Optimizing Pricing and Packing of Variable-Sized Cargo

Martinez-Sykora, A. a.martinez-sykora@soton.ac.uk
Bayliss, C. C.Bayliss@soton.ac.uk
Bennell, J.A. J.A.Bennell@soton.ac.uk
Currie, C.S.M. Christine.Currie@soton.ac.uk
So, M.C. m.so@soton.ac.uk

Sunday 25th March, 2018

Abstract

Dynamic pricing has been exploited by many organisation as a powerful strategy in revenue management. In general, research and practice has focused 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. These models assume the amount of resource needed to meet the demand of a given customer is known at the time the customer requests the booking. This is not the case for all industries. There are many applications where the amount of resource needed to meet the demand of one customer depends on how well it combines with the bookings of previously accepted customers. Booking requests that look attractive in isolation may combine poorly and lead to unusable resource therefore reducing revenues. Modelling these scenarios presents the additional challenge of valuing the opportunity cost of accepting a booking with respect to the remaining resource. A good example of such a scenarios arises in the vehicle ferry industry, where the resource is the physical space on the vehicle decks. While this application is the focus of the paper, there are numerous industries that face the same challenge including freight (air, sea and land) and airtime on television and radio, where the resource is time. In this paper we simultaneously solve the pricing and resource utilisation problem. Vehicle layouts in which the space usage is optimized can be modelled as a mixed integer linear program incorporating height and manoeuvrability constraints. We assume a discrete set of product types with stochastic demand. Our solution approach combines a dynamic pricing model, with packing heuristics to determine the best of a discrete set of fares to charge at different points in the selling period, enabling us to develop a multi-product dynamic pricing strategy over a finite selling period with no replenishment.We present results for real-world examples from the ferry industry and discuss extensions to the method to allow decisions to be made between different configurations of the available space.

Keywords: Dynamic pricing, packing problems, dynamic programming, integer programming

1 Introduction

Many organisations have adopted variable pricing to maximise revenues. This is particularly common where the product is time limited and fixed capacity, such as airline 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 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. We show that incorporating more 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 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. In this paper we demonstrate our methodology on the application of selling space on the vehicle decks of ferries.

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. The mix of vehicle sizes carried by ferries, ranging from motorcycles and private cars to large haulage trucks, means that 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 according to the mix of vehicles booked. A dynamic programming formulation is used to find the optimal prices to charge for each product on offer, where a product is defined by its physical dimensions. The packing problem can be solved to optimality using a mixed-integer linear program (MILP) but we also compare results for two heuristics: ”first fit decreasing” and ”minimum length”. The heuristics can speed up the computation, particularly for very large examples. We compare the different approaches using instances provided by a UK ferry operator. Our results show that linking assignment and packing algorithms with RM models can increase revenue by up to 65%.

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 on 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 in Section 4.1. 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.

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, which are a function of vehicle types and the remaining capacity on the ferry. Since the capacity it not a linear function of space, we generate the configuration of the vehicles on the ferry to determine the remaining capacity.

In an extension to the problem we consider a situation in which the configuration of decks can be modified up to the day of departure. We allow our method to choose dynamically which of several layouts would produce the greatest revenue. 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. These provide additional parking spaces for smaller vehicles but place a constraint on the height of vehicles that can be parked beneath them. The proposed approach will also determine the optimal deck configuration for any given instance of the selling season.

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 the willingness to pay a given price it might be possible that the solution which maximizes the revenues is not maximizing the utilization of the ferry, but the algorithm will seek to fill the ferry to capacity by varying the price.

1.2 Previous Research

Revenue Management (RM) has been a vibrant area of research over the past 30-40 years (see 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 also analogies to multi-product dynamic pricing (see Akcay et al. (2010)), where each product has different dimensions, but the optimal product capacities (the inventory level) are unknown a priori. The majority of the research in this area focuses on freight (see Slager and Kapteijns (2004), Kleywegt and Papastavrou (2001), Kleywegt and Papastavrou (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) and J.S.Billings (2003), Amaruchkul et al. (2007)). For this reason the objective of the optimization routines implemented is 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 relationship or density values provided by historical data. See Kasilingam (1996) for a further discussion, in which the capacity dimensions are combined with some flexibility for the schedule in such a way 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) proposed 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) used 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 and can therefore still be sold tickets. The packing solution becomes fixed once all of the capacity has been sold and the selling season has finished. Our work generalizes the problem presented in Kleywegt and Papastavrou (2001), who solved 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 off 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 model which address the logistics operations and maximizes the revenues. The authors proposed a bilevel mathematical formulation in which railroad operations and RM are considered.

We have found no previous research in the Operations Research literature on pricing for vehicle ferries. Maddah et al. (2010) consider dynamic pricing for 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 on the ferry 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. This approach effectively rejects requests by setting the price high enough that the customer has a low probability of accepting the price. Dynamic pricing algorithms were introduced by Kincaid and Darling (Kincaid and Darling (1963)) in the 1960s to describe the sale of goods in a shop, and their use in the RM literature has increased steadily since the publication of Gallego and van Ryzin (1997). 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. See Phillips (2005) for a useful discussion of price-response models. Bitran and Mondschein (1997) use a similar shape for the willingness-to-pay distribution for a retail pricing problem, but they used the Weibull distribution. Assuming a logistic distribution for the willingness-to-pay distribution means that a customer's price acceptance probability will follow a logistic curve distribution. This is a good description of price sensitivity in a competitive market as it captures the increased price sensitivity that occurs at prices close to the accepted market price. Other authors have used an exponential distribution to model acceptance probabilities, see 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 (see the static models of Littlewood (2005) and Belobaba (1989)).

