Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

A SIMHEURISTIC APPROACH TO THE VEHICLE FERRY REVENUE MANAGEMENT PROBLEM

Christopher Bayliss
Julia M. Bennell
Christine S.M. Currie
Antonio Martinez-Sykora
Mee-Chi So

Mathematical Sciences
University of Southampton
Salisbury Road
Highfield
Southampton, SO17 1BJ, UK

ABSTRACT

We propose a Simheuristic approach to the vehicle ferry revenue management problem, where the aim is to maximize the revenue by varying the prices charged to different vehicle types, each occupying a different amount of deck space. Customers arrive and purchase tickets according to their vehicle type and their willingness-to-pay, which varies over time. The optimization problem can be solved using dynamic programming but the possible states in the selling season are the set of all feasible vehicle mixes that fit onto the ferry. This makes the problem intractable as the number of vehicle types increases. We propose a state space reduction, which uses a vehicle ferry loading simulator to map each vehicle mix to a remaining-space state. This reduces the state space of the dynamic program, enabling it to be solved rapidly. We present simulations of the selling season using this reduced state space to validate the method.

1 INTRODUCTION

We propose a Simheuristic approach to solving the vehicle ferry revenue management (RM) problem. The objective is to maximize the revenue from selling tickets to vehicles, where the price can be changed dynamically during the selling season based on the time left until the ferry's departure and the mix of traffic for which tickets have already been sold. The selling season begins when tickets are first put on sale for for a crossing, approximately 6 months before departure.

Vehicle ferries transport passengers and their vehicles, each of which occupies a region of the available deck space. Vehicles range in size from large trucks to motorcycles, each requiring a different amount of the ferry's available capacity. Typically, the vehicle deck space is the binding capacity constraint and the lifeboat capacity, which dictates the maximum number of passengers allowed, is not met. Customer arrivals into the booking system are stochastic and customers' willingness-to-pay (i.e., the probability that they will pay a given price) varies according to their vehicle type, and the time left until departure. The latter is a typical characteristic of pricing problems; see for example Littlewood (2005) in which higher paying customers arrive closer to the departure time.

The amount of remaining capacity during the selling season depends on the mix of accepted vehicles and the procedure that is used to pack/load them onto the ferry. The loading procedure is manually controlled by deck crew who have to take account of maneuverability, disabled access and other case-specific constraints.

We implement a *dynamic pricing* algorithm to set the prices for each vehicle type during the selling period, dependent on the remaining deck capacity. One of the main benefits of dynamic pricing is its potential to regulate demand in scenarios in which supply is limited, in a profitable way. The additional complexity in the vehicle ferry RM problem is that when setting prices, the ferry operator needs to solve a packing problem at each time step to determine the remaining available space. To solve this exactly, the states of the dynamic program should correspond to the mix of vehicles for which tickets have already been sold. However, as the number of vehicle types increases, the state space soon becomes too large to be tractable.

In this paper, a simulation model of the ferry loading process is used to derive transition functions for a two-dimensional approximation of the actual state space. The two-dimensional state approximation consists of one dimension for the main vehicle deck and another for the upper (car-only) deck. The transition functions in this reduced state space are based on how easily a new vehicle can be fitted onto the ferry. We define the *packing loss* as a measure of the extra space a vehicle takes up after it is packed, in addition to its area. Making use of the packing loss estimate, the transition functions return the remaining space on each deck once a given vehicle type has made a booking and these are used in the dynamic program. Our results show that this approach reduces the state space sufficiently to allow a dynamic programming approach to be used to solve large problems in a reasonable time.

The Simheuristic approach to solving the vehicle ferry RM problem is characterized by three main stages

1. Use a simulation of the loading procedure to derive packing loss prediction functions, which predict the packing loss for any given mix of vehicles. These packing loss functions are used to calculate the transition functions for a dynamic programming formulation of the vehicle pricing problem that is solved in a two-dimensional *remaining-space* state space. The loading simulator is described in Section 4.
2. Solve the dynamic program to find the prices that should be offered at each time in each remaining-space state, described in Section 5.
3. Use the loading simulation to map from the the mix of accepted vehicles to the remaining-space state and find the set of prices that should be offered.

The remainder of the paper is organized as follows. We review the relevant literature in Section 2, before describing the problem in more detail in Section 3. Results are presented in Section 6 and we draw some conclusions and discuss future work in Section 7.

2 RELATED LITERATURE

Simheuristic algorithms tackle optimization problems by integrating simulation in heuristic methods and Gonzalez-Martin et al. (2014) provide an overview of the concept of a Simheuristic algorithm with particular reference to the vehicle and arc routing family of problems. This is a growing area, with diverse applications (e.g., renewable energy Mallor et al. (2015), social networks Pérez-Rosés and Sebé (2015)) as well as the more traditional location modeling and production optimization examples.

The Simheuristic approach that is considered in this work can also be classified as an approximate dynamic programming approach. The textbooks of Bertsekas and Tsitsiklis (1996) and Powell (2007) provide a comprehensive treatment of this subject area. Approximate dynamic programming is concerned with finding approximate solutions for problems that have an intractably large state space. This work proposes an approximate value function structure based on the idea of mapping high-dimensional states to two-dimensional states, and then deriving transitions functions for the dimensions of the original problem in the reduced state space.

