if we end up with two lanes not being used then, in the following iterations of the enumeration to find the Pareto front we use simple, fast heuristics since feasible solutions will be found easily. This reduces computation time significantly. The second term, u, is included for similar reasons: we know the space remaining in any lanes that are partially used. This again reduces the computational effort used on subsequent iterations of the enumeration.

MILP1 is then as follows:

$$\text{Min} \sum_{j \in J} \sum_{k \in J_j} c_{jk} y_{jk} - u$$

$$\sum_{j \in J} \sum_{k \in J_j} x_{ijk} = d_i \qquad \forall i \in I \tag{4.3}$$

$$d_i y_{jk} \geq x_{ijk} \qquad \forall j \in J, \forall i \in I \tag{4.4}$$

$$\sum_{i \in I} l_i x_{ijk} \leq \hat{l}_j \qquad \forall j \in J,\ k \in J_j \tag{4.5}$$

$$u \leq \hat{l}_j - \sum_{i \in I} l_i x_{ijk} - M(1 - y_{jk}) \qquad \forall j \in J,\ k \in J_j \tag{4.6}$$

$$x_{ijk} \geq 0 \text{ and integer} \qquad \forall j \in J,\ k \in J_j,\ \forall i \in I \tag{4.7}$$

$$u \in \mathbb{R} \tag{4.8}$$

Inequalities (4.3) ensures that the demand of all of the vehicles is met. Inequalities (4.4) activate an additional lane if it is used by any vehicle. Note that d_i is used as a big-M constant. Inequalities (4.5) ensure that lane lengths are not exceeded. Note that width and height are always met by the way the variables have been defined. Finally, inequality (4.6) forces us to use the lane with most available space sparingly. This forces the optimization to find solutions that fill the lanes up.

In order to obtain all of the solutions in the Pareto set we enumerate all of the potentially feasible combinations of the other vehicle types and then we try to find a feasible solution by applying first the First Fit Heuristic, then the Minimum length placement heuristic, and finally, in the case that the two heuristics fail to provide a feasible solution, we solve the MILP1 model. We start by setting all of the demands $d_i = 0$, $i \in I \setminus \{i'\}$ and, in each iteration, we increase the demand of one vehicle type by one and solve the model again, making sure all of the feasible combinations are considered. Note that when increasing the numbers of vehicles of any type in $I \setminus \{i'\}$ we might end up with an infeasible MILP1. In this case, the last feasible MILP1 solved to optimality provides a vehicle mix which belongs to the Pareto set.

It is important to highlight that an upper bound of the number of vehicles of each type that can fit onto the ferry is given by the space available on the lanes in which the given vehicle type fits, which generally avoid using the space available in the ferry for just one vehicle type.

4.4.4 Overlapping Lanes.

The optimization routines presented above assigns vehicles to lanes in such a way that all three dimensions of the vehicle should fit within the lane. In real applications, it is possible to place vehicles across more than one lane, which may be necessary for vehicles that are wider than any of the available lanes. While for ease of loading it is still important to respect the lanes boundaries as much as possible, this opens up the opportunity for placing more vehicles, as well as placing wider vehicles. Doing this will expand the capacity envelope of the ferry. Therefore, the formulation of MILP2, presented here, makes the lane width a soft constraint, allowing vehicles to be assigned to two lanes instead of one.

To allow vehicles to overlap lanes, first we need to identify which are the adjacent lanes. Let J'_j be the set of lanes of type j with an adjacent lane that can be used for the same vehicle. We identify each lane k of type j, that can be used simultaneously with an adjacent lane, which we denote by k' and has type j'. Note that consecutive lanes might be from either the same or different types.

The number of x_{ijk} variables thus increases and we define these variables if one of the following conditions are satisfied:

1. $w_i < \hat{w}_j$ and $h_i < \hat{h}_j, \forall j \in J,\ i \in I$
2. If $w_i > \hat{w}_j$, $w_i < \hat{w}_j + \hat{w}_{j'}$, $h_i < \hat{h}_j$ and $h_i < \hat{h}'_j,\ \forall i \in I,\ j \in J,\ k \in J'_j$, where k' is a lane of type j', which is adjacent to lane k of type j.

In addition, we introduce a new set of integer variables, z_{ijk}, which represent the number of vehicles of type i assigned to lane k of type j which use lane k' of type j'. Again, these variables are defined only if the second condition is satisfied.

MILP2 is then as follows:

$$\text{Min} \sum_{j\in J}\sum_{k\in J_j} c_{jk} y_{jk} - u$$

$$\sum_{j\in J}\left(\sum_{k\in J_j} x_{ijk} + \sum_{k'\in J'_j} z_{ijk'}\right) = d_i \qquad \forall i \in I \tag{4.9}$$

$$M y_{jk} \geq x_{ijk} \qquad \forall j \in J, \forall i \in I,\ \forall k \in J_j \setminus J'_j \tag{4.10}$$

$$M y_{jk} \geq x_{ij'k'} + z_{ijk} \qquad \forall j \in J, \forall i \in I,\ \forall k \in J'_j \tag{4.11}$$

$$\sum_{i\in I} l_i x_{ijk} \leq \hat{l}_j \qquad \forall j \in J,\ k \in J_j \setminus J'_j \tag{4.12}$$