In the problems we consider here the capacity available depends on the layout 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. We implement exact algorithms to evaluate the impact of optimizing the prices together with the packing and will also describe heuristic algorithms that we have developed to reduce the computational time.

1.3 Approach Overview

We solve the pricing problem to optimality using a dynamic programming model where the states are defined by the number of vehicles of each type in the solution rather than the area used in the ferry. We followed this approach to be able to capture the usability of the area left in the ferry. Also, note that two solutions with the same total used area can be very different in terms of the flexibility to allocate new vehicles. Therefore, by using the area as the state variable we would not be accounting for the wasted space that might result from including an additional vehicle. The price offered to a vehicle depends on how it fits in the ferry with the other vehicles rather than just the volume of the vehicle.

In order to obtain the states of the dynamic program we enumerate all the possibilities in terms of number of vehicles of each type. This is done by formulating this problem as a multiobjective problem, as many objectives as different types of vehicles, aiming to maximize the number of vehicles of each type. These objectives clearly conflict each other as placing more vehicles of a given type reduces the space in the ferry to place other types of vehicles. The exact Pareto front of this problem is obtained by solving to optimality a set of MILPs, or it is approximately solved by using heuristics such as the first fit algorithm and the minimum length algorithm. Once the Pareto front is identified it is then straightforward to list all the feasible solutions (the dominated ones), called vehicle mixes, which correspond to the states of the dynamic programming model.

Figure 1 depicts the progressong of the algorithms we describe here to produce the final methodology. We describe the dynamic pricing formu-

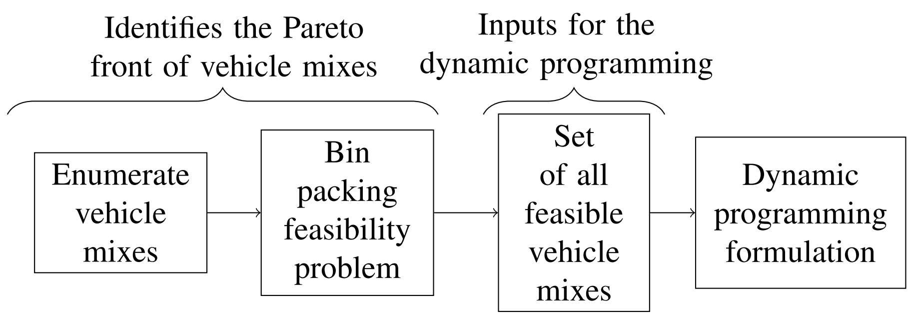

Figure 1: *Approach overview*

lation in Section 2, setting out the dynamic programming algorithm, before going on to describe the packing models and the construction heuristics used for assigning vehicles to lanes. We can solve to optimality real-world instances with up to 5 vehicle types and ferries which can fit up to 214 standard-sized cars. In conclusion, we discuss future lines of research to explore how to increase problem sizes that can be solved in a reasonable time.

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 realised demand. We present the fixed and the dynamic allocation problem in Section 2.2. Input from the packing algorithms takes the form of a Pareto frontier of vehicle combinations to capture the capacity of the space and also serve as the states of the dynamic programming formulation of the pricing problem. The determination of this Pareto frontier is described in Section 2.3 and the heuristic packing rules in Section 2.4.

Two versions of the bin packing are considered: the first assumes products must be placed in lines (or lanes) and the second allows wider products to overlap lines. As an extension to the basic method, in Section 3 we explain how these algorithms can be adapted to decide between space configurations. Finally, in Section 4 we present the computational results for real and simulated instances coming from the vehicle ferry industry. These show the benefits of keeping the allocation of space to each product type flexible compared with fixing these allocations in advance and also demonstrate the value of incorporating optimal packing algorithms when solving the dynamic pricing problem.

2 Model Formulation

We assume 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 3) and we also consider a situation in which products are allowed to overlap lanes in Section 2.4.4. However, the general rule is that products will be placed in lanes within the space given. We begin with the most basic example of this, where lanes are fixed and no overlap is allowed. Due to the decision to split the space into lanes with different dimensions, we can classify and solve the packing problem as a One Dimensional Multiple Bin Size Bin Packing Problem (1DMBSBPP) under the typology defined in Wäscher et al. (2007), where the bins represent the lanes.

We begin by defining notation and the function used to describe the price acceptance probability of a customer before going on to present the dynamic programming formulation, which is the same in all the approaches presented in the paper. Following the dynamic programming model we present two construction packing heuristics (first fit algorithms and minimum length), which are adapted to fit the application, to provide a fast decision as to whether a new vehicle can fit into the ferry or not. Alternatively, we use a MILP model to solve the packing problem to optimality. We extended this model in the following sections in order to consider vehicles that are wider than the width of the lanes by allowing them to overlap adjacent lanes.