We have found no other work related to RM in the vehicle ferry industry but Maddah et al. (2010) apply RM to optimal pricing of cabins on cruise ships. With a cruise ship, there are two competing constraints: lifeboat capacity and cabin capacity. In this sense, the cruise ship RM problem is similar to the vehicle ferry RM problem, although in our example, the lifeboat constraint is rarely binding and the capacity constraint for the vehicles is continuous and two-dimensional rather than integer and one dimensional, which is the case for cabins. Cruise ship RM suffers from a large state space, which Maddah et al. (2010) solve using heuristics, including single dimension state space reduction schemes and dynamic programming

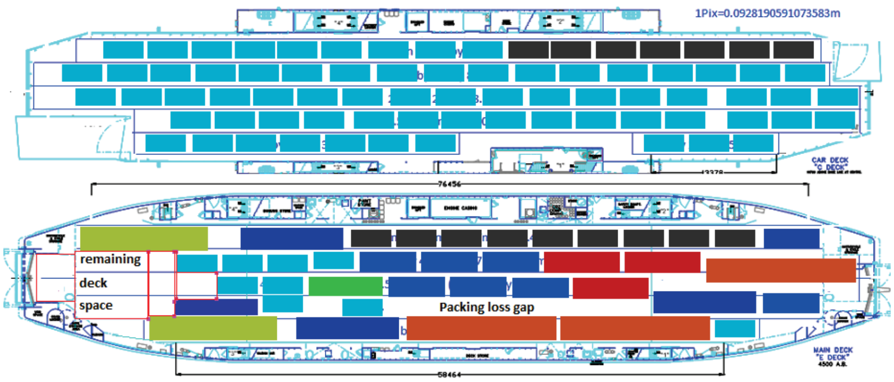


Figure 1: Ferry lane layout diagram with an illustration of packing loss.

heuristics. The latter are based on decomposing the problem into two separate dynamic programs where value functions are derived separately for each dimension of the problem, and then combined when the solutions are implemented.

An area of research that is closely related to the vehicle ferry RM problem is that of air cargo RM. In this problem airlines face many of the same capacity constraints as vehicle ferry operators, the main difference being that weight tends to be a more important constraint in the airline case. The additional complexities can force model solutions to be more simplified, e.g., by using density values from historical data. More details can be found in Kasilingam (1996). Generally, in the airline case, the objective of the optimization is to minimize overbooking rather than to optimize revenue, partly due to the uncertainty over the dimensions of the packages at the time of booking. More details can be found in Kasilingam (1996), J.S.Billings (2003), Amaruchkul et al. (2007). Solution methods used include Markov Decision processes Han et al. (2010); dynamic stochastic knapsack Kleywegt and Papastavrou (1998); newsvendor models, e.g., Wong et al. (2009), Zou et al. (2013).

3 CASE STUDY

The methods we describe have been tested on data from the Red Funnel vehicle ferry service operating between Southampton and the Isle of Wight, UK. Red Funnel have 3 near identical Vehicle Ferries, each with 2 permanent vehicle decks, approximately 80 meters long and 12.5 meters wide. The main deck has a high ceiling (approximately 5 meters) and accommodates all vehicle types from motorcycles to large freight vehicles and coaches. The upper deck has a relatively low ceiling (approximately 2 meters) and accommodates cars only.

As can be seen in Figure 1, the ferry is nearly symmetrical both laterally and longitudinally, with the lifts from the vehicle deck only on one side of the ship. Vehicles enter at one end and exit at the other, swapping directions for the return journey.

We consider 13 main types of vehicles: cars, motorcycles, vans, minibuses, coaches, medium and large freight vehicles, drop trailers, caravans, other towed vehicles, parcel cages, unaccompanied cars and a miscellaneous category. The price acceptance probability model and the physical dimensions vary between vehicle types.

4 LOADING SIMULATOR

Before describing the loading simulator, it is useful to define the term *packing loss*, which is used extensively in this section. Packing loss is the term used to describe the unusable spaces between vehicles that are created by the procedure that is used to load vehicles onto the ferry. It includes parking gaps,which are required for passengers to exit the vehicle deck, as well as larger gaps that occur when a parking space is blocked off by another (usually larger) vehicle.

The loading simulator is built in Java and can provide a visual output, as demonstrated in Figure 1. It is designed to mimic the loading procedure used by the ferry operator on the day of departure. This enables a better estimation of the space used by each vehicle, which includes the packing loss, which may come about due to the decisions made during loading. The procedure we implement in the simulation is based on observations and conversations with loading staff.

Loading of the main deck and loading of the upper car deck can be considered separately. The reason for this distinction is that all cars fit within the lanes of the upper deck, which means that the loading of the upper deck is purely procedural. On the other hand, the main deck accommodates all vehicle types and not all vehicle types fit within the lanes. In such cases the adjacent lanes to a large vehicle type cannot be used and a staggered loading pattern may emerge (as in a two-dimensional bin-packing problem).