$$\sum_{i\in I} l_i (x_{ij'k'} + z_{ijk}) \leq \hat{l}_j \qquad \forall j \in J,\ k \in J'_j \tag{4.13}$$

$$u \leq \hat{l}_j - \sum_{i\in I} l_i x_{ijk} - M(1 - y_j) \qquad \forall j \in J,\ k \in J_j \tag{4.14}$$

$$x_{ijk} \geq 0 \text{ and integer} \qquad \forall j \in J,\ k \in J_j,\ \forall i \in I \setminus \{i'\} \tag{4.15}$$

$$u \in \mathbb{R} \tag{4.16}$$

The objective function is the same as in MILP1. Equalities (4.9) ensure that all the vehicles are placed. Inequalities (4.10) and (4.11) force activation of a lane if a vehicle has been assigned to it. Inequalities (4.12) and (4.13) ensure that the lengths of the lanes are not exceeded, and inequalities (4.14) force the use of lanes with more space available, encouraging placement of the vehicles only in one lane if possible.

A further constraint is that the solution obtained by MILP2 should be achievable in practice, in that the layout can be produced by loading the vehicles one at a time. The challenge is to ensure that everything loaded in an adjacent lane is loaded in such a way that vehicle types assigned to two lanes and vehicles types assigned to one lane can be arranged, and that vehicles assigned to two lanes should have the same position in both lanes. To do this, we place the vehicles in the following order. First place vehicles using two lanes lanes, k and $k+1$, where k is an odd number, then place all the vehicles assigned to only one lane and finally place vehicles using two lanes, k and $k+1$, where k is an even number.

Figure 2 illustrates an example of a ferry with four lanes and 12 vehicles. Assuming that the top lane is the first lane, we first load vehicle 1, 2 and 3 since they use two lanes where the first lane is odd. Then vehicles 4 to 11 are assigned to individual lanes. Finally, vehicle 12 is loaded, which is assigned to lanes 2 and 3. Note that this procedure will always lead to an achievable solution.

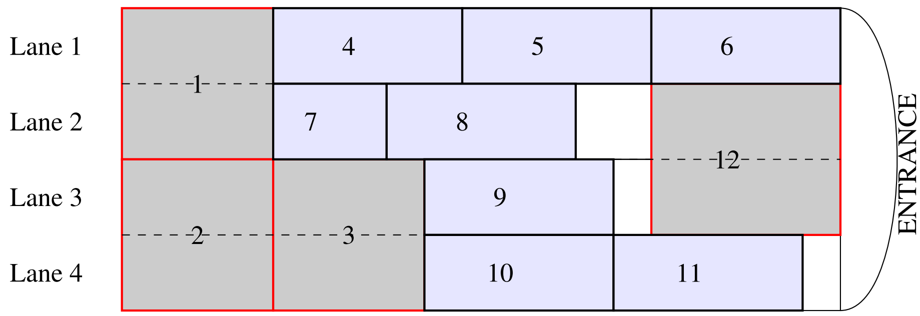

Figure 2: *Order of vehicles to be loaded in the ferry. There are four lanes and four vehicles requiring two lanes.*

5 Dynamic Configurations

In this section we generalize the model described in Section 4.2 to allow different lane configurations. This extension is motivated by work with vehicle ferries, many of which feature temporary decks that can be lowered to allow the placement of more small vehicles, as a result restricting the placement of high vehicles. This more general model extends the range of applications as it also allows the widths of the lanes to be optimized in order to satisfy different demand sets.

There are a finite set of configurations of the ferry's vehicle decks and a finite set of ways of configuring the lanes within those decks. Combining these we can predetermine a complete set of configurations. Hence instead of optimizing over a given configuration, here we optimize over them all.

Let $\phi \in \{1, \ldots, \Phi\}$ be Φ different lane configurations where the number of each lane type in configuration ϕ is denoted by $J^\phi = \{J_1^\phi, \ldots, J_{n_\phi}^\phi\}$ and n_ϕ is the total number of different lane types in configuration ϕ. Instead of making the decision on the best configuration to use in advance, we allow the decision to be made during the selling season. Therefore, in a given time period and depending on the set of vehicles to be placed in the ferry, the decision of the price offered to a new customer arises from considering all of the different configurations that can be used.

Let $t' \in T$ be the time period in which a new customer with a vehicle of type $i' \in I$ arrives. In order to decide whether the new vehicle fits, it is enough that the algorithms used in Section 4.4 identify a feasible solution for one configuration $\phi \in \{1, \ldots, \Phi\}$. If there are one or more configurations that cannot accommodate vehicle i' and the vehicle is accepted, then these configurations are not considered in the next time periods. For the dynamic ferry configuration formulation of the pricing problem, the capacity envelope is constructed from the union of the capacity envelopes for each individual ferry configuration. In practice we find that if selling a ticket to a given vehicle type leads to a reduction in the possible ferry configurations then this will in general lead to a lower expectation on future revenues. To offset this fall, a higher price would be offered to the customer.

In order to solve the DP equations presented in Section 4.2 to optimality, we calculate the value functions in all of the possible states obtained by all of the possible configurations in each time period.

