UNIVERSITY OF Southampton
School of Mathematics

# Using Simulation To Solve The Vehicle Ferry Revenue Management Problem

Chris Bayliss, Julia Bennell, **Christine Currie**, Antonio Martinez-Skyora, Mee-Chi So

OR58, September 2016

This work was funded by the EPSRC under grant number EP/N006461/1

UNIVERSITY OF Southampton
School of Mathematics

# Talk Overview

1) Problem description
2) Loading simulator
3) Dynamic pricing formulation
4) Results
5) Conclusions and future work

UNIVERSITY OF Southampton
School of Mathematics

# PROBLEM DESCRIPTION

UNIVERSITY OF Southampton
School of Mathematics

# Problem description

**Objective:** derive a dynamic pricing policy that maximises the expected revenue from the sale of vehicle tickets on a ferry

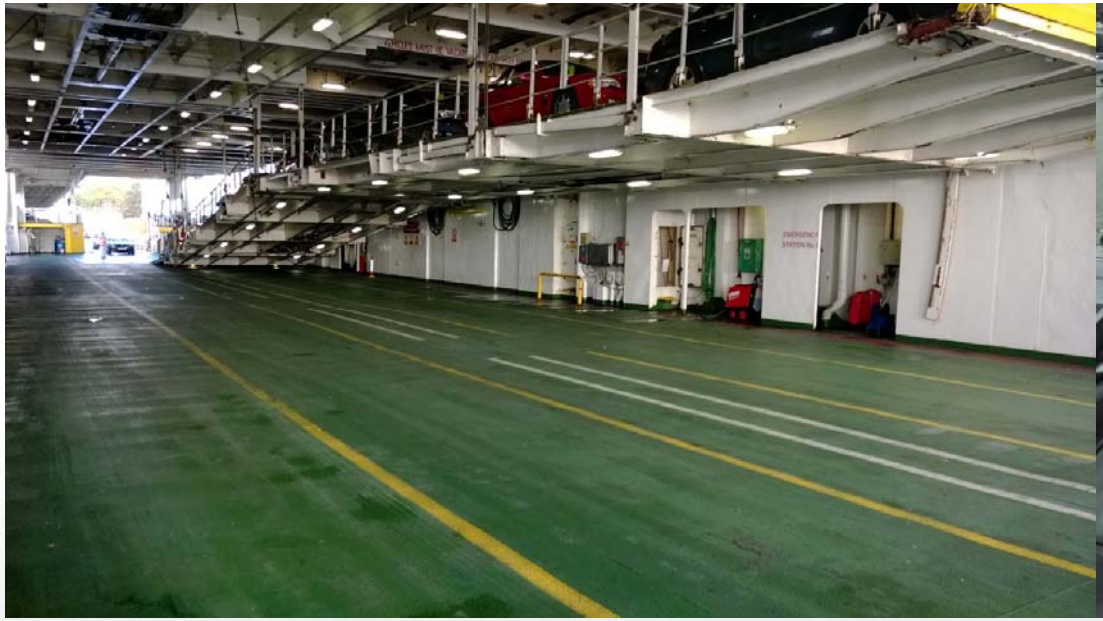
- **Constraint:** Limited **capacity** which **depends on packing**
- Customers
  - Arrive at random during the **selling season** (beginning 6 months before departure)
  - Customer **willingness to pay** is related to vehicle size and time until departure
  - Their vehicles vary in **shape and size**

# Case study

- Red Funnel: regular crossings between Southampton and the Isle of Wight
- Vehicles: private vehicles (cars, vans, caravans, trailers, mopeds, ...) and commercial freight vehicles
- Decks:
  - Car deck (cars and motorbikes only)
  - Main deck (all vehicle types)
  - 2 Mezzanine decks (movable dependent on traffic)
- Lanes: parking in lanes is always possible on the car deck but not on the main deck due to wide vehicle types

UNIVERSITY OF Southampton
School of Mathematics

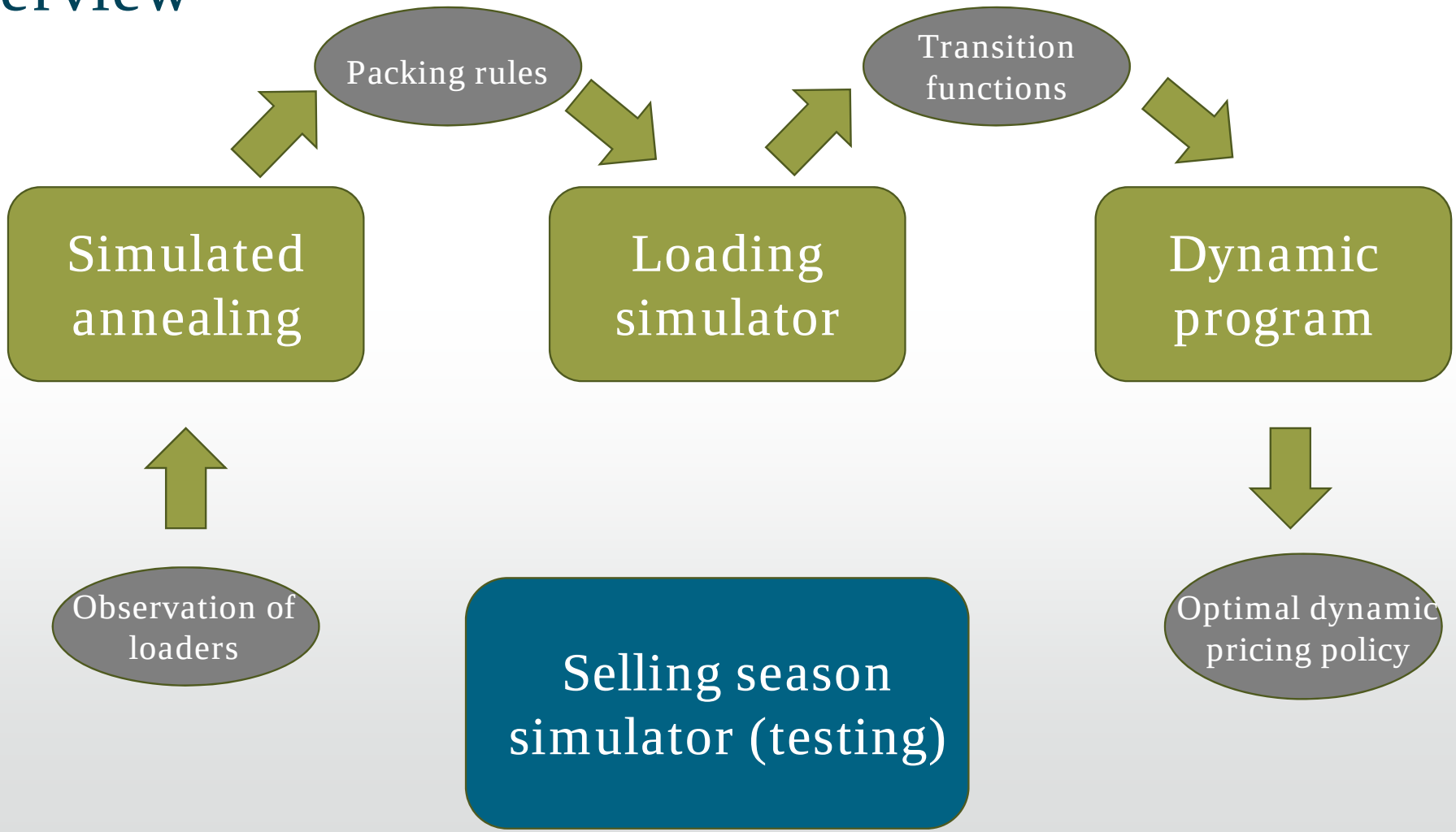
# Goal is to solve a real world instance