Vehicle ferry operators face a number of practical constraints on placing vehicles that prevent them from treating the loading process as a two-dimensional bin-packing optimization problem (see Lodi et al. (2002) for a survey of two-dimensional bin-packing problems and Csirik and Woeginger (1998) for a tour of performance results for on-line packing heuristics). For example, vehicles with hazardous materials have to be parked at the front or the back of the main deck underneath the sprinklers; large vehicles cannot be parked in the corners next to the exits due to issues of maneuverability; customers may require unimpeded access to the lifts; and some customers pay an additional charge for priority boarding.

The loading procedure is manually controlled by loaders and yard personnel. Yard personnel sort vehicles into different lanes on the dockside prior to boarding, according to their types and any special requirements, and notify the loaders of the number of vehicles of each type that are to be loaded. Loading of the upper and main decks can take place simultaneously, and cars are loaded onto the upper deck if possible. As the loading of the upper deck is straightforward, we do not describe it further here and instead concentrate on the loading of the main deck.

Loading of the main deck is complicated by the wide variety of vehicle types that it can accommodate. During the loading process, the loaders request vehicle types for the main deck and if these are available the yard personnel release the requested vehicle types onto the ferry. We assume in this work that loaders always get the vehicle type they request; however it would be a relatively simple addition to model the case where this is not possible. The choice for the next vehicle type is based on special requirements (e.g., lift access, positioning of hazardous materials or particularly heavy loads), position/height restrictions, but when these have been dealt with, loaders will tend to fill the space closest to the side of the deck before directing vehicles to the inner lanes.

Within the simulator, the loading algorithm generates a list of currently available parking positions; these *open positions* are the positions where a vehicle can be parked in front of the previous vehicle and adjacent to the wall, or next to another vehicle. Open positions are similar to the positions that would be considered if the "bottom left (and right)" packing algorithm was being used. The available open positions are then sorted according to their distance from the nearest wall and their distance from the ferry exit and the algorithm allocates a vehicle to the first open position. Open positions are ordered following a maximum length rule if adjacent to the wall and following a minimum length rule if in the middle of the ferry. In choosing a vehicle to place, the algorithm will first try to place any vehicles with special requirements, before moving onto the remainder of the vehicles. In both cases it will aim to fit the largest vehicle into the available space first of all. This enables the loading simulator to take account of all of the special requirements but the packing heuristic also helps to reduce the number of unreachable gaps in which a vehicle could have been parked. When a vehicle is placed onto the ferry by the simulator, a parking gap between the vehicle in front is also added. The rules used in the algorithm are based on discussions with the loaders. In order to validate the simulation, historical loadings were simulated and discussed with our commercial partners and the feedback was used to improve the simulation.

4.1 Estimating Packing Loss

Our proposed state reduction approach is to map the vehicle mix state to a two-dimensional state that is defined to be the *remaining-space* on each of the two decks. The loading simulator calculates the remaining deck space for each deck every time a vehicle is loaded. To calculate the remaining deck space on the upper car deck, the simulator sums the unused length of each non-full lane multiplied by the average car width. For the main deck, where a mix of traffic can be loaded, the remaining deck space is estimated by generating a set of non-overlapping polygons that do not overlap any of the parked vehicles, where each polygon has a corner which is an open position big enough to accommodate either the shortest or narrowest vehicle type. The remaining space on the main deck is assumed equal to the sum of the areas of this set of polygons. Furthermore, the packing loss that can be attributed to a given vehicle type is set equal to the change in the remaining deck space after loading that vehicle type minus the area occupied by that vehicle type. In Section 4.2 packing loss estimates are used to generate the transition functions for the dynamic program.

The packing loss that can be attributed to any given vehicle type is strongly and non-trivially dependent upon the amount of remaining deck space. As a demonstration of this, we fit piecewise linear regression functions to predict the total packing loss as a function of the number of vehicles of each type on both decks combined. Using data from the loading simulator, a series of piecewise linear regression functions with increasing numbers of remaining deck space intervals were derived for predicting the total packing loss (i.e., (deck area)-(remaining deck space)-(sum of the vehicle areas)) as a function of the number of vehicles of each type on both decks combined. The packing loss prediction functions take the following linear form.

$$\text{Packing loss prediction} = \sum_{j=1}^{vTypes} a_{i,v} N_v, \tag{1}$$

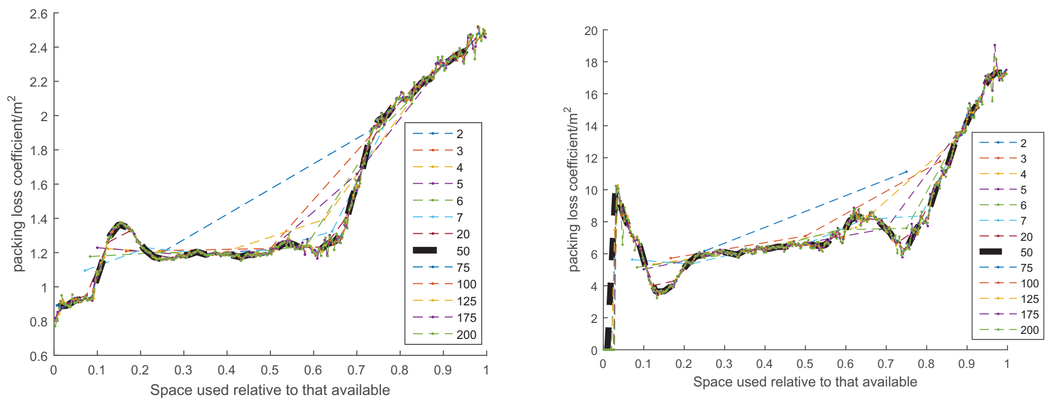
where the $a_{i,v}$ are regression coefficients for vehicle type v in the i^{th} section of the piecewise function and N_v is the number of vehicles of type v. No constant terms were included in the linear functions, meaning that all packing loss is directly attributed to the vehicles that are parked. The coefficients of the linear regression equations can be interpreted as the packing loss caused by each additional unit of that vehicle type for the given amount of remaining deck space.