6 Computational experiments

The computational results are generated using simulations based on the characteristics of real data from a vehicle ferry operator, with up to 5 vehicle types and a capacity of approximately 214 car equivalent units. We divide the computational experiments into four parts. In Section 6.1 we present the price acceptance model used in our experiments. In Section 6.2 we show the benefit of allowing dynamic capacity vectors when solving the pricing problem to optimality. In addition, we compare the three different methodologies we adopted in this paper to solve the derived packing problem: exact method, FF and ML heuristics. The results show that there is a significant benefit when the packing is solved to optimality. In Section 6.4 we analyze the results obtained for instances where the configuration of the ferry can be altered. We show that the algorithm allowing dynamic configurations described in Section 5 obtains significant improvements in expected revenue compared to the solutions obtained by solving each configuration separately. Finally, and based on the results obtained in Sections 6.2 and 6.4, we present results obtained when implementing the models on the largest real instances provided by a UK ferry operator.

The algorithms were coded in C++, MVS2013, and run on a i5-5300U CPU, with 2.30 GHz and 16 GB of RAM. The MILP models were solved using CPLEX, version 12.6.2.0.

6.1 Price acceptance model

This price acceptance model has been described elsewhere (Bayliss et al. (2018), Bayliss et al. (2016)) and is included here for completeness. The method will work with different functional forms for the price acceptance model, assuming that the key assumption that probabilities of purchase are independent of the prices on offer for different vehicle types is true.

We assume that the probability that a customer will purchase space for an item at price p, at time t before departure is equal to

$$\alpha_{t,p} = d\left(\frac{1}{1 + e^{k\left(\frac{p}{q}-f\right)}}\right)\left(a + (b-a)\left(1 - \frac{t}{T}\right)^c\right), \tag{6.1}$$

where $d = 1 + e^{-kf}$ is a normalising factor and a, b, c, k, f and q are parameters to be set for a particular problem instance.

The model has two multiplicative components, one for time and another for price. A logistic curve is used to model the price component, whilst a general non-linear model is used to capture the effect of time-until-departure on price acceptance. The parameter f controls the mid-point of the logistic curve and equivalently the skewness of the reservation price distribution. The choice of price-dependence suits the sale of goods in a competitive market because the demand elasticity is highest at prices close to the market price (Phillips, 2005). The parameter k controls the steepness of the sigmoidal price acceptance curve and represents the the inverse of the variance of the willingness-to-pay distribution.

Parameters a and b scale the probability of price acceptance at the beginning and end of the selling season respectively. The relative values of a and b result in three situations: (i) $b > a$ price acceptance increases over time (e.g. transportation); (ii) $a > b$ price acceptance decreases over time (e.g. fashion retailing); (iii) $a = b$ price acceptance is independent of time (e.g. durable goods). The value of c (> 0) accounts for any non-linear effects of time on the probability of price acceptance.

Since the parameters of this price acceptance model $\{a, b, c, k, f, q\}$ have intuitive meanings, the burden of fitting them to real data can be simplified. We suggest that a and b can be estimated from website click data; f is located at the mode of the willingness to pay distribution and k can be estimated from the average variance in price acceptance over time. The q parameter is the assumed upper limit on the price.

6.2 Data instances with one ferry

We have generated eight instances based on real data arising from two different ferry types and the number of vehicle types ranging between two and five. We specify below the ferry dimensions, the vehicle types and finally the parameters used to estimate the customer arrival rates and the price acceptance probability used in each time period. In all eight instances we assume $T = 1000$ time periods. Note that we assume that at most one arrival occurs during each time period. Therefore, we set the length of the time periods based on the arrival rates. If we assume that the arrival rate for each vehicle type is constant during the selling season then the time periods would have the same length, so for a typical selling period of six months the length of the time periods correspond to $6/1000 \approx 4.38$ hours.

Ferries: The first four instances were generated from a ferry which has six lanes in total, all with the same available length (37.04m). There is no restriction on the height but the lanes have different widths: two lanes with 2.34m, two lanes with 2.93m and two lanes with 3.42m to accommodate wider vehicles. We denote this ferry as RMF and consider four instances: RMF_2, RMF_3, RMF_4 and RMF_5, with two, three, four and five vehicles types respectively.

The second ferry, denoted as RFF, has one main deck, another top deck and two movable mezzanine decks which can be used when necessary. The mezzanine decks, when used, reduce the height available on the main deck in some areas from 4.9m to 2.7m (see Figure 3). Therefore, by using the mezzanine decks we reduce the capacity for higher vehicles in the ferry whilst increasing the capacity for lower vehicles.

We consider four instances based on this ferry, with 2, 3, 4 and 5 vehicle types, called RFF_2, RFF_3, RFF_4 and RFF_5.

Vehicles: Details of all five vehicle types are given in Table 1. The choice of categorization of vehicle types where the number of types is less than five is based on differentiating the most important groups. With two vehicle types we split between cars and lorries, types V2 and V5 such that the first vehicle type includes both V1 and V2, and the second V3, V4 and V5. With three vehicle types we add the cars with trailers and caravans, V2, V4 and V5. The classification with 4 vehicle types differentiates the smaller cars, V1, V2, V4 and V5.

Figure 3: *Decks and lanes specifications in the ferry*

When two or more groups are merged we combine their demands and use the dimensions of the larger group in the packing algorithms.