- The **dimensionality** of the proposed formulation is the number of vehicle types
- The number of possible vehicle combinations also rises exponentially with ferry capacity
  - We can solve this instance exactly for up to 5 vehicle types using IP for packing and dynamic programming

UNIVERSITY OF Southampton
School of Mathematics

# Overview



UNIVERSITY OF Southampton
School of Mathematics

# LOADING SIMULATOR

UNIVERSITY OF Southampton
School of Mathematics

# Demo – wish me luck!

UNIVERSITY OF Southampton
School of Mathematics

# Loading Simulator

- Simulates the online vehicle ferry loading process
- Loading rules:
  - Optimised by simulated annealing
  - In future we will develop rules that mimic real loading
- Measures the remaining space on each deck after each vehicle is loaded
- Accounts for the parking gaps that are required for passengers to exit (and subsequently re-enter) the vehicle decks

UNIVERSITY OF Southampton
School of Mathematics

# Optimising the loading rules

- The loading algorithm is used to select which vehicle to load next and where
- Possible positions are generated and ordered in terms of a weighted sum of a number of efficiency based criteria
- Example attributes:
  - Distance from the far end of the ferry
  - Tightness (vehicle width/parking position width)
  - Parking loss (space lost due to staggered parking)
- Weights are set via simulated annealing

UNIVERSITY OF Southampton
School of Mathematics

# DYNAMIC PRICING FORMULATION

UNIVERSITY OF Southampton
School of Mathematics

# Notation

- T : Number of time intervals in the selling season
- $t$ : Time period, $t \in \{T, T-1, \dots, 1,0\}$
- $I$ : Set of vehicles types
- $\lambda_i$: Arrival rate for vehicle type $i \in I$
- $P = \{P_0, P_1, \dots, P_{max}\}$: set of available price points
- $\alpha_{i,p,t}$: Probability that a customer with vehicle type $i \in I$ accepts price $p \in P$ at time period $t \in \{T, T-1, \dots, 1,0\}$
- $X$ : Current state/accepted vehicle mix/sales history
- $X'$ : Next state (after a sale)

UNIVERSITY OF Southampton
School of Mathematics

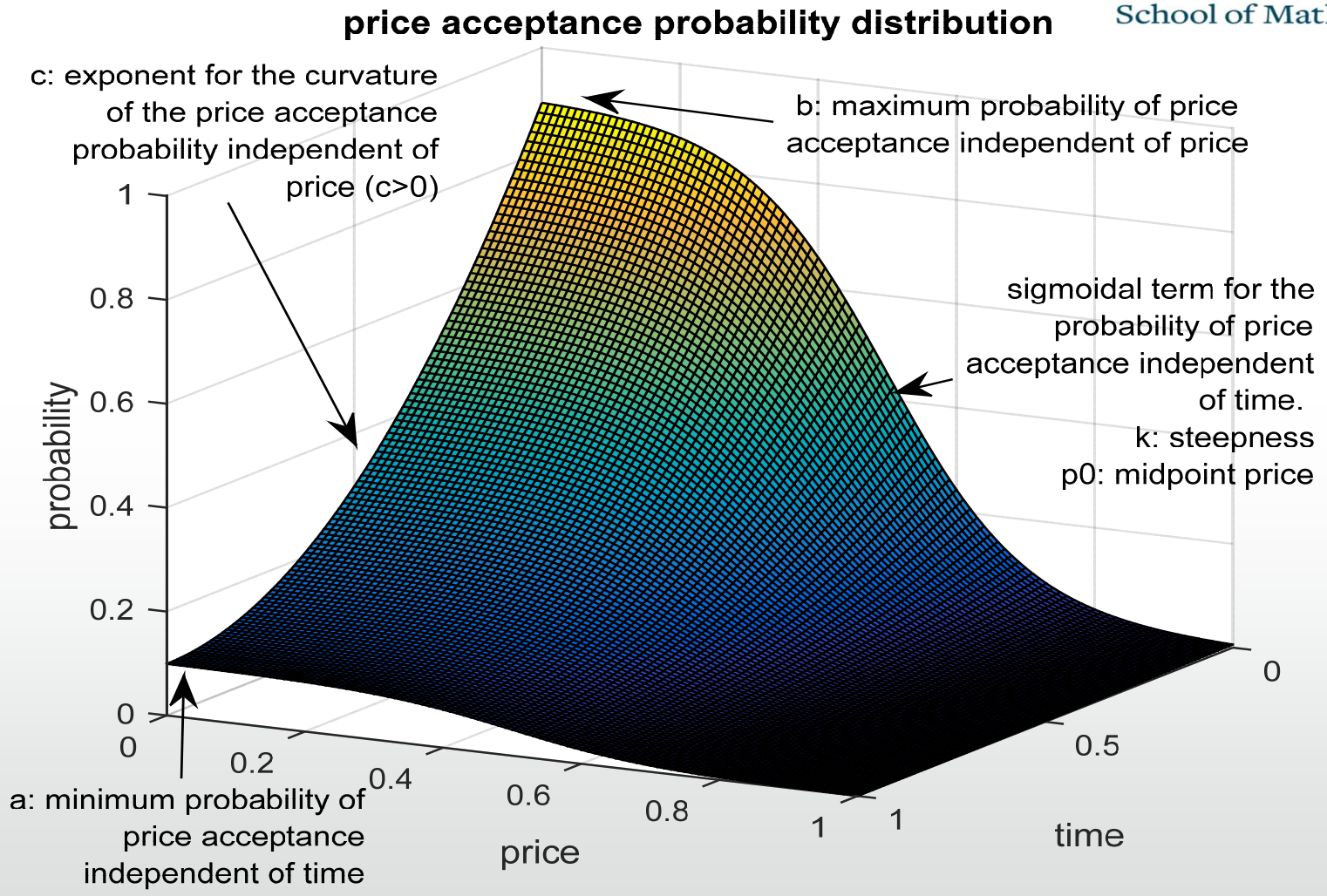
# Formulation

- $V_{t,s}$: optimal expected revenue from period $t$ to the end of the selling season if the current state is X
- Yields the price points for all vehicles, times and states that maximise the revenue