2.1 Notation

The selling season is divided into T time periods, where $t \in \{1, \ldots, T\}$ is the number of time periods remaining until the end of the selling period. 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. In cases where arrivals follow a non-homogeneous Poisson process we can use procedures such as that of Leemis (1991) to define the length of each time period in the selling season such that the arrival rate is constant for each time period.

We assume that the space can be split into lanes where $J = \{1, \ldots, n\}$ denotes the set of lane types available, in which each lane type j, $j \in \{1, \ldots, n\}$ is described by the maximum available height ($\hat{h}_j$), width ($\hat{w}_j$) and 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 3 we extend this formulation in order to consider different configurations of the space.

Let I be the set of vehicle types. Each vehicle type $i \in I$ has dimensions width w_i, length l_i and height h_i and a known present demand d_i, which represents the number of customers already booked onto the ferry. In addition, each vehicle $i \in I$ has a probability λ_i of arriving during time period t.

2.2 Dynamic Programming Formulation

For each time period $t \in 1, \ldots, T$, there is a price vector $p_t = (p_t^1, \ldots, p_t^{|I|})$ that gives a price for each vehicle type $i \in I$. The values of p_t^i are discrete and limited to a predefined set of prices, $Y_1^i, \ldots, Y_{q_i}^i$, 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 $Y_1^i < Y_2^i < \cdots < Y_{q_i}^i$. Let $\alpha_t(i, Y_{r_i}^i)$ be the probability that a customer with a vehicle type i accepts price $Y_{r_i}^i$ in time period t.

Let $Z = (Z_1, \ldots, Z_{|I|})$ be a utilisation vector, where each entry Z_i denotes the number of bookings received for vehicle type i. We consider two formulations:

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

$$P_{ZT}^1 = \{p_t \geq 0;\ \alpha_t(i, Y_{r_i}^i) = 0\ \forall r_i = 1, \ldots, q_i \text{ if } Z_i = \overline{Z_i}, i = 1, \ldots, |I|\}.$$

For the *dynamic allocation problem* we assume the state variable Z is not bounded. This formulation allows for dynamic allocation of capacities to vehicles during the selling period and consequently is able to react to arrivals. Therefore, in the case where the number of bookings of one vehicle type is much higher than that expected we accept all of them if there is space in the ferry. In this case, we define a state-dependent action space as

$$P_{ZT}^2 = \{p_t \geq 0;\ \alpha_t(i, Y_{r_i}^i) = 0\ \forall r_i = 1, \ldots, q_i \text{ if the packing algorithm fails to pack vehicle } i \text{ and vehicles in Z}, \quad i = 1, \ldots, |I|\}.$$

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.

$$V(Z,t) = \max_{p_t \in P_{ZT}} \left\{ \left\{ \sum_{i=1}^{|I|} \lambda^i \left\{ \alpha_t(i, Y^i_{r_i}) \left[Y^i_{r_i} + V(Z', t-1) \right] + \left(1 - \alpha_t(i, Y^i_j) \right) V(Z, t-1) \right\} \right\} + \lambda^0 V(Z, t-1) \right\}, \tag{1}$$

where $\lambda^0 = 1 - \sum_{i=1}^{|I|} \lambda^i$ is the probability that no arrival occurs, Z' is the utilisation vector if vehicle type i buys and P_{ZT} can be either P^1_{ZT} or P^2_{ZT}. 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 $Y^i_{r_i}$ but increases the utilisation vector to Z', which satisfies $Z'_i \leq Z_i - 1$. If no sale occurs, then the capacity vector remains the same and the benefit is $V(Z, t-1)$. Where booking limits are flexible and no upper bounds are considered on the Z vector, we embed packing algorithms when solving equations (1) to determine whether it is feasible to accept a vehicle of type i.

2.3 Finding Optimal Fixed Booking Limits

Considering the first case described above of a fixed booking limit for each vehicle type, we describe here how fixed booking limits can be determined to optimize expected revenue. We begin by finding the non-dominated solutions to the packing problem on the Pareto frontier, where the number of each vehicle type placed in the ferry is a different objective, and run the dynamic programming routine with each of these to find the solution with the greatest revenue. This solution can be regarded as the optimal fixed booking limits. Note that a non-dominated solution will lead to the best expected revenues. If the solution is dominated then the expected revenues can be improved easily by adding at least one vehicle.

In order to generate the Pareto frontier of combinations of product vehicle types that fit onto the ferry we consider a multi-objective problem in which the maximization of one vehicle type is considered as a single objective. In order to find all the non-dominated solutions we iteratively formulate a single objective MILP which is solved to optimality. Each formulation fixes the number of vehicles of each type except one, and the objective is to maximize the number of the unfixed vehicle type placed in the space. In order to guarantee that we consider all the solutions in the Pareto Frontier we enumerate all possible combinations for the fixed number of each vehicle type, and we keep only the non dominated solutions. In what follows we present the ILP model used at each iteration, ILP1, and then we explain the iterative procedure used to obtain all the solutions in the Pareto frontier.