Table 1: *Vehicle types used in the computational experiments. Dimensions listed are the maximum allowed for each vehicle type. Ticks indicate if a vehicle category is being used. (λ_i)*

Vehicle	Width	Height	Length	RF2/RM2	RF3/RM3	RF4/RM4	RF5/RM5
V1	1.6m	1.5m	3m			✓	✓
V2	1.9m	1.5m	5m	✓	✓	✓	✓
V3	2.3m	2.5m	7m				✓
V4	2.9m	3m	9m		✓	✓	✓
V5	3.5m	4m	11m	✓	✓	✓	✓

Arrival rates and price acceptance model.

The arrival rates for each vehicle type vary between the instances and are summarized in Table 2. We consider high-demand scenarios in which the ferry is likely to fill up as this provides the best test of the methods described here.

The price acceptance probabilities for each customer type in each time period are given by the price acceptance model presented in (6.1). We set $c = 2$ to capture the effect that price acceptance probabilities increase at a faster rate as the time of depar-

Table 2: *Arrival rates (λ_i)*

Instances	V1	V2	V3	V4	V5	P(no arrival)
RMF2/RFF2	-	0.65	-	-	0.25	0.1
RMF3/RFF3	-	0.65	-	0.2	0.05	0.1
RMF4/RFF4	0.40	0.25	-	0.2	0.05	0.1
RMF5/RFF5	0.40	0.25	0.15	0.05	0.05	0.1

ture draws closer; $d = 0.5$ to capture a competitive market where price sensitivity is at its highest at the average market price; $k = 10$ is used to model a fairly wide variance of willingness-to-pay, which can be justified as considering a worst case scenario. Parameters $a = 0.5$ and $b = 1$ are used to model low levels of price acceptance at the beginning of the selling season, which increases as we approach departure. In this work we assume that the maximum price q_i that a customer for each vehicle type can pay is proportional to $\sqrt{l_i}$. This approximates the trend for larger (freight) vehicles to pay less per meter than smaller, more infrequent traffic. Freight vehicles tend to travel more frequently and expect a discount on the price per lane meter in return for a guaranteed minimum level of crossings. The other parameters of the model are assumed to be equal for all vehicle types.

Table 3 shows the solutions to the Fixed Allocation Problem (see Section 4.3) obtained for the RM instances. The first column shows the instance name, the second column the number of capacity vectors (non-dominated solutions when applying the algorithm from Section 4.3), and the third, fourth and fifth columns show, respectively, the average, minimum and maximum expected revenues obtained when solving Equation 4.1 for all of the capacity vectors. For instance, in instance RMF_5 we solve 3601 different problems. The values of the revenues compared between instances show the benefit of considering a better discretization for the vehicle types.

Table 3: *Number of possible solutions, average, minimum and maximum revenue for the Fixed Allocation Problem.*

Instances	#states	Av. R	Min. R	Max. R
RMF_2	34	30.76	18.76	44.77
RMF_3	252	31.52	19.11	44.02
RMF_4	532	32.27	21.87	47.63
RMF_5	3601	32.31	23.30	48.12

In Table 4 we present the solutions obtained for the Dynamic Allocation Problem (see Section 4.4), when applying the FF and ML heuristics and the exact model to decide whether a new vehicle fits in the ferry or the new customer should be rejected. Each of these approaches solves the dynamic pricing problem in Equation (4.1) to optimality. For each approach we report the total number of states considered, the expected revenues and the time needed to solve both the packing problems and the pricing problem. As expected, the number of states is always greater when solving the packing problem to optimality using the MILP (Exact). Between instances with four and five vehicles types, RM_4 and RM_5, we can observe that FF obtains a slightly higher number of states and expected revenues than ML.

For RMF_5, the computation time for the exact algorithm is 40% greater than for applying just the heuristics. However, note that we are solving the off-line problem, i.e, the policy price to offer each vehicle type in each state at each time period will be known before starting the selling season. Therefore, the computational time is not a strong restriction when solving this pricing problem.

Table 4: *Number of states, expected revenue and computational time in seconds for the Dynamic Allocation Problem.*

Instances	ML			FF			Exact		
	#states	Exp R	Time (sec)	#states	Exp R	Time (sec)	#states	Exp R	Time (sec)
RMF_2	252	60.27	1	254	60.35	1	256	60.47	4
RMF_3	2173	60.53	9	2293	60.79	9	2386	61.10	25
RMF_4	51577	77.57	1806	55484	77.32	1927	62771	79.00	2595
RMF_5	334816	78.04	12368	366644	78.07	13261	441378	79.60	17484

When comparing the results obtained for the Fixed Allocation Problem in which booking limits are predetermined before starting the selling season (Table 3), with those of the Dynamic Allocation Problem (Table 4), we can observe that the optimal fixed booking limits (Max. column in Table 3) produce expected revenues lower than the expected revenues obtained by any of the dynamic approaches. Note that the dynamic capacity approaches do not reject any customer if the vehicle fits, while the fixed booking limits strategy has fixed quotas for vehicle types and therefore may waste more space. From these results we can conclude that there is a considerable increase in the expected revenues (over 30%, and up to 65%) when solving the packing problem dynamically rather than fixing the capacities.

In Figures 4 and 5 we present the results obtained from $10,000$ runs of a simulation model for instances RM_2 and RM_3 respectively for the Dynamic Allocation Problem. Backing up the results from the dynamic programming, we can observe that with more vehicle types there is a greater difference between the heuristics and the exact algorithm. In both cases the ML and FF heuristics behave similarly, obtaining almost the same average and same shape for the revenue frequency distribution.