Figure 2 shows the regression coefficients for cars (Figure 2a) and large freight vehicles (Figure 2b) for piecewise linear functions with different numbers of intervals. This demonstrates very clearly that as the ferry fills up, the amount of packing loss increases, i.e., packing becomes more awkward as the remaining-space decreases. We also see some peaks in the function, e.g., an early peak for freight vehicles corresponding to the fact that large freight vehicles cannot be parked in the corners of the main deck due to maneuverability constraints. When the ferry is empty, cars incur little packing loss, but we can observe a peak when cars requiring lift access need loading onto the main deck.

Increasing the number of intervals in the piecewise linear approximation initially causes the packing loss coefficients to converge towards a smooth curve that depends on the amount of space used in the ferry, but using a very large number of intervals can cause the packing loss coefficients to include random fluctuations.

The loading simulator is used to derive per vehicle average packing loss contributions for each vehicle type for each level of remaining space on both the main and upper decks. We make 5000 iterations, in each of which the loading simulator is used to generate and load random vehicles one at a time until the ferry is full. Before each vehicle is loaded the simulator adds and removes each vehicle type, calculating the packing loss. The average packing loss contributions for each vehicle type are calculated for each discretized interval of remaining-deck-space on each of the main and upper decks. Figure 3 illustrates the packing loss contributions that were derived from the loading simulation.

We can see in Figure 3a that the packing loss per car remains nearly constant until an unusable gap is created at the end of each lane. Six of the peaks correspond to lane endings, the other two correspond to the reverse gap that is required in two of the lanes. Similar characteristics can be seen in Figure 3b but this demonstrates the general trend that larger vehicle types incur a greater packing loss. Packing loss is predictable at low levels of usage, but as the ferry fills up packing is complicated by the variable rectangle packing problem that emerges when different vehicle types are loaded onto the same deck. The peak that



(a) Car coefficients.

(b) Large freight coefficients.

Figure 2: Packing loss coefficients as the amount of used space varies for (a) cars and (b) large freight. The lines correspond to different numbers of intervals for the piecewise linear function.

occurs when the amount of remaining space is around $600m^2$ corresponds to the spaces adjacent to the sides and exit being filled, which means that packing becomes more awkward. Each vehicle type also has an additional peak in packing loss as the main deck reaches capacity, which corresponds to that vehicle being the last vehicle that can be fit. The zero packing losses indicate that no additional vehicles of that type could be loaded.

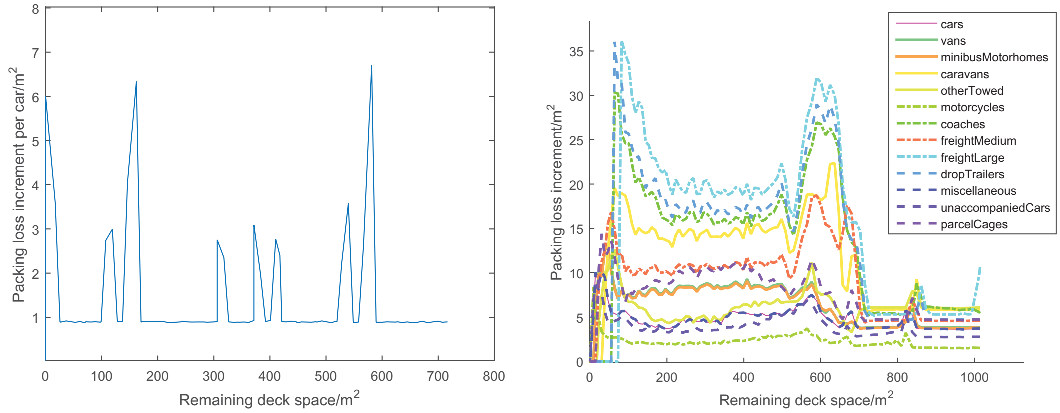
The difference between Figures 2 and 3 is that Figure 2 groups the decks together and is focused on the prediction of the total packing loss. In contrast Figure 3 splits the two types of decks and is focused on the prediction of per vehicle packing loss contributions. The packing estimates in Figure 3 are now used to derive the transition functions for dynamic program for the vehicle pricing problem. For each discrete interval of remaining deck space packing loss estimates are derived for vehicles which have at least a 95% chance of still fitting onto the ferry at the given level of remaining deck space.

4.2 Transition Functions

During the selling season, transitions within the state space are always orthogonal, i.e., each time a ticket is purchased for a particular vehicle type, the vehicle will either detract from the capacity of the upper car deck or the main deck but not from both. The dimensions of the remaining-deck-space state are also discretized into intervals to facilitate a look-up-table approach to solving the dynamic program, where the number of intervals is chosen so that the interval size is approximately the area of a single car.

