Dynamic Pricing in the Vehicle Ferry Industry

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

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

- Problem description
- General framework for integrating packing and pricing
- Packing models (Exact and Simulation based state definitions)
- Price acceptance model
- Results
- Conclusions and future work
PROBLEM DESCRIPTION
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 efficiency**

- Customers
  - Arrive at random during the **selling season** (beginning 6 months before departure)
  - Customer **willingness-to-pay** is dependent on time until departure and varies between vehicle types
  - Vehicles vary in **size**
Selling Tickets

Start of selling season

Discrete time

At most one customer arrival per time period

Departure
General framework for integrating packing and pricing

Packing algorithm
1. 1D Bin Packing
2. 2D Packing Heuristic

Capacity model

Pricing algorithm
1. Dynamic programming
2. Approximate dynamic programming

States capturing the sold/remaining capacity
General framework for integrating packing and pricing

• Input variables
  
  – The state $s$ at any given time interval captures the ferries remaining capacity for vehicles. The key question is how to define $s$, we consider exact and approximate approaches
  
  – $s'$ denotes the new remaining capacity state after one sale whilst in state $s$
  
  – $V_t(s)$ denotes the ‘revenue-to-go’ or the expected future revenue if the state is $s$ at time $t$
  
  – $\lambda_{t,i}$ denotes the probability that a customer with vehicle type $i$ arrives at time $t$
General framework for integrating packing and pricing

- **Input functions**
  - **Price acceptance function:** \( \alpha(i, p, t) \) returns the probability that a customer with vehicle type \( i \) will pay a price \( p \) at time \( t \)
  - **Transition function:** \( f(s, i) \) returns the remaining capacity capturing state \( s' \) if a customer with vehicle type \( i \) purchases a ticket at a time when the state is \( s \) (derived from packing models)
General framework for integrating packing and pricing

- Dynamic pricing formulation
  - The optimal dynamic pricing look-up-table policy can be derived by computing the Bellman equations by backwards recursion
  - In each state at each time 3 events can occur
    1. No customers arrive
    2. A customer arrives but does not purchase a ticket
    3. A customer arrives and purchases a ticket

\[
V_t(s) = \max_{p \in P} \left\{ \sum_{i \in I} \lambda_{t,i} \{ \alpha(i, p, t)(p + V_{t-1}(f(s, i))) + (1 - \alpha(i, p, t))V_{t-1}(s)\} + \lambda_{t,0}V_{t-1}(s) \right\}
\]
Exact and Simulation-based state definitions

PACKING MODELS
Exact and Simulation based state definitions

**Exact**
- 1-d bin packing model (optimal lane parking)
- State=count of vehicles of each type
- 1-dimension per vehicle type
- Tickets sold state definition

**Simulation based**
- 2-d packing heuristic (ignores lanes)
- State=remaining deck area per deck region
- 1-dimension per deck region
- Remaining space state definition

Transition functions for one vehicle type 0 sale

\[ \text{Transition functions} \]

\[
\begin{align*}
\text{state} &= 3,2,1,1,4 \\
\text{state} &= A = 950.8, \\
\text{transition value} &= 26.2 \\
\end{align*}
\]

\[
\begin{align*}
f \left( \{3,2,1,1,4\}, 0 \right) &= \{3,2,1,1,4\} + \{1,0,0,0,0\} = \{4,2,1,1,4\} \\
f(950.8, 0) &= 950.8 - 26.2 = 924.6
\end{align*}
\]
Exact (1D Bin Packing)

- Assumes that vehicles can be allocated to lanes that they fit within
- A fast IP formulation is used to enumerate all possible vehicles mixes
Pareto front of vehicle mixes

Vehicle type 2 > vehicle type 1

Capacity envelope = states

(0,5) (1,4) (3,3) (5,2) (7,1) (8,0)
Loading Simulator

- Simulates the vehicle ferry loading process for a known set of vehicles
- **2D packing problem** on the main deck, as not all vehicle types fit within the lanes
- **Sequential** weighted sum loading algorithm for placing vehicles (simulated annealing is used to tune the weights to increase packing efficiency)

**Real world constraints:**
- Manoeuvrability; lift access; mezzanine decks; drop trailers which are towed onto the ferry, parking gaps and also reverse gaps

**Purpose:**
- Map vehicle mix **states** to lower dimensional **remaining space states**
- Generate efficient **2D packing solutions**
- Prevent overselling (assuming 100% show rate)
Mapping a vehicle mix state to a remaining space state

Available parking positions

Remaining area which is used to map vehicle mix state to remaining space state
Simulation based approach state transition functions

- **Transition functions** specify the average amount of *area used by each vehicle* type including area lost due to staggered parking (parking loss)
- Transition functions are **derived from** the transitions that occur in a large sample of simulated vehicle loads
- The **loading efficiency** of each simulated vehicle load maximised via a simulated annealing algorithm
Approximating the value function (Simulation based approach)