$$V_{t,s} = \max_{p \in P} \left\{ \left\{ \sum_{i \in I} \lambda_v \left\{ \alpha_{i,p,t} \left( p + V_{t-1,F(s,i)} \right) + \left( 1 - \alpha_{i,p,t} \right) V_{t-1,s} \right\} \right\} + \lambda_0 V_{t-1,s} \right\}$$

UNIVERSITY OF Southampton
School of Mathematics

$$\alpha_{i,p,t} =$$



UNIVERSITY OF Southampton
School of Mathematics

# Simheuristic Approach

- States are defined by **remaining area**
- Define **transition functions** to specify the amount of space used by each vehicle type
- The transition functions are derived from a custom built **ferry loading simulator**

UNIVERSITY OF Southampton
School of Mathematics

# State transitions

- $s$ : **Current state**, defined as the remaining space on the upper deck ($r_u$), the remaining space for low vehicles ($r_l$) and the remaining space for high vehicles on the main deck ($r_h$)
- $s'$: **Next state**, $F(s,i)$ denotes how the next state depends on the current state and which vehicle has arrived and purchased a ticket for the ferry

$$s = \{r_u, r_l, r_h\}$$

$$s' = F(s,i) = \begin{cases} r_u \leftarrow r_u - f_u(i) : \text{if vehicle fits on the upper deck} \\ r_l \leftarrow r_l - f_l(i) \\ r_h \leftarrow r_h - f_h(i) \end{cases} : otherwise$$

UNIVERSITY OF Southampton
School of Mathematics

# Numerical examples of state transitions

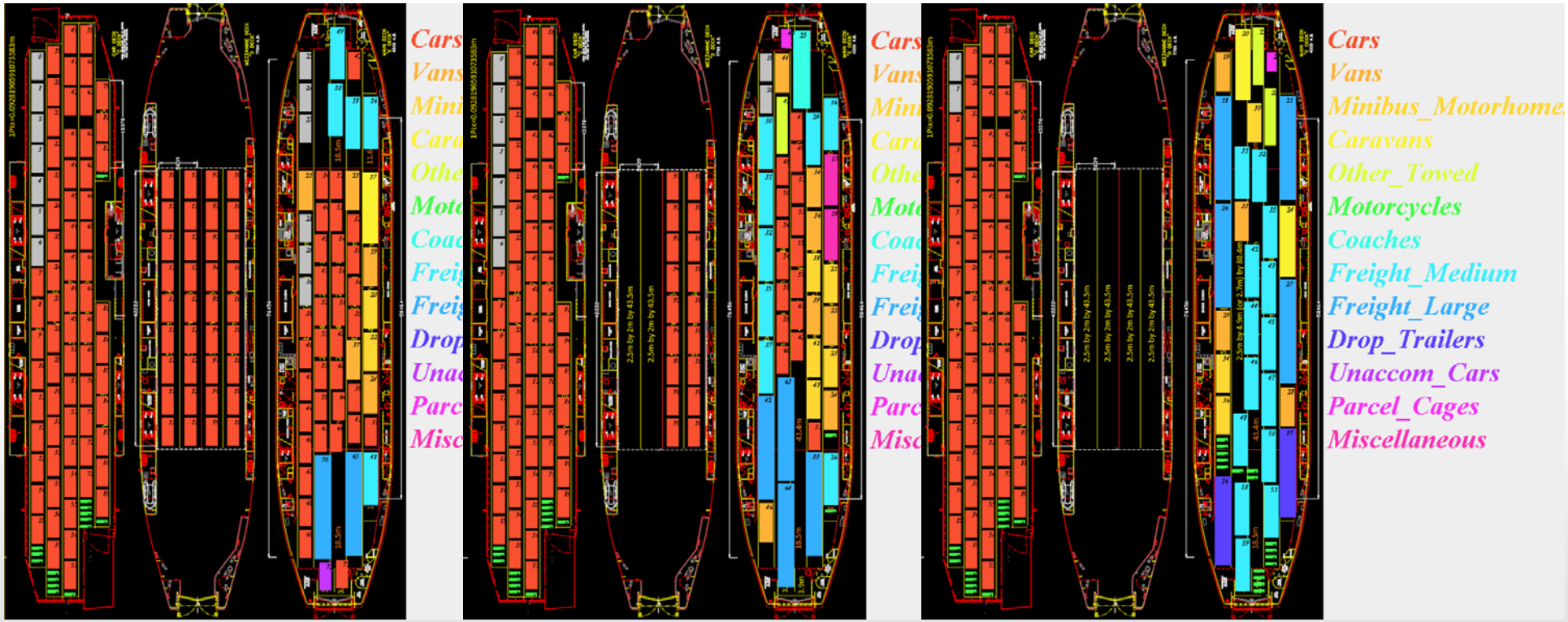
- Empty ferry state: $s = \{800,1000,600\}$ (units in metres squared)
- Selling season transitions
  - Car purchases a ticket: $\{800,1000,600\} - \{15,0,0\} = \{785,1000,600\}$, i.e. the car uses $15\text{m}^2$ on the upper deck
  - Then a van purchases a ticket: $\{785,1000,600\} - \{0,20,10\} = \{785,980,590\}$, i.e. the van is parked on the main deck half under a mezzanine deck
  - Then a large freight vehicle purchases a ticket: $\{785,980,590\} - \{0,60,60\} = \{785,920,530\}$, the large freight vehicle is parked in high vehicle space (not under a mezzanine deck), which fully overlaps with the space available to low vehicles

UNIVERSITY OF Southampton
School of Mathematics

# RESULTS

UNIVERSITY OF Southampton
School of Mathematics

# Deck configurations and demand scenarios




High car demand
2 Mezzanine decks

Medium demand
1 Mezzanine deck

High freight demand
0 Mezzanine decks

UNIVERSITY OF Southampton
School of Mathematics

# Total revenues for different ferry configurations in different demand scenarios

| Demand scenario | 0 Mezzanine decks | 1 Mezzanine deck | 2 Mezzanine decks | Non-fixed deck configuration |
|---|---|---|---|---|
| High car | 67.028 | *70.590* | 64.713 | ***71.332*** |
| Medium | 62.392 | *65.448* | 58.807 | ***65.579*** |
| High freight | *57.449* | 53.536 | 42.219 | ***57.752*** |

| Demand scenario | Best Deck Configuration (% revenue compared to full dynamic pricing) |
|---|---|
| High car | 96.5% |
| Medium | 98.5% |
| High freight | *97.6%* |