In Figures 6 and 7 we present different quantiles of the total length of the vehicles booked in each time period when solving the problem with the exact algorithm for instances RM_2 and RM_3 respectively. It is worth noting that in the first and in the last time periods, the total length range is in a much narrower interval, with capacity running out at the end of the selling season. It is interesting to observe that despite the somewhat greater variability in the middle of the selling season, the dynamic pricing ensures that the utilisation is in a relatively narrow range when sales close.

6.3 Interaction between packing and pricing

In Figure 8 we plot the future expected revenue for instance RMF_2, at the beginning of the selling season (left vertical axis) against the total length of vehicles (x-axis) for each of the possible states. This illustrates the interaction between packing and pricing: we see that it is possible for states which have a greater length of vehicles booked to

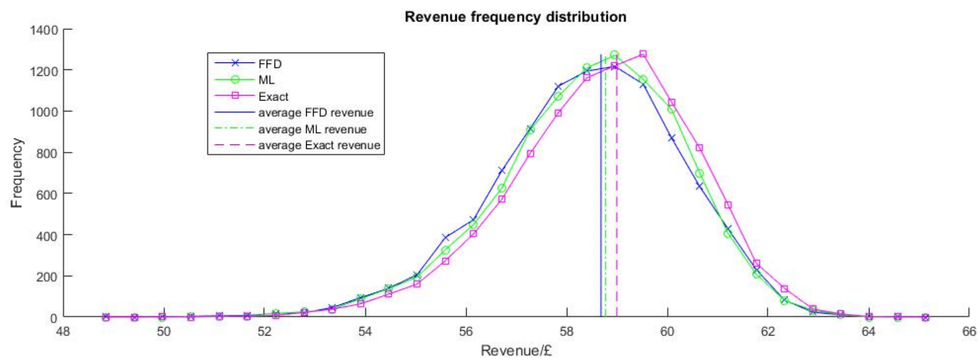

Figure 4: *Expected revenues with instance RM_2*

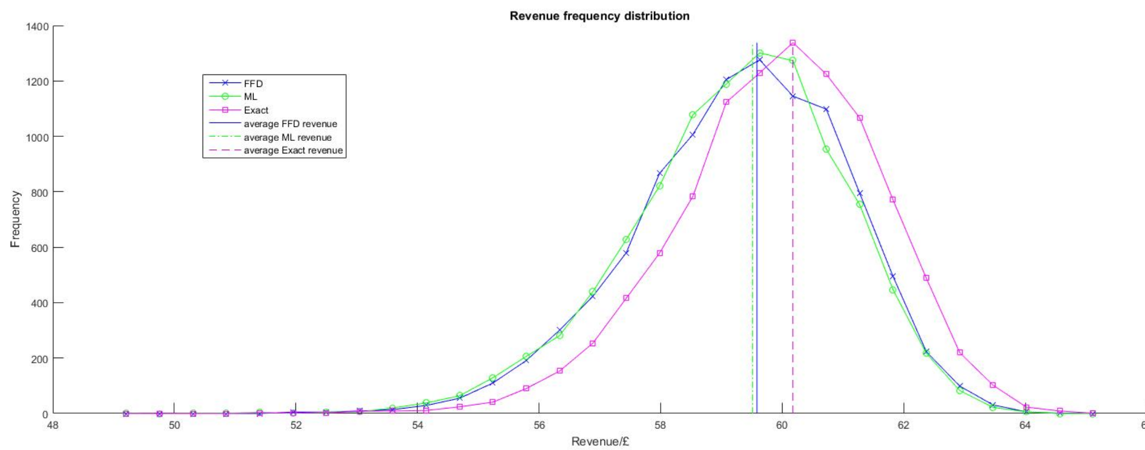

Figure 5: *Expected revenues with instance RM_3*

have higher expected future revenues than states with a shorter total length of vehicles booked. This is due to the efficiency of the packing: some vehicle mixes will be easier to pack efficiently than others, resulting in less wasted space. By taking account of the quality of the packing, the dynamic pricing algorithm will set prices for different vehicle types that reflect the ease of packing the resulting vehicle mix whilst also taking future expected demand into account. The effects of packing interactions become more pronounced as deck space runs out and the ferry nears capacity.

It is important to highlight here that the approach in which the MILP1 model is used finds all of the possible states, whereas the two packing heuristics may fail to find some states. This can result in lower revenues if the states that the heuristics do not find are likely to be used in practice.