Let vehicle type $i' \in I$ be the vehicle which will be maximized, i.e, we fix the number of all the vehicle types in the ferry except vehicle type i', and set our objective to be the number of vehicles of type i' that we place. Let d_i be the number of vehicles of type $i \in I \setminus \{i'\}$ to be placed in the space. Let x_{ijk} be an integer variable which denotes the number of vehicles of type i assigned to lane k of type j. We define x_{ijk} only if vehicle type i fits in lane type j, i.e, $w_i \leq \hat{w}_j$ and $h_i \leq \hat{h}_j$. The ILP1 model can be formulated as follows.

$$\text{Max} \sum_{j=1}^{n} \sum_{k=1}^{|J_j|} x_{i'jk}$$

$$\sum_{j \in J} \sum_{k \in J_j} x_{ijk} = d_i \qquad \forall i \in I \setminus \{i'\} \tag{2}$$

$$\sum_{i \in I} l_i x_{ijk} \leq \hat{l}_j \qquad \forall j \in J,\ k \in J_j \tag{3}$$

$$x_{ijk} \in \mathbb{N} \qquad \forall 1 \leq j \leq n,\ k \in J_j,\ \forall i \in I \setminus \{i'\},\ \forall t \in \{1, \ldots, d_i\} \tag{4}$$

The objective function maximizes the number of vehicles of type i'. Equality constraints (2) force the algorithm to satisfy demand for the other vehicle types. Note that these values are fixed. Inequality constraints (3) ensure that the length constraints are satisfied.

In order to obtain all the solutions in the Pareto frontier we will enumerate the combinations of the other vehicle types and solve ILP1 for each, i.e, we start setting all 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 the 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 ILP1.

2.4 Dynamic packing

We now assume flexible booking limits during the selling season, which makes it necessary to decide whether a vehicle fits in the space on sale given the vehicles booked previously. Note that most of the time this problem is trivial, especially when the ferry is fairly empty. With the two constructive heuristic algorithms, the first fit decreasing (FF) and minimum length placement (ML), it is straightforward to place the next vehicle if it fits easily and uses little computational time. Solving the MILP is computationally more demanding. Therefore, we use constructive heuristics to speed up the calculations early in the selling period and the exact method to find the most difficult states, which the heuristics fail to obtain.

2.4.1 Firs 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 $O(n \log n)$, where n is the number of vehicles. Note that different initial permutations of the vehicles leads to different solutions. 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 $j \in J$, we calculate the sum of the arrival rates of all the vehicles types that can fit in that lane, $a_j = \sum_{i \in I,\ i \text{ fits in } j} \lambda_i$, 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.

In every time period in which a new arrival occurs we use the FF algorithm to pack all of the vehicles including the new arrival. If the FF algorithm is not able to find a feasible solution we do not consider that state in the given time period and the current arrival (customer) is rejected for the remaining time periods.

2.4.2 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 and the particular lane is 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 ($n_i,\ i \in I$, set of lanes in the ferry J)
Sort vehicles by non increasing length
Sort the lanes by non-increasing $a_j = \sum_{i \in I | i\ fits} \lambda_i$
Set $j'_1 = 1, \ldots, j'_{|J|} = 1$
for each $i = 1, \ldots, |I|$ **do**
  for each $i' = 1, \ldots, n_i$ **do**
    Let j_k the most suitable lane type (with lower a_j such that the vehicle type i fits)
    Place vehicle i' of type i on lane j'_k of type j_k
    if vehicle fits **then**
      Update j'_k as the lane of type j_k with more space available
    else
      Return false
Return true

2.4.3 Exact Model.

We introduce a second Mixed Integer Linear Program, MILP2, which is used to check the feasibility of adding an additional vehicle to the previously-booked vehicle mix. Although the objective of MILP2 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. Therefore, 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, so in the next iteration we know if a given vehicle fits in one lane without solving the model again by applying one of the heuristics from a previous optimal packing solution. We use the objective function to set priorities on which lanes to use, to avoid issues with symmetry and guarantee that the space remaining will decrease each time a new vehicle is considered. This objective is also set in order to obtain practical solutions when solving the models, although note that, in practice, the vehicles are placed in such a way that the *usable* space remaining is maximized.

As before, 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 really packed. We define a real variable u, which represents the 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, minimize u.

MILP2 is then as follows:

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

$$\sum_{j \in J} \sum_{k \in J_j} x_{ijk} = d_i \qquad \forall i \in I \tag{5}$$

$$M y_{jk} \geq x_{ijk} \qquad \forall j \in J, \forall i \in I,\ \forall t \in \{1, \dots, d_i\} \tag{6}$$

$$\sum_{i \in I} l_i x_{ijk} \leq \hat{l}_j \qquad \forall j \in J,\ k \in J_j \tag{7}$$

$$u \leq \hat{l}_j - \sum_{i \in I} l_i x_{ijk} - M(1 - y_j) \quad \forall j \in J,\ k \in J_j \tag{8}$$

$$x_{ijk} \geq 0 \text{ and integer} \qquad \forall j \in J,\ k \in J_j,\ \forall i \in I \setminus \{i'\} \tag{9}$$

$$u \in \mathbb{R} \tag{10}$$

Inequalities (5) are similar to (2), but the difference is that we ensure that the demand of all the vehicles is met. Inequalities (6) activate an additional lane if it is used by any vehicle. Inequalities (7) ensure that all of the vehicles assigned to lane $j \in J$ do not exceed the lane dimensions. Finally, inequality (8) forces us to use the lane with most space available sparingly. This forces the optimization to find solutions that fill the lanes up.