UNIVERSITY OF Southampton
School of Mathematics

# Capacity based pricing

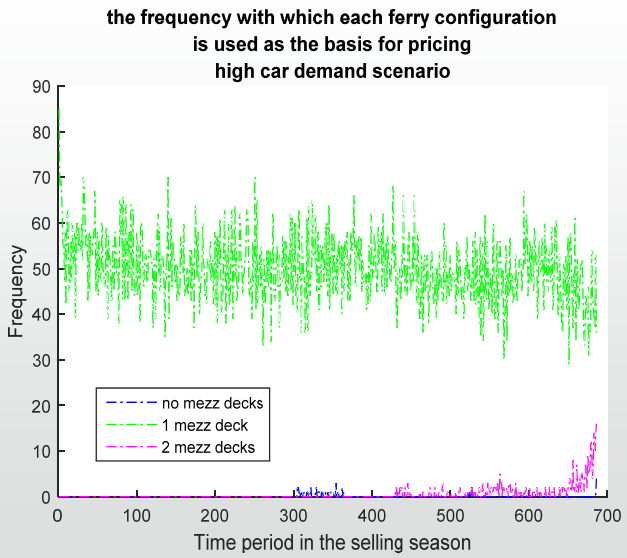
| Demand scenario | Best Deck Configuration (% revenue compared to full dynamic pricing) |
|---|---|
| High car | 96.5% |
| Medium | 98.5% |
| High freight | *97.6%* |

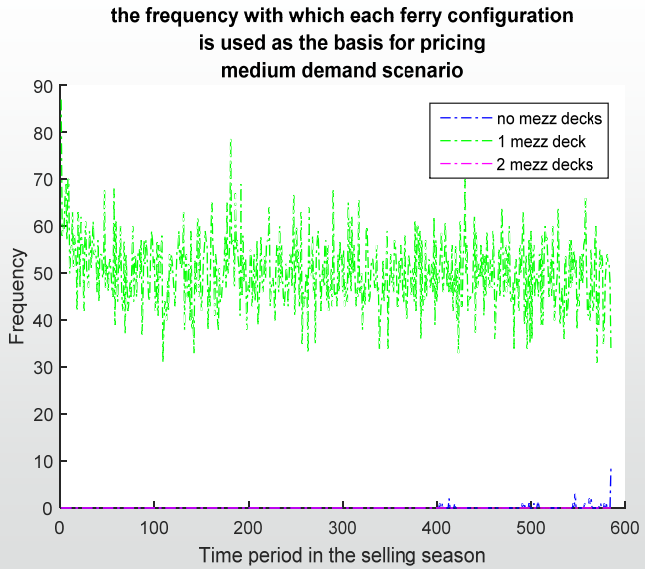
- The capacity based pricing policy has a single price for each vehicle type for each level of remaining space
- Derived from the optimal dynamic pricing policy using the expected demand trajectory

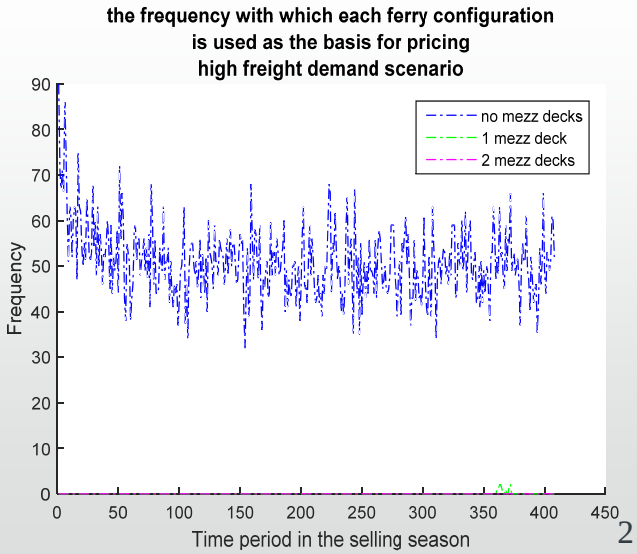
UNIVERSITY OF Southampton
School of Mathematics

# Non-fixed ferry configuration solution use frequencies

| Demand scenario | 0 Mezzanine decks | 1 Mezzanine deck | 2 Mezzanine decks |
|---|---|---|---|
| High car | 42 | 34325 | 348 |
| Medium | 34 | 29417 | 0 |
| High freight | 20787 | 12 | 0 |