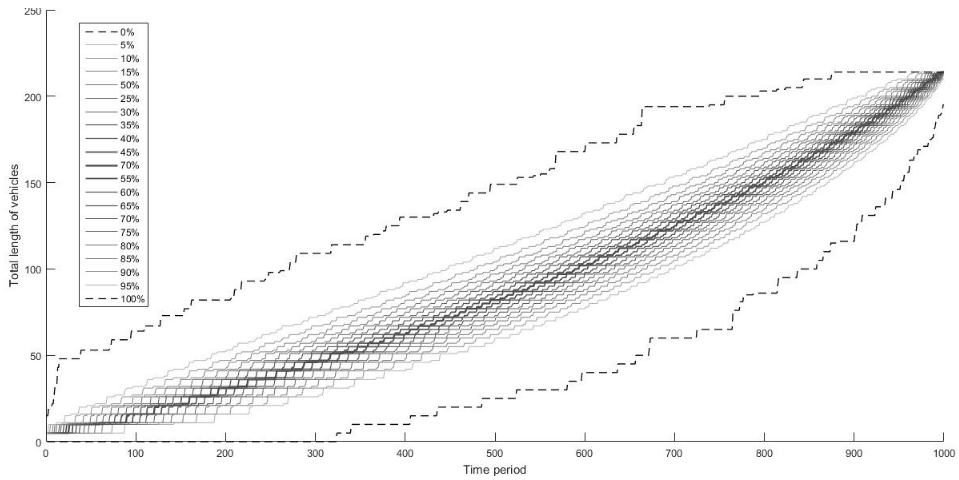

Figure 6: *Quantiles of the lengths of vehicles booked over time using the exact model for scenario RM_2*

6.4 Results for dynamic configurations

In this section we compare the results obtained when solving instances RMF_2, RMF_3 and RMF_4 with all of the 18 possible ferry configurations. We compare the exact algorithm (Section 4.4.3) and the dynamic configuration approach presented in Section 6.4 to assess the impact of the layout on the expected revenues.

We use the ferry defined previously and allow the width of the lanes to be 2m, fitting only two vehicles types; 3m, fitting the third vehicle type as well; or 4m, where all vehicle types can be placed. We consider 18 possible combinations of lanes of width 2m, 3m and 4m within the width of the ferry. The feasible non-dominated combinations are shown in Table 5.

Table 5: *Combination of lanes*

Combination	#2m	#3m	#4m	Combination	#2m	#3m	#4m
1	8	0	0	10	2	4	0
2	7	1	0	11	2	3	1
3	6	0	1	12	2	0	3
4	5	2	0	13	1	5	0
5	5	1	1	14	1	2	2
6	4	3	0	15	1	1	3
7	4	0	2	16	0	4	1
8	3	2	1	17	0	3	1
9	3	1	2	18	0	0	4

Table 6 presents the results obtained by applying the exact algorithm to each of the 18 configurations. The final row shows the result obtained by the Dynamic Configuration algorithm described in Section 5, where all of the possible ferry configurations are considered. We observe that the revenue is higher than for any of the fixed con-

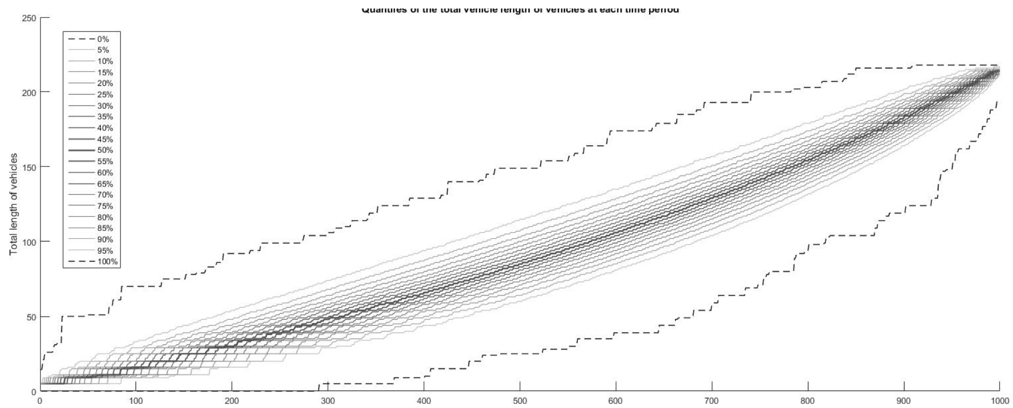

Figure 7: *Quantiles of the lengths of vehicles booked over time using the exact model for scenario RM_3*

figurations but, for each instance, there is a configuration which produces an expected revenue close to that for the Dynamic Configuration algorithm, e.g., configuration 2 for RMF_2. The reason being that this configuration matches the vehicle demand pattern well. The number of states varies between ferry configurations because the number of possible vehicle mixes will be different; and a higher number of states does not necessarily mean a greater expected revenue.

6.5 Larger Instances (RFF)

In Table 7, we present the solutions obtained for the larger instances derived from real data from a UK ferry operator. These are bigger than the instances used in the previous experiments. We apply both the ML and FF heuristics as well as the exact model described in Section 4.4.

The results obtained mimic those of Table 4 but with longer computation times. We enumerate more states when solving the packing problem to optimality using MILP1 than using the ML or FF heuristics and the additional states we find allow us to obtain greater expected revenues. This demonstrates that, for real-life problems, the quality of the packing solution used when optimizing the prices, has a definite beneficial effect on the revenues. Although we see a large increase in computational time for five vehicle types on this problem, it can still be solved using a reasonable amount of computational effort given that the problem is to be solved off-line.