2.4.4 Overlapping Lanes.

The optimization routines presented above assign 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 the opportunity to place more vehicles as well as placing wider vehicles. Therefore, the formulation of MILP3, 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} (denoting the number of vehicles of type i assigned to the kth lane of type j) 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 $h_i < \hat{h}'_j$, $\forall i \in I,\ j \in J,\ k \in J'_j$, being k' the lane of type j' which corresponds to the consecutive lane of 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 uses lane k' of type j'. Again, these variables are defined only if the second condition is satisfied.

MILP3 is then as follows:

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

$$\sum_{j\in J} \left(\sum_{k\in J_j} x_{ijk} + \sum_{k'\in J'_j} z_{ijk'} \right) = d_i \qquad \forall i \in I \tag{11}$$

$$M y_{jk} \geq x_{ijk} \qquad \forall j \in J, \forall i \in I,\ \forall k \in J_j \setminus J'_j \tag{12}$$

$$M y_{jk} \geq x_{ij'k'} + z_{ijk} \qquad \forall j \in J, \forall i \in I,\ \forall k \in J'_j \tag{13}$$

$$\sum_{i\in I} l_i x_{ijk} \leq \hat{l}_j \qquad \forall j \in J,\ k \in J_j \setminus J'_j \tag{14}$$

$$\sum_{i\in I} l_i (x_{ij'k'} + z_{ijk}) \leq \hat{l}_j \qquad \forall j \in J,\ k \in J'_j \tag{15}$$

$$u \leq \hat{l}_j - \sum_{i\in I} l_i x_{ijk} - M(1 - y_j) \quad \forall j \in J,\ k \in J_j \tag{16}$$

$$x_{ijk} \geq 0 \text{ and integer} \qquad \forall j \in J,\ k \in J_j,\ \forall i \in I \setminus \{i'\} \tag{17}$$

$$u \in \mathbb{R} \tag{18}$$

The objective function is the same as in MILP2. Equalities (11) ensures that all the vehicles are placed. Inequalities (12) and (13) force activation of a lane if a vehicle has been assigned to it. Inequalities (14) and (15) ensure that the length of the lanes are not exceeded, and inequalities (16) 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 MILP3 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 type 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 using 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.

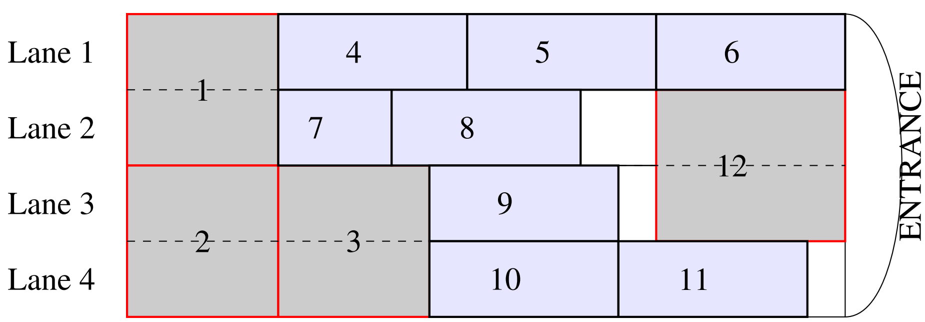

Figure 2: *Order of vehicles to be loaded in the ferry. There are four lanes and four vehicles requiring two lanes.*

3 Dynamic Configurations

In this section we generalize the model described in Section 2.2 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 configuration of the ferry vehicle decks and a finite set of ways of configuring the lanes within those decks. Together we can predetermine a complete set of configurations. The possible lane widths are set based on the widths of the expected vehicles. Hence instead of optimizing over a given configuration, here we optimized over them all.

Let $m \in \{1, \ldots, M\}$ be M different lane configurations where the number of each lane type in configuration m is denoted by $J^m = \{J^m_1, \ldots, J^m_{n_m}\}$ and n_m is the total number of different lane types in configuration m. 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 the new customer arises from considering all the different configurations that can be used.

Let $t' \in T$ be the time period in which a new customer of type $i' \in I$ arrives. In order to decide whether the new vehicle fits, it is enough that the algorithms used in Section 2.4 find a feasible solution for one configuration $m \in \{1, \ldots, M\}$. If there are one or more configuration that cannot accommodate vehicle i' and the vehicle is accepted, then these configurations are not considered in the next time periods. When combined with the dynamic programming, we find that if removing a configuration produces a lower expectation on future revenues, then a higher price would be offered to this customer.

In order to solve the DP equations presented in Section 2.2 to optimality, we calculate the value functions in all the possible states obtained by all the possible configurations in each time period.

4 Computational experiments

The computational results are based on data instances generated from examining the characteristics of real data from a vehicle ferry operator, with up to 5 vehicle types and a capacity of approximately 214 vehicles. We divide the computational experiments into four parts. In Section 4.1 we present the price acceptance model used in our experiments. In Section 4.2 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 4.3 we analyze the results obtained on instances where the layout of the ferry is different, but the ferry has the same dimensions. We show that the algorithm allowing dynamic configurations described in Section 3 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 4.2 and 4.3, we present results from implementing the models on the 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.