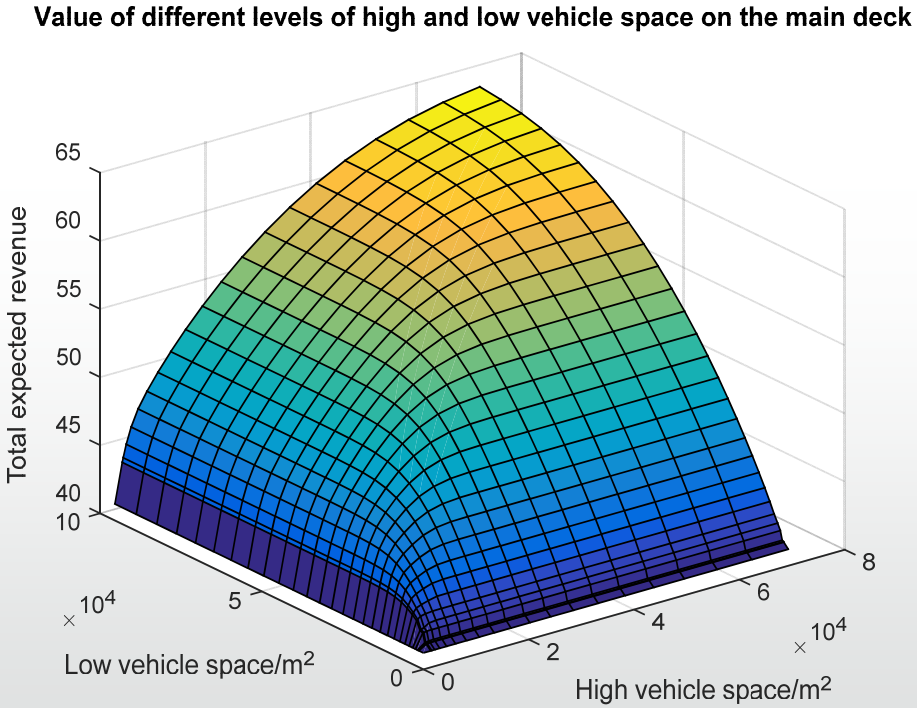


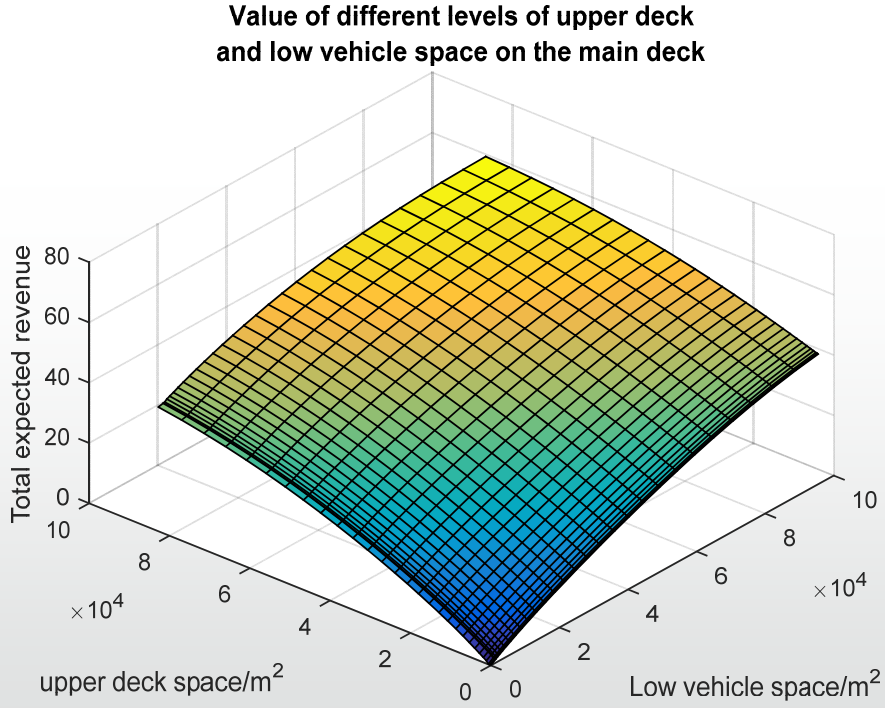



UNIVERSITY OF Southampton
School of Mathematics

# Piecewise value function approximation

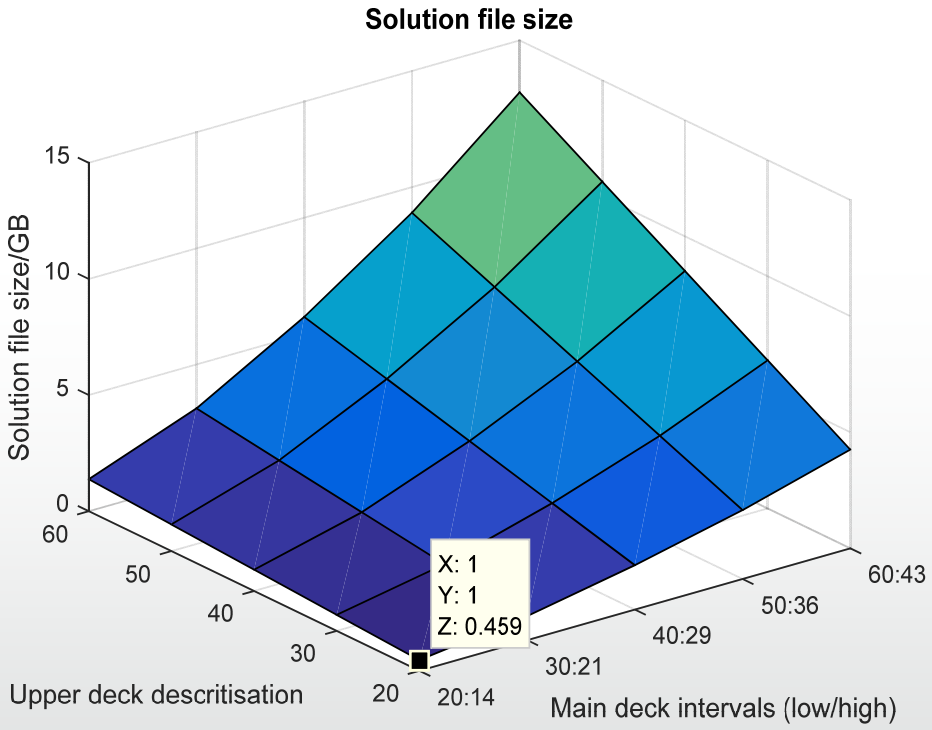


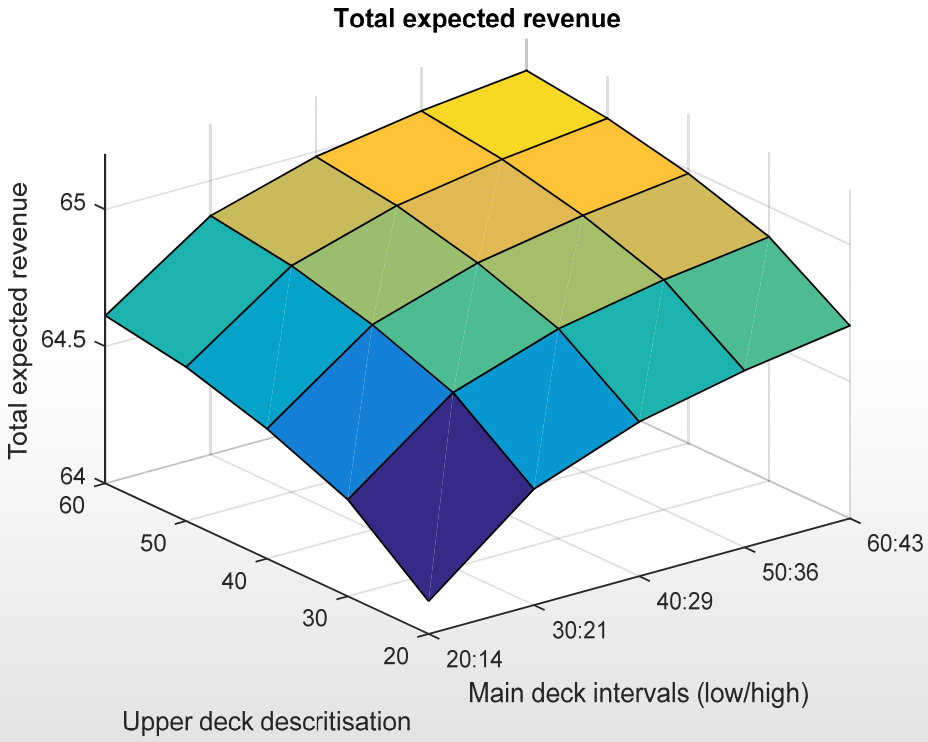




The values of intermediate states are interpolated

UNIVERSITY OF Southampton
School of Mathematics

# Interval size effects: discretization



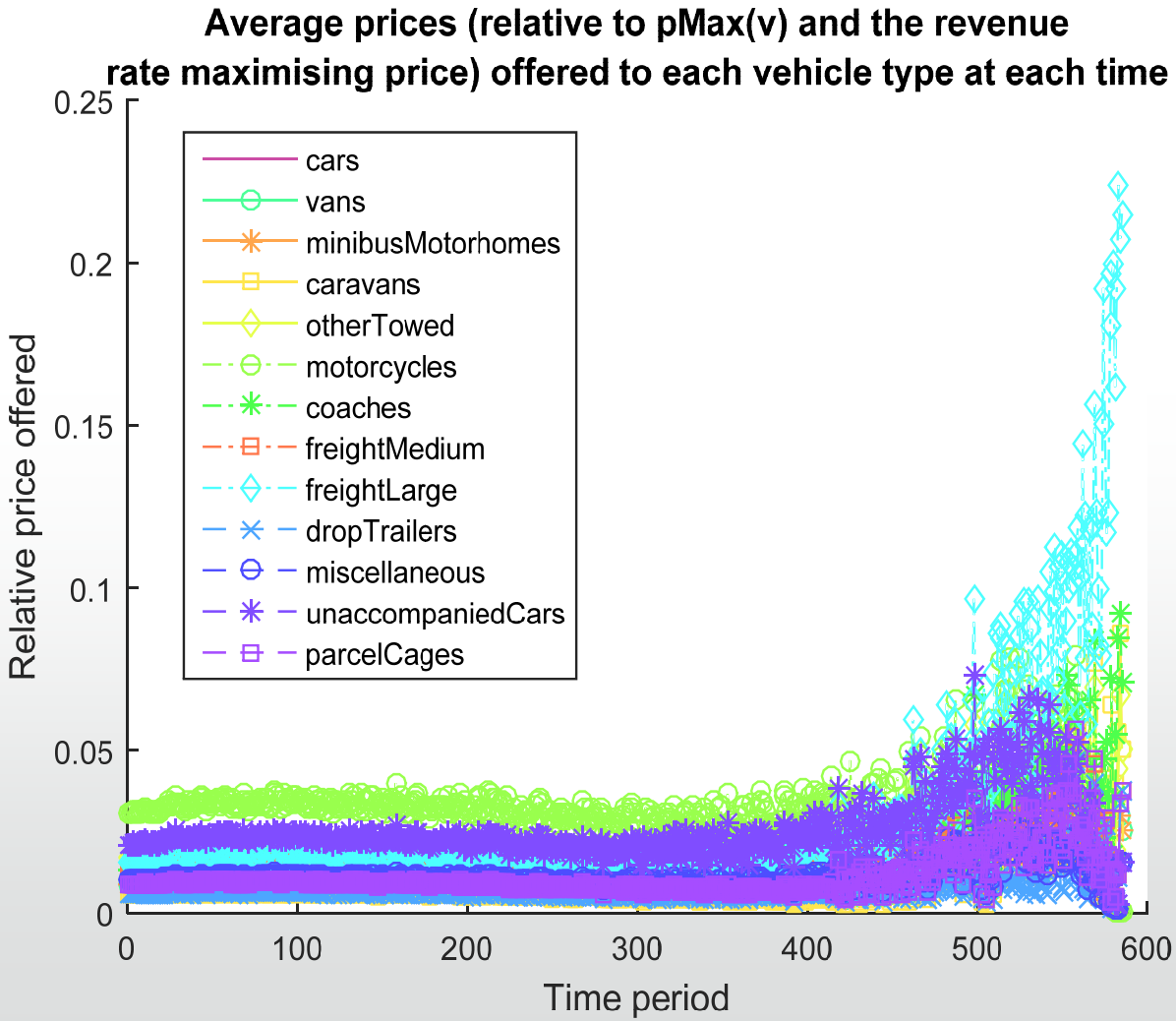




Solution time graphs looks the same as this
Min:3 minutes (3 million prices)
Max:2 hours (90 million prices)

64.12 for the coarsest discretisation
65.09 for the finest discretisation

UNIVERSITY OF Southampton
School of Mathematics

# Prices



UNIVERSITY OF Southampton
School of Mathematics

# CONCLUSIONS AND FUTURE WORK

UNIVERSITY OF Southampton
School of Mathematics

# Conclusions

- This approach should (at the very least) make the ferry companies' models of vehicle capacity more realistic.