Upon the sale of a ticket to a vehicle during the selling season the two-dimensional remaining-deck-space state is updated dependent on the size of the vehicle. Within a deck, the new remaining-deck-space is equal to the current value minus the vehicle's area and the packing loss that it incurs.

Let $s = \{r_u, r_m\}$ denote the remaining deck space state at a given time, where r_u is the remaining area on the upper car deck and r_m is the remaining area on the main deck. A_v is the area occupied by vehicle type v and $PL_{d,r_d,v}$ is the packing loss that occurs if a vehicle of type v is loaded onto deck d when the remaining space is r_d. If a vehicle (v) purchases a ticket and is loaded onto a deck (d) the new remaining deck space state element is $r_d^{'} = r_d - A_v - PL_{d,r_d,v}$. Every time a sale occurs the state changes in only one of the state dimensions because each vehicle is assigned to a single deck. In particular, cars are loaded onto the upper car deck if possible, and then the main deck if this is not possible. We write the transition function as $s' = F(s,v)$, where $s^{'}$ denotes the new state, given that the current state is s and and vehicle type v purchases a ticket.



(a) Upper deck packing loss.

(b) Main deck packing loss.

Figure 3: Packing loss directly attributable to vehicle types added to (a) the upper car deck; and (b) the main deck for different levels of remaining-deck-space.

5 FORMULATION OF THE DYNAMIC PROGRAM

In this section we formulate the vehicle ferry RM problem as a dynamic program. We discretize time into intervals which are small enough such that it is reasonable to assume that at most one customer will arrive into the booking system during each time interval, and the remaining-space states are discretized into units approximately equal to the area of a car. To allow for smaller vehicle types such as motorcycles, the value function is treated as a piecewise approximation by estimating the value of the new state as a weighted sum of the closest states in the dimension of the transition, where the weights depend on the distance from the neighboring state boundaries. The arrival rates of different vehicle types are derived from demand data for the target selling season and are assumed to be constant throughout the selling season. Non-homogeneous arrival patterns can easily be simulated but we have not done so here. Customers' willingness-to-pay, or the probability that they will purchase a ticket of price p_v, $\alpha_{v,p_v,t}$, is assumed to vary between vehicle types v and with the time remaining until the end of the selling season t. Taken in combination, the vehicle arrival rates and the price acceptance models simulate a time non-homogeneous demand pattern. Let λ_v denote the arrival rate of vehicle type v during each interval of the selling season. The vehicle pricing problem can then be solved by finding the set of prices that satisfy the Bellman Equation,

$$V_{t,s} = \left\{ \sum_{v=1}^{vTypes} \lambda_{t,v} \max_{p_v \leq pMax_v} \left\{ \alpha_{v,p_v,t} \left(p_v + V_{t-1,F_{s,v}} \right) + \left(1 - \alpha_{v,p_v,t}\right) V_{t-1,s} \right\} \right\} + \left(1 - \lambda_{t,0}\right) V_{t-1,s}. \tag{2}$$

Equation 2 states that the expected future revenue of the remaining-space state s at time t is found by determining the set of prices to offer each vehicle type which maximize the sum of the marginal revenue attained from sales in that time period plus the expected future revenue of following an optimal pricing policy thereafter. Notice that the price optimization is carried out for each vehicle type individually, as a consequence of the assumption that the time intervals are small enough that a maximum of one vehicle can arrive in each time period. A backwards dynamic programming approach can then be used to determine the optimal value function and the optimal pricing policy.

The structure of the willingness-to-pay function is given by

$$\begin{gathered} \alpha_{p,t} = CF \left(1 - \left(\frac{1}{1 + e^{-k\left(\frac{p}{pMax} - r_0\right)}} \right) \right) \times \left(a + (b-a)\left(1 - \frac{t}{T}\right)^{c} \right) \\ CF = \left[1 - \left(\frac{1}{1 + e^{kr_0}} \right) \right]^{-1}, \end{gathered} \tag{3}$$

where CF is a correction factor to ensure that the probability of purchase is equal to 1 when the price is zero. The parameters a, b, c, k and r_0 are set for each vehicle type and affect the shape and scale of the price acceptance function. The variable $pMax$ is the maximum price that a customer is willing to pay for that vehicle type and T is the length of the selling period. We base the model on the assumption that the probability of accepting the minimum price is less than or equal to a at the beginning and b at the end of the selling season.

This price acceptance model has two multiplicative components, one for time and another for price. A logistic curve is used to model the price component, which implicitly assumes that the distribution of reservation prices is bell-shaped (e.g., see Philips (2005)). This shape has considerable flexibility and can model a wide range of behavior. The time component can capture non-linear effects that time may have on the probability of price acceptance, through the parameter c.

For each vehicle type, each state and each time we need to find the optimal price p^* for a problem of the general form

$$\alpha_{p,t}\left(p + V_{t-1,s'}\right) + (1 - \alpha_{p,t})V_{t-1,s}. \tag{4}$$

As $\alpha_{p,t}$ is decreasing in p, we can show that the Golden Search optimization routine will converge to the optimal solution (See Burley (1974) for a good introduction to Golden Search).