4.1 Price acceptance model

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

$$\alpha_{t,p} = d\left(\frac{1}{1 + e^{k\left(\frac{p}{q}-f\right)}}\right)\left(a + (b-a)\left(1 - \frac{t}{T}\right)^{c}\right), \tag{19}$$

where $d = 1 + e^{-kf}$ is a normalising factor with parameters a, b, c, k, f and q.

This price acceptance model assumes a logistic curve for the price-component, a function that is frequently used to describe willingness-to-pay (see **?**). 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. The parameter k controls the steepness of the sigmoidal price acceptance curve and represents 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.

4.2 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 of them 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, used by Red Funnel (denoted as RFF), has one main deck, another top deck and two movable decks which can be used when necessary. These movable decks, when used, reduce the height available of the main deck in some areas from 4.9m to 2.7m (see Figure 3). Therefore, by using the movable decks we reduce the capacity of 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.

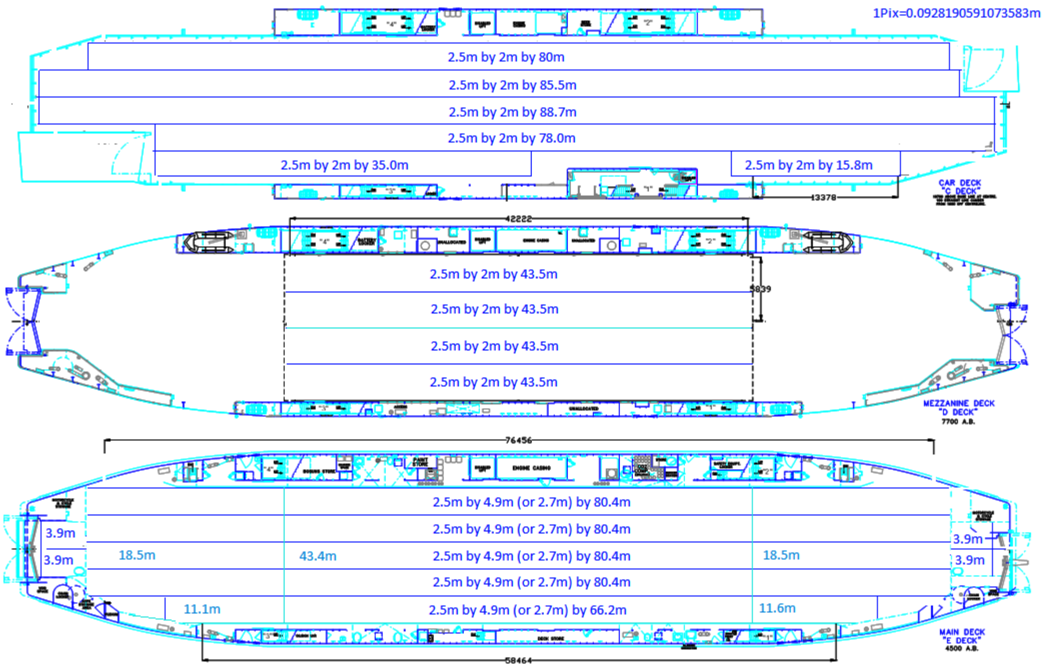

Figure 3: *Decks and lanes specifications in the ferry*

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. With three vehicle types we added 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.

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 (19). We set $c = 2$ to capture the effect that price acceptance probability increases at a faster rate as the time of departure draws closer; $d = 0.5$ to capture a competitive

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			x	x
V2	1.9m	1.5m	5m	x	x	x	x
V3	2.3m	2.5m	7m				x
V4	2.9m	3m	9m		x	x	x
V5	3.5m	4m	11m	x	x	x	x

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

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. For notational clarity we have omitted vehicle type indices from the parameters of Equation 19. 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 metre 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 obtained for the RM instances when considering fixed capacity vectors. We first calculate all the non-dominated capacity vectors. Then, for each of these capacity vectors we solve to optimality the dynamic programming equations described in (1). The first column shows the instance name, the second column the number of capacity vectors (non-dominated solutions when applying the algorithm from Section 2.3), and the third, fourth and fifth columns show, respectively, the average, minimum and maximum expected revenues obtained when solving (1) for all of the capacity vectors. For instance, in instance RMF_5 we solve 3601 different problems. The value of the revenues compared between instances shows the benefit of considering a better discretization for the vehicle types.

Table 3: *Number of possible solutions, average, minimum and maximum revenue where capacity is fixed.*

Instances	#cap	Av.	Min.	Max.
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 when applying the FF and ML heuristics and the exact model described in Section 2.4 to decide whether a new vehicle fits in the ferry or the new customer should be rejected. Each one of these approaches solves the dynamic pricing problem in equation (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. Between instances with four and five vehicles types, RM_4 and RM_5, we can observe that the ML and FF obtains the same number of states and the same expected revenues.

For RMF_5, the computation time for the exact algorithm is 40% greater than for the heuristics. Note that we are solving the off-line problem, i.e, the policy to be used 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: *Dynamic capacity*

Instances	ML			FF			Exact		
	#states	Exp R	Time	#states	Exp R	Time	#states	Exp R	Time
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 fixed booking limits, predetermined before starting the selling season (see Table 3), with the dynamic capacity approaches (see Table 4) we can observe that the most promising 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 and the fixed booking limits strategy may waste more space. Therefore, from this 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 the most efficient way.