- Explicitly models the effect of the packing method on the ferries' capacity.
- The use of the loading simulator to track the state in the selling season allows the approach to take the exact effects of the realised vehicle demand scenario into account.
- Allows the realised demand to determine what ferry configuration will be most profitable

UNIVERSITY OF Southampton
School of Mathematics

# Future work

- Run simulated annealing for longer with more repeats and on more demand scenarios to further improve the loading algorithm parameters
- Fit the packing rules to actual loaders using simulated annealing
- Compare to existing practices (which do not include the time remaining until departure)
- Improve the transition value estimation
- Non-linear interpolation approach could make coarse discretisation work as well as a fine discretisation

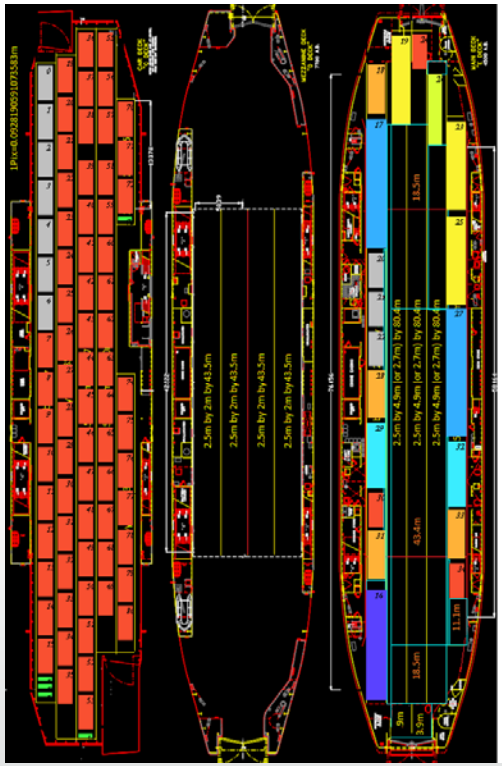
UNIVERSITY OF Southampton
School of Mathematics