6 EXPERIMENTAL RESULTS

We present results for three different demand scenarios corresponding to: 1. Average demand rate; 2. High proportion of large freight vehicles, low proportion of cars; 3. Low proportion of large freight vehicles, high proportion of cars.

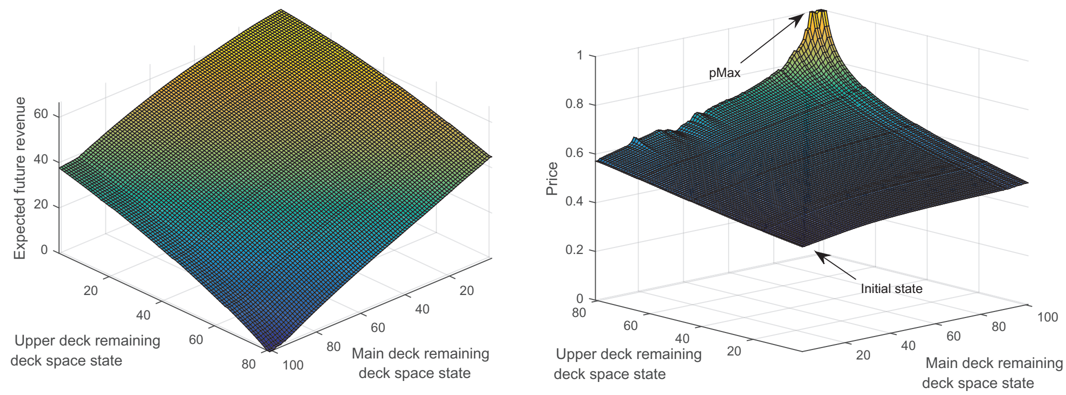
The latter two scenarios correspond to typical sailings in the early morning and middle of the day, respectively. Note also that the term high freight demand is relative, as in both scenarios 2 and 3 car demand is higher than that of large freight vehicles.

The following results are based on the assumption that a customer's maximum willingness-to-pay is proportional to the area that vehicle type occupies on the ferry (excluding packing loss). This assumption does not change the characteristics of the resultant pricing policy but helps to highlight how it is influenced by packing loss considerations. Furthermore, in the price acceptance model of Equation 3 the maximum price for each vehicle type ($pMax_v$) is scaled so that the maximum price for a car is set to 1 (i.e., $pMax_v = \frac{A_v}{A_{car}}$), this step scales the pricing problem linearly so that all prices can be intuitively compared to the maximum prices that are offered to cars.

For each demand scenario the dynamic program was solved to determine the optimal price to offer each vehicle type v, in each state s at each time period t during the selling season. The dynamic program takes around 5 minutes to solve for each demand scenario (around a second for each time step). Figure 4a illustrates part of the value function evaluated at the beginning of the selling period; and Figure 4b shows the car pricing strategy at $t = 0$, the end of the selling season, demonstrating how the price increases between its minimum and maximum as space on the upper car deck and the main deck becomes scarcer. At the maximum price, $pMax_v$, the willingness-to-pay drops to zero, preventing overbooking. Both are produced using scenario 3, but we obtain similar shapes for each of the scenarios considered.

The dynamic program predicts expected revenues of 64.61, 61.56 and 65.44, for scenarios 1 to 3 respectively. With a single, fixed price for each vehicle, set at the *stock-clearing rate*, as defined in Talluri and Ryzin (2004), we predict expected revenues of 59.46, 56.26 and 61.35, suggesting that the dynamic pricing policy is outperforming the basic fixed price policy by up to 9.4%. Using the stock-clearing rate, which is defined to be the rate that would be charged to sell all of the inventory over an infinite time horizon, is a low bar for comparison with our policy, but it acts as validation of the dynamic program.

In order to provide a crude comparison with current practice, we compared our results with a situation in which time slices of the dynamic programming pricing solution at times 0, $T/4, T/2, 3T/4$ and T were tested as pricing policies, where T is the length of the selling season. Each time slice has the property that prices increase in decreasing capacity. This mimics the current practice in which a fixed sequence of increasing prices are used with a price increase triggered when the next increment of the ferry capacity



(a) Value function at the beginning of the selling season.

(b) Prices offered to cars as the remaining-deck-space varies at the beginning of the selling season.

Figure 4: Demand scenario experiment results for demand scenario 3.

is reached. Results are given in Table 1 and show that allowing for demand fluctuations and the time remaining in the selling season has revenue benefits but if time slicing is used, the best revenue time slice pricing solutions occur during the middle of the selling season.

Table 1: Time slice solution results.

Demand scenario	Time slices				
	0	0.25	0.5	0.75	1
1	52.88	56.21	58.42	56.99	58.57
2	51.60	57.76	61.09	60.71	52.71
3	53.07	57.10	60.16	60.52	58.49

As a further test of the dynamic program, we use a simulation of the selling season with the price recommendations. Demand for each vehicle type is generated from an homogeneous Poisson distribution and each unit of demand purchases with a probability equal to the willingness-to-pay function given in Equation 3. The loading simulator must be run after each sale in order to estimate the current remaining-space state, which is necessary for setting the prices. The loading simulator is also used to test for feasibility, i.e., whether a particular vehicle type can fit onto the ferry. The same process would need to be carried out if the dynamic program was used in a real situation. Each iteration of the selling season simulation takes approximately 1 second. We run 10000 iterations of the selling season simulation for each of the scenarios, and results are displayed in Figure 5. These demonstrate that scenario 2, in which we have a high proportion of freight vehicles, has a significantly lower revenue than scenarios 1 and 3. Larger vehicles are more awkward to pack, corresponding to high packing loss functions and so this result makes intuitive sense.