- The remaining space is continuous but we solve the value function for a **discrete** set of **remaining space states**

- Transitions from the discrete states lead to **intermediate states**

- The values of intermediate states are **interpolated** from the values of neighbouring states

\[ V_t(s) = c(\text{linear interpolation}) + (1 - c)(\text{gradient based interpolation}) \]
The concave structure of the value function is exploited to speed up the solution time of the dynamic program.
Value interpolation

\[ V(s) = x + \frac{2}{1} \left( \frac{V(s) - V(s-1)}{1} \right) \]

\[ V(s) = x + \frac{z}{1} \left( \frac{V(s) - V(s-1)}{1} \right) \]

\[ \frac{\partial}{\partial s} V(s) \approx \frac{V(s) - V(s-1)}{1} \]

\[ \frac{\partial}{\partial s} V(s) \approx \frac{V(s) - V(s-1)}{1} \]

\[ z = \frac{2}{1} \left( \frac{V(s) - V(s-1)}{1} \right) \]

\[ x = \frac{\partial}{\partial s} V(s) \]

\[ y = V(s) \]

\[ x = \frac{\partial}{\partial s} V(s) \]
Price acceptance model (for both models)

- Accounts for bell shaped WTP distribution and monotonic time effects

\[ \alpha(p, t) = cf \left( 1 - \left( \frac{1}{1+e^{-k \left( \frac{p}{p_{Max}} - m \right)}} \right) \right) \times \left( a + (b - a) \left( 1 - \frac{t}{T} \right)^c \right) \]

- \( cf = \left( \frac{1}{\left( 1- \left( \frac{1}{1+e^{k\cdot m}} \right) \right)} \right) \)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>The probability of price acceptance at the beginning of the selling season at price 0</td>
</tr>
<tr>
<td>b</td>
<td>The probability of price acceptance at the end of the selling season at price 0</td>
</tr>
<tr>
<td>c</td>
<td>Curvature of the effect of time on the probability of price acceptance</td>
</tr>
<tr>
<td>k</td>
<td>Steepness of the midpoint of the sigmoidal price part of the function</td>
</tr>
<tr>
<td>m</td>
<td>Relative position of the midpoint of the sigmoidal part of the function</td>
</tr>
<tr>
<td>pMax</td>
<td>Maximum price a random customer will pay</td>
</tr>
</tbody>
</table>
\[ \alpha(i, p, t) = \]

- **c**: exponent for the curvature of the price acceptance probability independent of price (c > 0)
- **a**: minimum probability of price acceptance independent of time
- **b**: maximum probability of price acceptance independent of price
- **k**: steepness
- **p0**: midpoint price

**price acceptance probability distribution**

- Sigmoidal term for the probability of price acceptance independent of time.
Experiments

• Exact
  – The impact of

• Simulation based
  – 13 vehicle types, real ferry design with a car deck and a main deck with up to 2 mezzanine decks, \( p_{\text{Max}}(i) \propto area_i \), 3 deck configurations, 3 demand scenarios \((\lambda_{t,i})\)

• Exact versus Simulation based
  – Simulation based approach implemented with 1D and 2D packing approaches in the Exact test instances
Experiments

• Exact
  – 5 vehicle types, 3 lane types, \( pMax(i) \propto \sqrt{\text{length}_i}, T=1000 \)

• Simulation based
  – 13 vehicle types, real ferry design with a car deck and a main deck with up to 2 mezzanine decks, \( pMax(i) \propto \text{area}_i \), 3 deck configurations, 3 demand scenarios \( (\lambda_{t,i}) \)

• Exact versus Simulation based
  – Simulation based approach implemented with 1D and 2D packing approaches in the Exact test instances
## Exact Experiment Parameters

### The customers

<table>
<thead>
<tr>
<th>Vehicle type</th>
<th>Arrival rate</th>
<th>Max price (SQRT(L))</th>
<th>Length</th>
<th>Width</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4</td>
<td>1.732051</td>
<td>3</td>
<td>1.6</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>2.236068</td>
<td>5</td>
<td>2.3</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>0.15</td>
<td>2.645751</td>
<td>7</td>
<td>2.4</td>
<td>2.5</td>
</tr>
<tr>
<td>4</td>
<td>0.1</td>
<td>3</td>
<td>9</td>
<td>2.9</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>0.05</td>
<td>3.316625</td>
<td>11</td>
<td>3.4</td>
<td>4</td>
</tr>
</tbody>
</table>

### The ferry

<table>
<thead>
<tr>
<th>Lane type</th>
<th>quantity</th>
<th>Length</th>
<th>Width</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>37.04</td>
<td>2.34</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>37.04</td>
<td>2.93</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>37.04</td>
<td>3.42</td>
<td>5</td>
</tr>
</tbody>
</table>
The interaction between packing and pricing

- X-axis: vehicle mix state sorted by total length of vehicles
- Y-axis: Total expected revenue
- Interpretation: future profit does not strictly monotonically increase with remaining lane space
- This is due to packing effects:
  - Some vehicle mixes lead to unusable gaps at the ends of lanes
  - Our framework will increase the price of sales that lead to such bad states
Vehicle type discretisation

To **reduce the dimensionality** of the problem when using the Exact approach vehicles can be mapped to fewer categories.