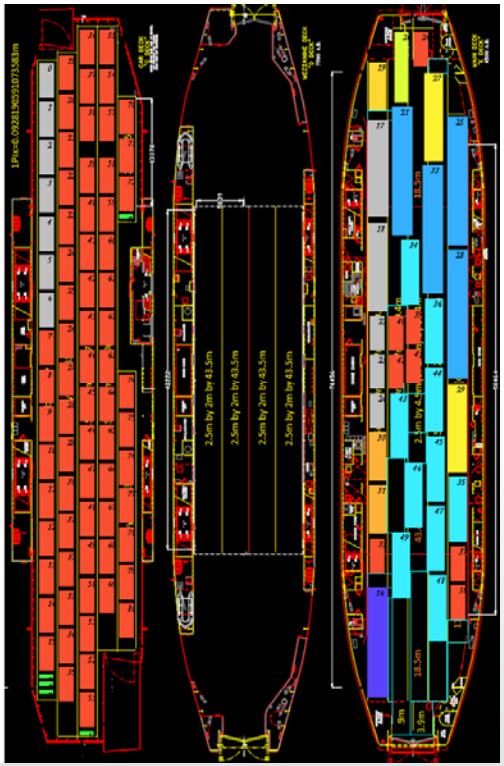
# QUESTIONS?

UNIVERSITY OF Southampton
School of Mathematics

# Space use dependence upon remaining space

When the ferry is empty vehicle are generally loaded efficiently without the creation of unusable gaps



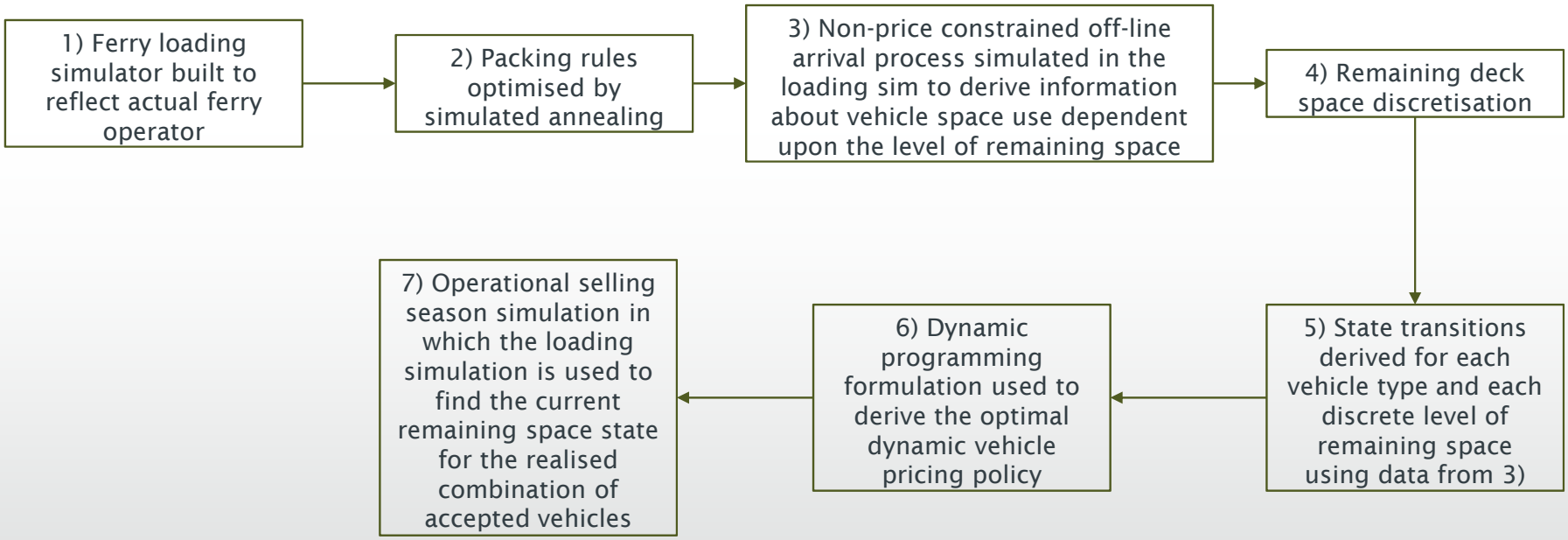


When the ferry is full vehicles tend to cause unusable gaps

The transition functions capture such effects.
A transition function graph will be given later on.

UNIVERSITY OF Southampton
School of Mathematics

# Overview



UNIVERSITY OF Southampton
School of Mathematics

# Simulated annealing for optimising the loading rules

- Let
  - R: Online remaining space after loading all queued vehicles
  - G: Number of unreachable gaps in which a minimum dimensioned vehicle fits
  - U: Vector containing counts of the unloaded queued vehicles of each type
  - W: Vector containing the weights of the
  - c: weight given to minimising unreachable gaps
  - d: vector of penalties for not loading vehicles (one for each type)
- Objective: $\max_{W}\{R - cG - d \cdot U\}$

UNIVERSITY OF Southampton
School of Mathematics

# Simulated annealing algorithm details

- The weight given to each attribute varies linearly between two values dependent upon the level of remaining space
- This approach allows the behaviour of the loading algorithm to vary over the course of loading
- During the SA algorithm one or both of the weight corresponding to a particular attribute are modified
- The parameter modifications can be random or +/- additive or multiplicative steps
- The probability of random parameter modification decreases over the course of the algorithm

UNIVERSITY OF Southampton
School of Mathematics

# Price acceptance model

- **Sigmoidal in price non-linear (or linear) in time**
- $\alpha_{p,t} = cf\left(1 - \left(\frac{1}{1+e^{-k\left(\frac{p}{pMax} - pr_0\right)}}\right)\right) \times \left(a + (b-a)\left(1 - \frac{t}{T}\right)^c\right)$
- $cf = \left(\frac{1}{\left(1 - \left(\frac{1}{1+e^{k \cdot pr_0}}\right)\right)}\right)$