7 Conclusions

We have described an exact optimization method that can be used for pricing different-sized products that must be packed into a three-dimensional space. The problem was simplified by assuming that the products have to be allocated to lanes, but clearly demonstrates through the numerical results that understanding how the products will

Table 6: *Expected revenues, number of states and computation times for the 18 ferry configurations and the Dynamic Configurations Algorithm (DC)*

Config	2 vehicle types			3 vehicle types			4 vehicle types		
	#states	Exp R	Time(sec)	#states	Exp R	Time(sec)	#states	Exp R	Time(sec)
1	85	52.59	1	85	43.83	1	6547	49.07	103
2	639	**62.06**	4	639	57.66	5	41415	60.89	589
3	973	61.95	9	973	**64.24**	10	51630	65.15	759
4	1087	59.53	8	1087	61.93	11	46196	63.49	835
5	981	56.19	8	981	57.67	11	34117	60.54	813
6	372	58.7	2	811	51.59	5	54463	56.07	729
7	576	61.12	3	2059	57.93	17	117501	61.85	1574
8	697	59.5	5	3420	61.24	39	161000	65.38	2273
9	735	56.19	6	4485	59.49	66	167489	63.37	2413
10	690	52	5	4845	54.86	52	136687	58.52	2109
11	788	61.87	5	2554	63.01	26	139662	66.28	1874
12	984	59.53	7	3417	63.92	31	147988	66.48	2099
13	960	56.19	7	3400	59.49	31	115457	62.77	2013
14	854	59.53	7	4255	63.58	49	190031	**67.13**	2625
15	912	56.19	6	4911	60.19	48	167212	63.37	2436
16	837	56.19	6	5333	60.15	63	187084	63.47	2713
17	737	52	5	5379	54.97	60	142573	58.52	2281
18	562	47.03	5	4091	48.93	53	85503	52.93	1798
DC	1338	**62.57**	12	8385	**64.71**	143	327599	**67.53**	5323

Table 7: *Expected revenue, number of states and computation times for real instances*

	ML			FF			Exact (MILP1)			Exact (MILP2)		
Instances	#states	Exp R	Time (sec)	#states	Exp R	Time (sec)	#states	Exp R	Time(sec)	#states	Exp R	Time(sec)
RFF_2	1031	118.22	8	1031	118.22	8	1031	118.22	10	2271	119.34	41
RFF_3	5189	120.32	57	5189	120.32	58	5189	120.32	70	25729	121.98	1081
RFF_4	47504	122.73	694	47749	122.73	724	130330	133.85	2894	215021	137.19	9261
RFF_5	600676	126.09	12292	600722	126.09	12600	888950	131.89	113472	898645	139.05	117402

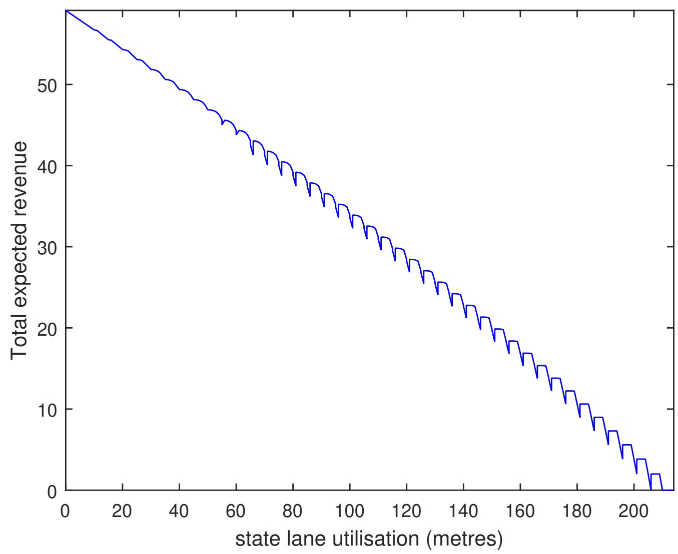

Figure 8: *Expected revenues from each state on instance RMF_2 (two vehicle types)*

be packed and providing good measures of the utilization of the available space during the selling season results in higher revenues. Fixing the individual booking limits for product types in advance is shown to be a poorer strategy with regard to revenue than being flexible about the numbers of each product type to accept. Accounting for the packing dynamically and allowing allocations to change dynamically based on realized demand can increase revenue by up to 65%.

The method we adopt involves allocating products to lanes of a fixed width and height using a bin-packing methodology. By introducing a soft width constraint in MILP2, we allow products to be assigned to two lanes if they exceed the width of one lane. This is something that we have observed in practice but have not seen modeled before. As we are still solving a one-dimensional bin-packing problem, the computational effort involved in solving this is not dramatically higher.

Allowing the decision over the configuration of the lanes to be flexible also improves the revenue results and allows the method to be used in situations where there is high variability in the numbers of different product types being bought during each time period. As the algorithm is designed to run offline, the computational time is not a strong constraint but we show in the final set of computational results that we can solve real-world instances in a reasonable length of time.

Acknowledgments: This work was funded by the EPSRC under grant EP/N006461/1. We are also grateful to Alexander Strong and Lee Hudson at Red Funnel, who provided us with practical advice when designing these methods, and to Professor Joerg Fliege at Southampton University, who kindly helped us with a very valuable input on the methodology and the scope of this research.

References

Y. Akcay, H.P. Natarajan, and S.H. Xu. Joint dynamic princing of multiple perishable products under consumer choice. *Management Science*, 56(8):1345–1361, 2010.

K. Amaruchkul, W. Cooper, and D. Gupta. Single-leg air cargo revenue management. *Transportation Science*, 41(4):457–469, 2007.