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 where we assume that booking limits are flexible (i.e. dynamic capacity). 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.

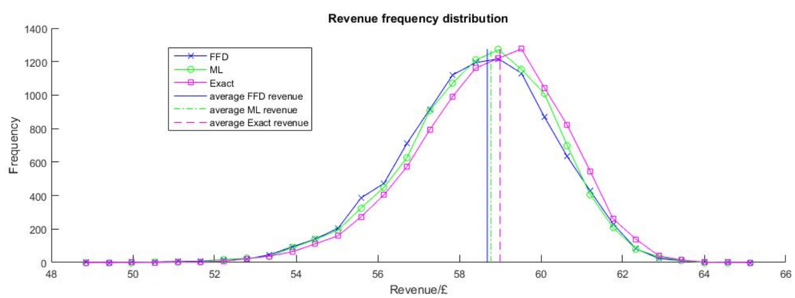

Figure 4: *Expected revenues with instance RM_2*

In Figures 6 and 7 we present different quantiles of the total length of the vehicles 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 same time as the selling season ends.

Although the problem can be solved to optimality within a reasonable computational time, in what follows, we use the ML and FF heuristics to speed up the full optimization routine. We solve the initial states (where the ferry is generally fairly empty) using the ML algorithm. Where the ML algorithm rejects the customer then, before rejecting, we apply the FF and finally the exact algorithm. Note that the FF algorithm is very fast and the

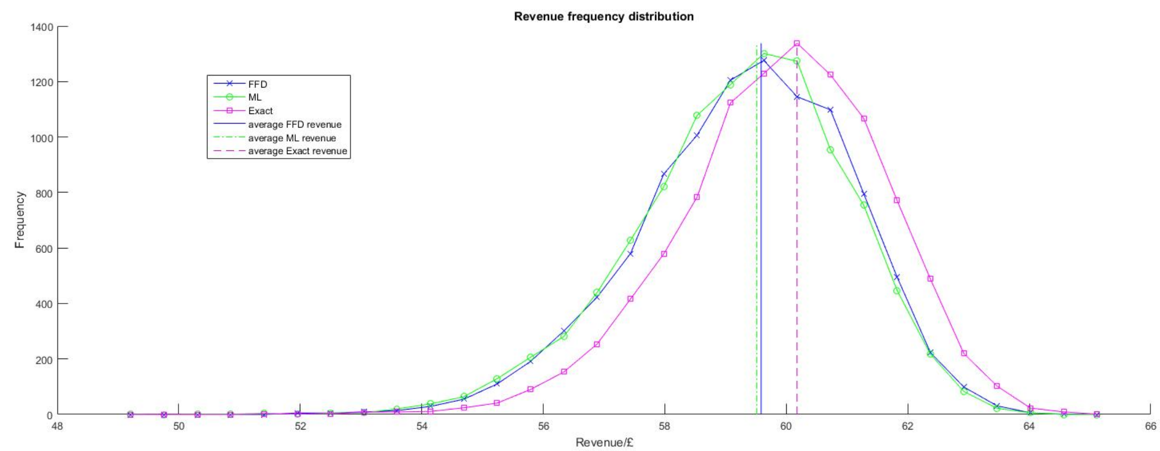

Figure 5: *Expected revenues with instance RM_3*

states found by this algorithm usually are different than the states found by the ML algorithm. This approach allows us to speed up the packing process but still solve the problem to optimality.

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 y-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 can have higher future expected revenue than states with a shorter total length of vehicles booked. This is due to the efficiency of the packing as 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. This effect becomes more pronounced as deck space runs out and the ferry nears capacity.

It is important to highlight here that the MILP model finds all 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.

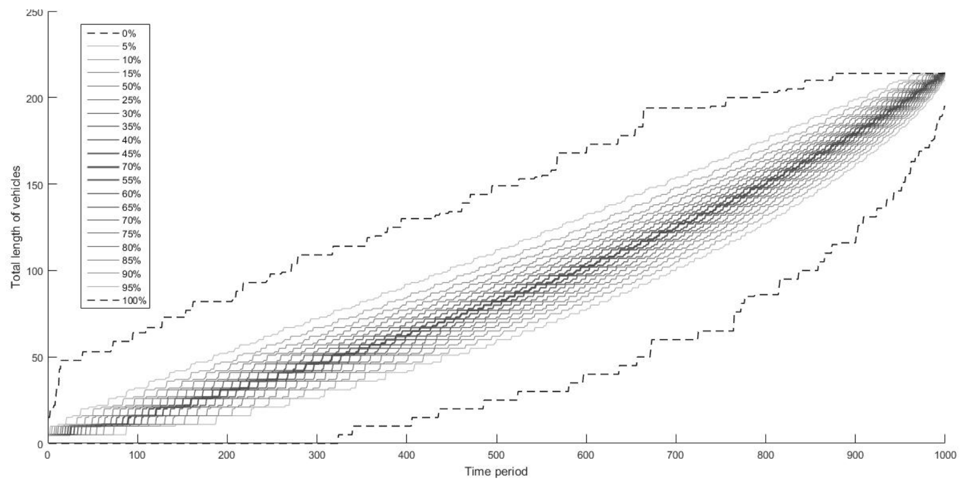

Figure 6: *Quantiles over the different time periods with the exact algorithm in RM_2*

4.3 Dynamic layout

In this section we compare the results obtained when solving instances RMF_2, RMF_3 and RMF_4 with all the 18 possible ferry configurations. We compare the exact algorithm (see Section 2.4.3) and the dynamic layout approach presented in Section 4.3 to assess the impact of the layout on the expected revenues.