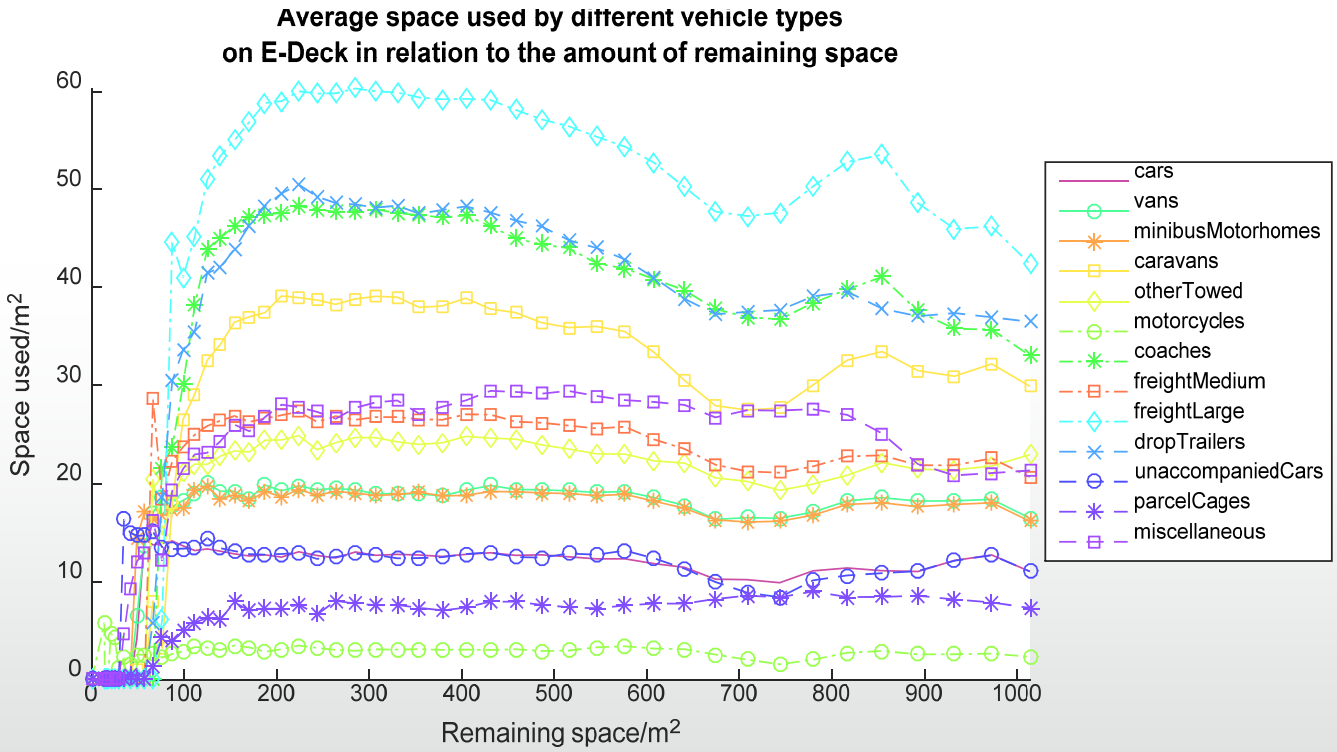
| Parameter | Interpretation |
|---|---|
| a | The probability of price acceptance at the beginning of the selling season at price 0 |
| b | The probability of price acceptance at the end of the selling season at price 0 |
| c | Curvature of the effect of time on the probability of price acceptance |
| k | Steepness of the midpoint of the sigmoidal price part of the function |
| $pr_0$ | Relative position of the midpoint of the sigmoidal part of the function |
| pMax | Maximum price a random customer will pay |

UNIVERSITY OF Southampton
School of Mathematics

# Transition values for 0 mezzanine deck case

When little space remains on the main deck:

- Some vehicle types can no longer be added
- Some vehicle types have the effect of recapturing lost space, as vehicles will packed differently if there are different numbers of vehicles to be packed




In general the amount of space used by each vehicle type increases as remaining space decreases, because packing becomes more awkward

UNIVERSITY OF Southampton
School of Mathematics

# Experimental results

- 3 demand scenarios
  - High car demand
  - Medium (car and freight) demand
  - High freight demand
- 3 ferry configurations
  - 0 Mezzanine decks
  - 1 Mezzanine deck
  - 2 Mezzanine decks
- Approach applied to each combination of the above

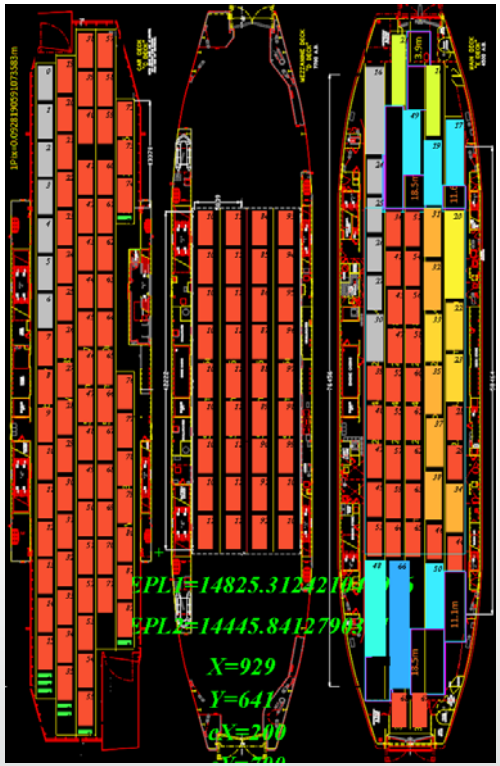
UNIVERSITY OF Southampton
School of Mathematics

# Another picture

In a two Mezzanine deck scenario the loading algorithm rules try to place low vehicle under the mezzanine deck.

Sometime this causes unused space.

The main point is that Mezzanine decks are only appropriate in situations where low vehicle demand is especially high and high vehicle demand is especially low




As deck decision depend on demand scenarios we segment demand scenarios in terms of ratios of spatial demand of low and high vehicles and derive loading rules and then pricing policies for each of these.

A solution is derived for each demand scenario in each ferry configuration

UNIVERSITY OF Southampton
School of Mathematics

- A table of expected revenues for each demand scenario with each ferry configuration
- And possible another column where the configuration is not fixed but instead the price from the configuration with the highest expected future demand is used.
- In each case the demand scenario is known, in some cases statistical fluctuations mean that the demand scenarios overlap, allow us to take advantage of the addition car capacity that the Mezzanine decks offer.

UNIVERSITY OF Southampton
School of Mathematics

# Finding the Exact Solution

- Exact optimal dynamic pricing formulation
  - Integrates **packing** and **dynamic pricing**
  - Integer programming approach to solve the packing problem (1-D bin-packing formulation)
  - The states of the dynamic program consist of the set of all possible vehicle mixes that could fit onto the ferry
  - Becomes **intractable** for more than a handful of vehicle types