P. Bartodziej, U. Derigs, and M. Zils. O&D revenue management in cargo airlines - a mathematical programming approach. *OR Spectrum*, 29(1):105–121, 2007.

C. Bayliss, C.S.M. Currie, J.A. Bennell and A. Martinez-Sykora Dynamic pricing for vehicle ferries: using packing and simulation to optimize revenues. *European Journal of Operational Research*, 273(1):288-304, 2018.

C. Bayliss, J.A. Bennell, C.S.M. Currie, A. Martinez-Sykora and M. So A simheuristic approach to the vehicle ferry revenue management problem. *Proceedings of the 2016 Winter Simulation Conference*, pp:2335–2346, 2016.

P. P. Belobaba. Application of probabilistic decision model to airline seat inventory control. *Operations Research*, 37(2):183–197, 1989.

D. Bertsimas and R. Shioda. Restaurant Revenue Management. *Operations Research*, 51(3):472–486, 2003.

G. R. Bitran and S. V. Mondschein. Periodic pricing of seasonal products in retailing. *Management Science*, 43(1):64–79, 1997.

B. Crevier, J-F. Cordeau, and G. Savard. Integrated operations planning and revenue management for rail freight transportation. *Transportation Research Part B*, 46: 100–119, 2012.

J.M. Valerio de Carvalho. LP models for bin packing and cutting stock problems. *Eur. J. Oper. Res.*, 141:253–273, 2002.

Gallego G. and van Ryzin G. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40:999 – 1020, 1994.

G. Gallego and G. van Ryzin. A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research*, 45(1):24–41, 1997.

G. Giallombardo and H. Jiang and G. Miglionico. New Formulations for the Conflict Resolution Problem in the Scheduling of Television Commercials. *Operations Research*, 64(4):838–848, 2016.

D. Han, L. Tang, and H. Huang. A Markov model for single-leg cargo revenue management under a bid-price policy. *European Journal of Operational Research*, 200: 800–810, 2010.

J. Kang and S. Park. Algorithms for the variable sized bin packing problem. *European Journal of Operational Research*, 147:365–372, 2003.

R. Kasilingam. Air cargo revenue management: Characteristics and complexities. *European Journal of Operational Research*, 96:36–44, 1996.

W.M. Kincaid and D.A. Darling. An inventory pricing problem. *Journal of Mathematical Analysis and Applications*, 7:183–208, 1963.

A.J. Kleywegt and J.D. Papastavrou. Successive approximations for finite horizon, semi-markov decision processes with application to asset liquidation. *Operations Research*, 46:17–35, 1998.

A.J. Kleywegt and J.D. Papastavrou. The dynamic and stochastic knapsack problem with random sized items. *Operations Research*, 49(1):26–41, 2001.

Donald Knuth. *The Art of Computer Programming, 2nd Edition*. Addison-Wesley, USA, 1998.

L. Leemis. Nonparametric estimation of the cumulative intensity function for a nonhomogeneous Poisson process. *Management Science*, 37:886–900, 1991.

D. Li and Z. Pang. Dynamic booking control for car rental revenue management: A decomposition approach. *European Journal of Operational Research*, 256(3):850–867, 2017.

K. Littlewood. Forecasting and control of passenger bookings. *Journal of Revenue and Pricing Management*, 4(2):111–123, 2005.

A. Lodi, S. Martello, and M. Monaci. Two dimensional packing problems a survey. *European Journal of Operational Research*, 141:241–252, 2002.

B. Maddah, L. Moussawi-Haidar, M. El-Taha, and H.Rida. Dyanmic cruise ship revenue management. *European Journal of Operational Research*, 207:445–455, 2010.

R. L. Phillips. *Pricing and Revenue Optimization*. Stanford Business Books, California, 2005.

D. Pisinger and M. Sigurd. The two-dimensional bin packing problem with variable bin sizes and costs. *Discrete Optimization*, 2:154–167, 2005.

B. Slager and L. Kapteijns. Implementation of cargo revenue management at KLM. *Journal of Revenue and Pricing Management*, 3:80–90, 2004.

K. Talluri and G. Van Ryzin. *The Theory and Practices of Revenue Management*. Springer Science, New York, USA, 2004.

G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research, 183, 109-1130*, 2007.

L. Wei, W-C. Oon, and W. Zhu. A goal-driven approach for the 2d bin packing and variable-sized bin packing problems. *Discrete Optimization*, 224:110–121, 2013.

W.H. Wong, A. Zhang, Y.V. Hui, and L.C. Leung. Optimal baggage-limit policy: airline passenger and cargo allocation. *Transportation Science*, 43(3):355–369, 2009.

B. Xiao and W. Yang. A revenue management model for products with two capacity dimensions. *European Journal of Operational Research*, 205:412–421, 2010.

B.B. Yuen, J.S. Billings and A.G. Diener. Cargo revenue optimization. *Journal of Revenue and Pricing Management*, 2:69–79, 2003.

W. Zhao and Y. Zheng. Optimal dynamic pricing for perishable assets with nonhomogenous demand. *Management Science*, 46(3):375–388, 2000.

L. Zou, C. Yu, and M. Dresner. The application of inventory transshipment modeling to air cargo revenue management. *Transportation Research Part E*, 57:27–44, 2013.