We use the ferry defined above 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 have considered 18 possible combinations of lanes of width 2m, 3m and 4m within the total ferry.

Table 5 presents the results obtained by applying the exact algorithm to each of the combinations 1 through 18. The final row shows the result obtained by the algorithm described in Section 3, where all the possible ferry configurations are considered. We observe that the revenue is higher than for any of the fixed configurations but, for each instance, there is a configuration which produces an expected revenue close to that for the dynamic layout (e.g. configuration 2 for RF_2). The reason being that this configurations matches the vehicle demand pattern well. The number of states varies between ferry configurations because the number of possible vehicle mixes will be different; but a higher number of states does not necessarily mean a greater expected revenue.

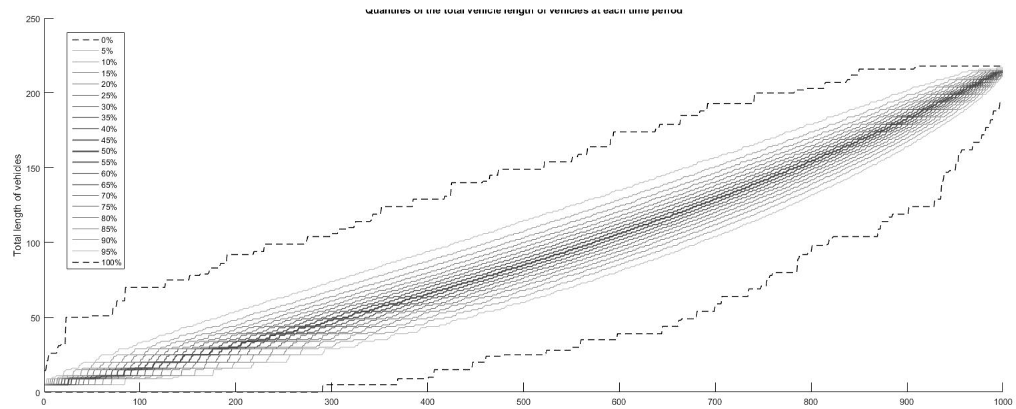

Figure 7: *Quantiles over the different time periods with the exact algorithm in RM_3*

4.4 Real-World Instances

In Table 4 we present the solutions obtained when solving instances provided by a UK ferry operator by applying the ML and FF heuristics and the exact model described in Section 2.4. Note that this instance is derived from real company data, and it is a greater size than the previous instances.

The results obtained mimic those of Table 4 but with greater computation times. We enumerate more states when solving the packing problem to optimality using MILP2 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, is having 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.

5 Conclusions

We have described a set of algorithms that can be used for pricing the sale of different-sized products that must be packed into a three-dimensional space. The problem was simplified by assuming the allocation of products to lanes, but clearly demonstrates through the numerical results that understanding how the products will be packed and providing good measures of the packing during the selling period results in higher revenues. Fixing the booking

Table 5: *Dynamic layout*

Config	2 vehicles type			3 vehicles type			4 vehicles type		
	#states	VF	Time	#states	VF	Time	#states	VF	Time
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
DL	1338	62.57	12	8385	64.71	143	327599	67.53	5323

Table 6: *Dynamic capacity*

	ML			FF			Exact			Extended Exact		
Instances	#states	Exp R	Time	#states	Exp R	Time	#states	Exp R	Time	#states	Exp R	Time
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

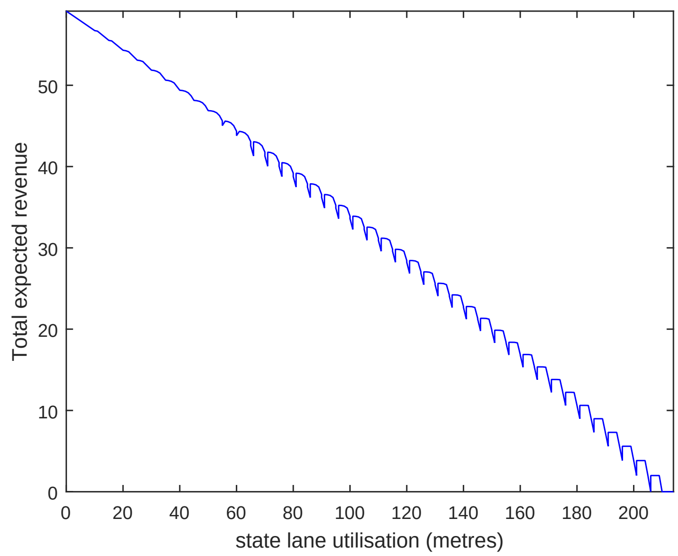

Figure 8: *Expected revenues from each state on instance RMF_2 (two vehicle types)*

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.

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 MILP3, 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 modelled 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.

Acknowledgement: 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.

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. *Eur. J. Oper. Res.*, 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.

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, 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 inverntory transshipment modeling to air cargo revenue management. *Transportation Research Part E*, 57:27–44, 2013.