Figure 6 shows the average prices that were on offer to each vehicle type in the selling season simulation. The prices for each vehicle type v are scaled such that 1 corresponds to $pMax_v$ and 0 corresponds to the stock-clearing rate for the same vehicle type. There are some interesting differences between Figures 6a and 6b, which show the prices charged for scenario 2 (high freight) and scenario 3 (high cars) respectively. In the high freight scenario, prices for large vehicle types tend to increase at the end of the selling season, but those for smaller vehicles actually appear to drop off, as the ferry reaches its capacity for the larger freight vehicles but still has some smaller spaces remaining. With a higher proportion of cars, we see an increase in the price towards the end of the selling season for all of the vehicle types.

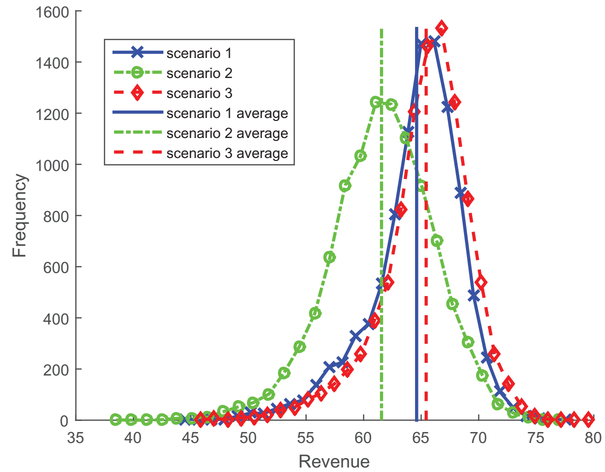
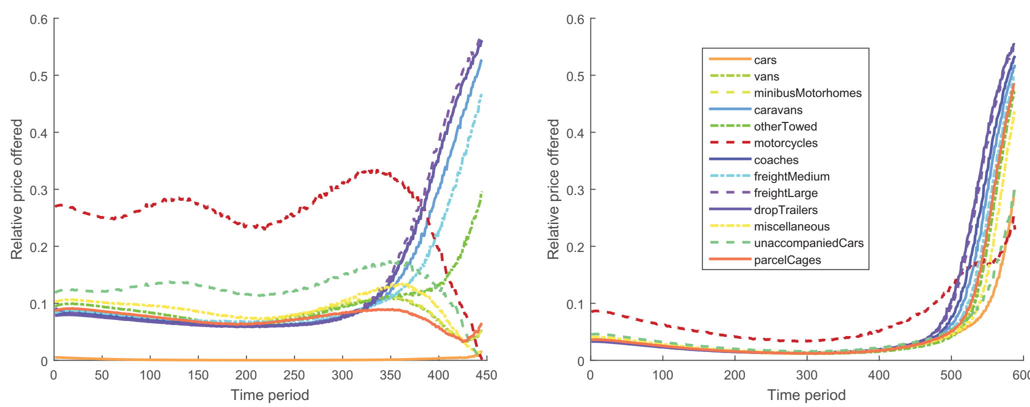


Figure 5: Distributions of the revenue for each of the three demand scenarios, derived from 10000 iterations of the selling season simulation.



(a) Demand scenario 2.

(b) Demand scenario 3.

Figure 6: Average prices on offer to each vehicle type at each time period relative to $pMax_v$ and the stock clearing price.

7 DISCUSSION

This paper describes a Simheuristic approach to the vehicle ferry RM problem. A loading simulator is used to estimate the remaining space on the vehicle decks of the ferry, and to deduce how the remaining-space decreases as an extra vehicle is added. This reduces the state-space of the dynamic program to two states, allowing dynamic pricing to be used to solve the pricing problem. In doing so, we show that the packing loss associated with a particular vehicle type increases as the area of the vehicle type increases, in line with intuition that larger vehicles are more awkward to pack. We show that the expected revenue from the dynamic programming approach is higher than that obtained using a fixed price pricing policy, although we acknowledge that we are not using the optimal fixed-price policy for a finite selling period. Computational times are of the order of 1 second per iteration for the simulation model and approximately 5 minutes for the dynamic program.

Future work will begin by extending this method to account for ferries in which different configurations are possible. For example, many ferries have mezzanine decks that can be lowered to provide additional space for cars. Lowering the mezzanine decks usually results in imposing more stringent height restriction in another part of the ferry and consequently can have implications for freight traffic. It would be useful to provide a method that allows for these different configurations and suggests which is optimal.

In forthcoming work, we have solved this pricing problem to optimality for a smaller number of vehicle types but solving to optimality can take considerable computational time and consequently must be carried out offline. We believe that the work presented here, with some adaptation, will be useful when considering the online packing problem.