1-3-5 discretisation

![Diagram showing 1-3-5 discretisation]

\[ state = 4,3,2,1,1 \]  

(state = 4,5,2)
Vehicle type discretisation results

<table>
<thead>
<tr>
<th>Vehicle type</th>
<th>Discretisation schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 vehicle categories</td>
</tr>
<tr>
<td></td>
<td>a b c d</td>
</tr>
<tr>
<td>1 (3m) (λ=0.4)</td>
<td>3</td>
</tr>
<tr>
<td>2 (5m) (λ=0.2)</td>
<td>5</td>
</tr>
<tr>
<td>3 (7m) (λ=0.15)</td>
<td>7</td>
</tr>
<tr>
<td>4 (9m) (λ=0.1)</td>
<td>9</td>
</tr>
<tr>
<td>5 (11m) (λ=0.05)</td>
<td>11aminer</td>
</tr>
<tr>
<td>Expected revenue</td>
<td>73.97</td>
</tr>
</tbody>
</table>

- Modelling small vehicle types with high arrival rates in detail is essential for maximising revenue.
- Use as many groups of vehicles as possible (but tractability becomes a problem).
- 2 vehicle types 93%, 3 vehicle types 98%, 4 vehicle types 99%: of the optimal revenue without discretisation.
Simulation based approach test instance scenarios

- **High car demand**: 2 Mezzanine decks
- **Medium demand**: 1 Mezzanine deck
- **High freight demand**: 0 Mezzanine decks
### Fixed vs flexible deck configuration average revenues

<table>
<thead>
<tr>
<th>Mezzanine decks</th>
<th>0 (fixed)</th>
<th>1 (fixed)</th>
<th>2 (fixed)</th>
<th>Flexible</th>
</tr>
</thead>
<tbody>
<tr>
<td>High car demand</td>
<td>68.80</td>
<td>70.95</td>
<td>71.12</td>
<td>71.61</td>
</tr>
<tr>
<td>Medium demand</td>
<td>68.36</td>
<td><strong>69.24</strong></td>
<td>66.10</td>
<td>67.73</td>
</tr>
<tr>
<td>High freight demand</td>
<td><strong>66.56</strong></td>
<td>60.77</td>
<td>46.73</td>
<td><strong>66.58</strong></td>
</tr>
<tr>
<td>Ave DP <strong>solution time</strong> (seconds)</td>
<td>16.05</td>
<td>189.87</td>
<td>129.66</td>
<td></td>
</tr>
</tbody>
</table>

- Identifies the best fixed deck configurations for each demand scenario
- Flexible pricing strategy is not always best
- The dynamic program can be solved very rapidly
- Choosing the best deck configuration improves revenues by an average of 17%
Benefit of 2-d vehicle packing (no lanes)

Comparison of average revenue results for the exact and simulation based models for various vehicle type discretization schemes

- 1-D Simheuristic: -2.52% revenue
- 2-D Simheuristic: +31.72% revenue
Future work

• Integrate dynamic pricing with CLV considerations

• Investigate the long term impact of optimal dynamic pricing policies

• Commercial partners are interested in their price acceptance model. The challenge is that their data is capacity constrained and click data is unreliable

• Investigate the impact of bottle necks in the loading procedure on packing feasibility
Conclusion

• The general framework for integrating packing and pricing was introduced and the main challenges involved were highlighted

• Two alternative approach—exact and simulation based—were presented

• Insights:
  – Revenue-to-go is not necessarily monotonically increasing in total remaining lane length
  – Careful vehicle type discretisation significantly improves revenues, small high demand vehicle types should be modelled in as much detail as possible

• Results
  – Finding the best deck configuration increases revenue by 17% on average
  – Simulation based approach achieve 97.48% of optimal revenue
  – Considering 2D packing increases revenues by an average of 31.72%
Simulation approach overview

- Derive seed loading parameters
- Observation of loaders
- Initialization

Simulated annealing

- Loading simulator
- Load optimizer

Transition functions

Dynamic program

- Pricing algorithm
- Dynamic pricing policy

Selling season simulator (testing)
Integer programming 1-D bin packing formulation (Exact)

$y_{1,1} = 1 \quad y_{2,1} = 1 \quad y_{3,1} = 0
\quad d = \{1,1,1,1,1\}$

$\mathcal{J}$: Set of lane types (bins) $j \in J$

$\mathcal{I}$: Set of vehicle types $i \in I$

$\ell_j$: Length of lane type $j$

$\ell_i$: Length of vehicle type $i$

$x_{1,1,1} = 1 \quad x_{2,1,1} = 1 \quad x_{3,2,1} = 1 \quad x_{4,2,1} = 1 \quad x_{5,2,1} = 1$
2D Packing on the main deck
### Simulation Heuristic Experiment Parameters

<table>
<thead>
<tr>
<th>Vehicle type</th>
<th>Dimensions (metres)</th>
<th>Demand scenario arrival rates</th>
<th>rate of lift</th>
<th>parking gags</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length</td>
<td>Width</td>
<td>Height</td>
<td>1. High car</td>
</tr>
<tr>
<td>Car</td>
<td>4.326</td>
<td>1.871</td>
<td>1.5</td>
<td>0.88264</td>
</tr>
<tr>
<td>Van</td>
<td>6.132</td>
<td>2.182</td>
<td>2.3</td>
<td>0.02655</td>
</tr>
<tr>
<td>Minibus</td>
<td>6</td>
<td>2.185</td>
<td>2.5</td>
<td>0.02106</td>
</tr>
<tr>
<td>Caravan</td>
<td>11.025</td>
<td>2.35</td>
<td>2.5</td>
<td>0.01958</td>
</tr>
<tr>
<td>Other towed</td>
<td>8.86</td>
<td>1.8</td>
<td>2.9</td>
<td>0.0003</td>
</tr>
<tr>
<td>Motorcycles</td>
<td>0.5</td>
<td>1.8</td>
<td>1.1</td>
<td>0.04421</td>
</tr>
<tr>
<td>Coaches</td>
<td>12.064</td>
<td>2.633</td>
<td>3</td>
<td>0.00017</td>
</tr>
<tr>
<td>Freight medium</td>
<td>8.109</td>
<td>2.252</td>
<td>3.2</td>
<td>0.00155</td>
</tr>
<tr>
<td>Freight large</td>
<td>16.093</td>
<td>2.57</td>
<td>4.6</td>
<td>0.00082</td>
</tr>
<tr>
<td>Drop trailer</td>
<td>13.75</td>
<td>2.57</td>
<td>4</td>
<td>0.00034</td>
</tr>
<tr>
<td>Unaccompanied car</td>
<td>4.326</td>
<td>1.871</td>
<td>1.5</td>
<td>0.00061</td>
</tr>
<tr>
<td>Parcel cage</td>
<td>3</td>
<td>1.5</td>
<td>1.5</td>
<td>0.00083</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>7.957</td>
<td>2.024</td>
<td>2.55</td>
<td>0.00134</td>
</tr>
</tbody>
</table>
Flexible configuration pricing policy decision frequencies

High car demand
2 mezzanine decks

High freight demand
0 mezzanine decks

Medium demand
1 mezzanine deck

1 mezzanine deck
0 mezzanine decks
2 mezzanine decks
The effect of packing consideration on pricing

• Value of the total remaining lane length is not monotonic (Graph 1)
• Careful discretisation of vehicle types is important (Table 1)
• In case study example optimal deck configurations are identified for different demand scenarios
• Dynamic deck configuration policies have their merits
• Simulation approach attains close to optimal revenue for in the 1-d bin packing model whilst remaining tractable for larger and more complex problem instances
## Comparison of methods

<table>
<thead>
<tr>
<th>Exact</th>
<th>Criteria</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic programming</td>
<td>Pricing model</td>
<td>Approximate dynamic programming</td>
</tr>
<tr>
<td>1-d bin packing (lane parking)</td>
<td>Packing model</td>
<td>2-d packing heuristic</td>
</tr>
<tr>
<td>Number of vehicles of each type</td>
<td>State definition</td>
<td>The remaining area in each distinct deck region (2 or 3 dimensions for a real world case study)</td>
</tr>
<tr>
<td>Yes</td>
<td>Optimality guaranteed</td>
<td>No (but close to)</td>
</tr>
<tr>
<td>1 day</td>
<td>Solution time</td>
<td>10 minutes</td>
</tr>
<tr>
<td>5 vehicle types</td>
<td>Max problem size</td>
<td>13 vehicle types handled easily</td>
</tr>
<tr>
<td>Lane parking with parking gaps included in allocated space also captures height restrictions</td>
<td>Real world constraints</td>
<td>Lift requirements, parking gaps, lowerable mezzanine deck height restriction, position reachability, drop trailer positions, large vehicle manoeuvrability</td>
</tr>
<tr>
<td>Packing modelled exactly in dynamic pricing and selling season</td>
<td>General</td>
<td>Approximates packing in dynamic pricing but exactly in the selling season</td>
</tr>
</tbody>
</table>