ACKNOWLEDGMENTS

This work was funded by the EPSRC under grant number EP/N006461/1. We are also grateful to Alexander Armstrong and Lee Hudson at Red Funnel, UK who provided us with useful practical advice when designing this method.

REFERENCES

Amaruchkul, K., W. Cooper, and D. Gupta. 2007. "Single-Leg Air Cargo Revenue Management". *Transportation Science* 41(4):457–469.

Bertsekas, D. P., and J. N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific.

Burley, D. 1974. *Studies in Optimization*. International Textbook Co. Ltd.

Csirik, J., and G. J. Woeginger. 1998. *Online Algorithms: The State of the Art*, Chapter On-line Packing and Covering Problems, 147–177. Berlin, Heidelberg: Springer Berlin Heidelberg.

Gonzalez-Martin, S., B. B. Barrios, A. A. Juan, and D. Riera. 2014. "On the Use of Biased Randomisation and Simheuristics to Solve Vehicle and Arc Routing Problems". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 1875–1884. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Han, D., L. Tang, and H. Huang. 2010. "A Markov Model for Single-Leg Cargo Revenue management under a bid-price policy". *European Journal of Operational Research* 200:800–810.

J.S.Billings, A.G. Diener, B. Y. 2003. "Cargo Revenue Optimization". *Journal of Revenue and Pricing Management* 2:69–79.

Kasilingam, R. 1996. "Air Cargo Revenue Management: Characteristics and Complexities". *European Journal of Operational Research* 96:36–44.

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

Littlewood, K. 2005. "Forecasting and Control of Passenger Bookings". *Journal of Revenue and Pricing Management* 4 (2): 111–123.

Lodi, A., S. Martello, and M. Monaci. 2002. "Two Dimensional Bin Packing Problems: a Survey". *European Journal of Operational Research* 141:241–252.

Maddah, B., L. Moussawi-Haidar, M. El-Taha, and H. Rida. 2010. "Dynamic Cruise Ship Revenue Management". *European Journal of Operational Research* 207:445–455.

Mallor, F., C. Azcárate, and P. Mateo. 2015. "Operational Management of Renewable Energy Systems with Storage Using an Optimisation-Based Simulation Methodology". *Journal of Simulation* 9:263–278.

Pérez-Rosés, H., and F. Sebé. 2015. "Synthetic Generation of Social Network Data with Endorsements". *Journal of Simulation* 9:279–286.

Philips, R. L. 2005. *Pricing and Revenue Optimization*. Stanford University Press.

Powell, W. B. 2007. *Approximate Dynamic Programming*. Wiley.

Talluri, K. T., and G. J. V. Ryzin. 2004. *The Theory and Practice of Revenue Management*. Kluwer Acedemic Publishers.

Wong, W., A. Zhang, Y. Hui, and L. Leung. 2009. "Optimal Baggage-Limit Policy: Airline Passenger and Cargo Allocation". *Transportation Science* 43(3):355–369.

Zou, L., C. Yu, and M. Dresner. 2013. "The Application of Inventory Transshipment Modeling to Air Cargo Revenue Management". *Transportation Research Part E* 57:27–44.

AUTHOR BIOGRAPHIES

CHRISTOPHER BAYLISS is a Research Assistant in the CORMSIS Research Group at the University of Southampton, UK. Christopher obtained his first degree in Mathematics at the University of Derby, and his his Ph.D. at the University of Nottingham. Christopher has published conference papers and a journal paper related to Airline Reserve Crew Scheduling, which was the subject of his Ph.D. His research interests include scheduling, simulation, heuristics, revenue management and optimization. Email: c.bayliss@soton.ac.uk.

JULIA M. BENNELL is Professor of Management Science in Southampton Business School at the University of Southampton, UK. Her research interests include cutting and packing, scheduling and routing problems and she has expertise in using both heuristic and exact optimization methods. She is most well known for her work in cutting and packing and has been an executive member of the European Working Group in Cutting and Packing since 2004. She has given numerous invited talks in cutting and packing and chaired two conferences. Email: J.A.Bennell@soton.ac.uk.

CHRISTINE S.M. CURRIE is Associate Professor of Operational Research in Mathematical Sciences at the University of Southampton, UK, where she also obtained her Ph.D. She is Editor-in-Chief for the Journal of Simulation. Christine was a Track Coordinator at WSC 2013 and has 9 published WSC papers. She chaired the UK Simulation Workshop, SW16. Her research interests include mathematical modeling of epidemics, Bayesian statistics, revenue management and optimization of simulation models. Email: christine.currie@soton.ac.uk.

ANTONIO MARTINEZ-SYKORA is a Lecturer in Business Analytics at the University of Southampton. He finished the PhD in June 2013 at the University of Valencia (Spain). He was chair of the XIII ESICUP conference and is program coordinator of the OR58 conference. His research interests include mathematical modeling and optimization in the area of logistic planning, cutting and packing problems and revenue management. Email: A.Martinez-Sykora@soton.ac.uk

MEE-CHI SO is a Lecturer at the Southampton Business School. She has expertise and publications in data mining, credit scoring and marketing analytics. She has participated in a number of consultancy projects to collaborate with a wide range of organizations such as Financial Conduct Authority, Global Radio, United Health, etc. Email: m.so@soton.ac.uk.