UNIVERSITY OF Southampton
School of Mathematics

Packing, Pricing and Loading Operations

Chris Bayliss (RF), Christine S.M. Currie (PI), Mee-Chi So (CI), Antonio Martinez-Sykora and Julia A. Bennell

CORMSIS Seminar 8th February

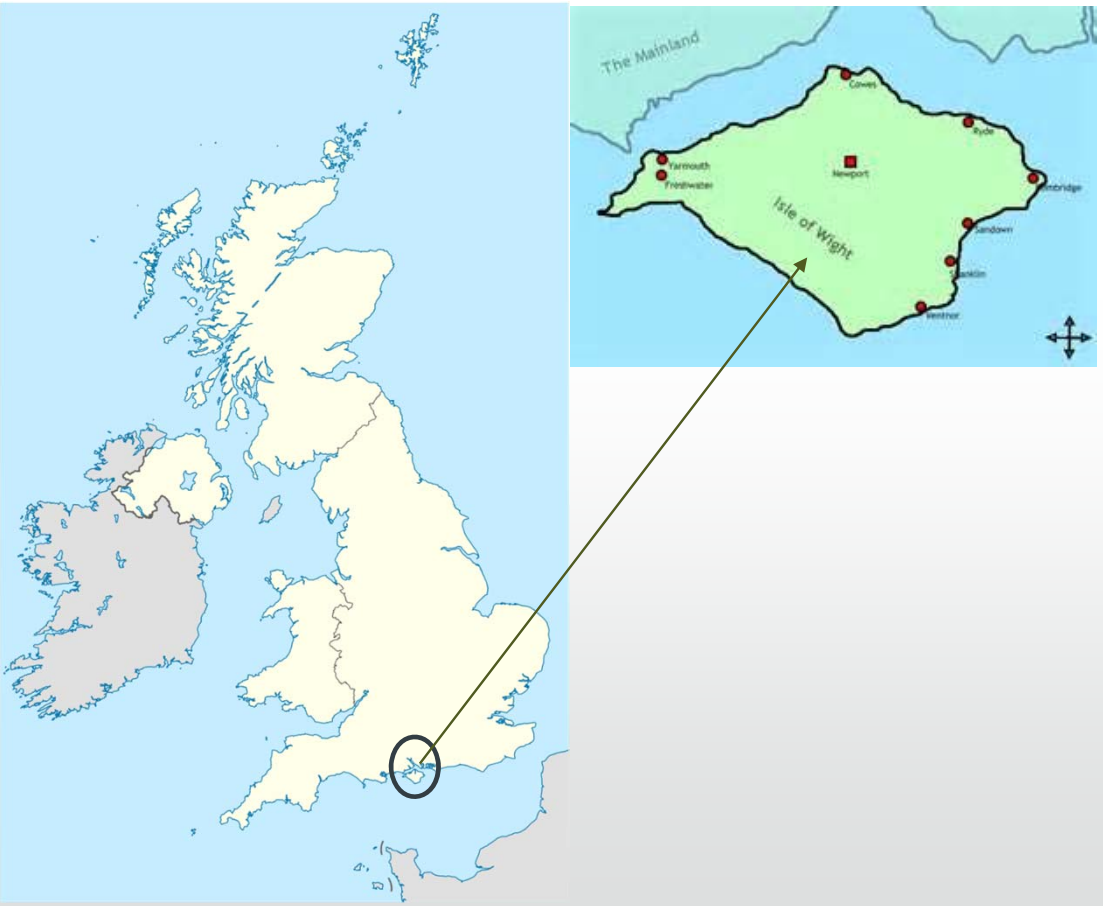
This work was funded by the EPSRC under grant number EP/N006461/1

UNIVERSITY OF Southampton
School of Mathematics

Talk contents

- Commercial partner: Red Funnel
- Blocks of work
  - Integrated dynamic pricing and packing
  - Queue constrained loading operations
  - Vehicle ferry loading simulator (demo)

Commercial partner



- Red Funnel: provide a road vehicle transport service between Southampton and the Isle of Wight
- From small vehicles such as motorbikes to large freight vehicles
- Tickets can be bought online up to 6 months prior to departure

UNIVERSITY OF Southampton
School of Mathematics

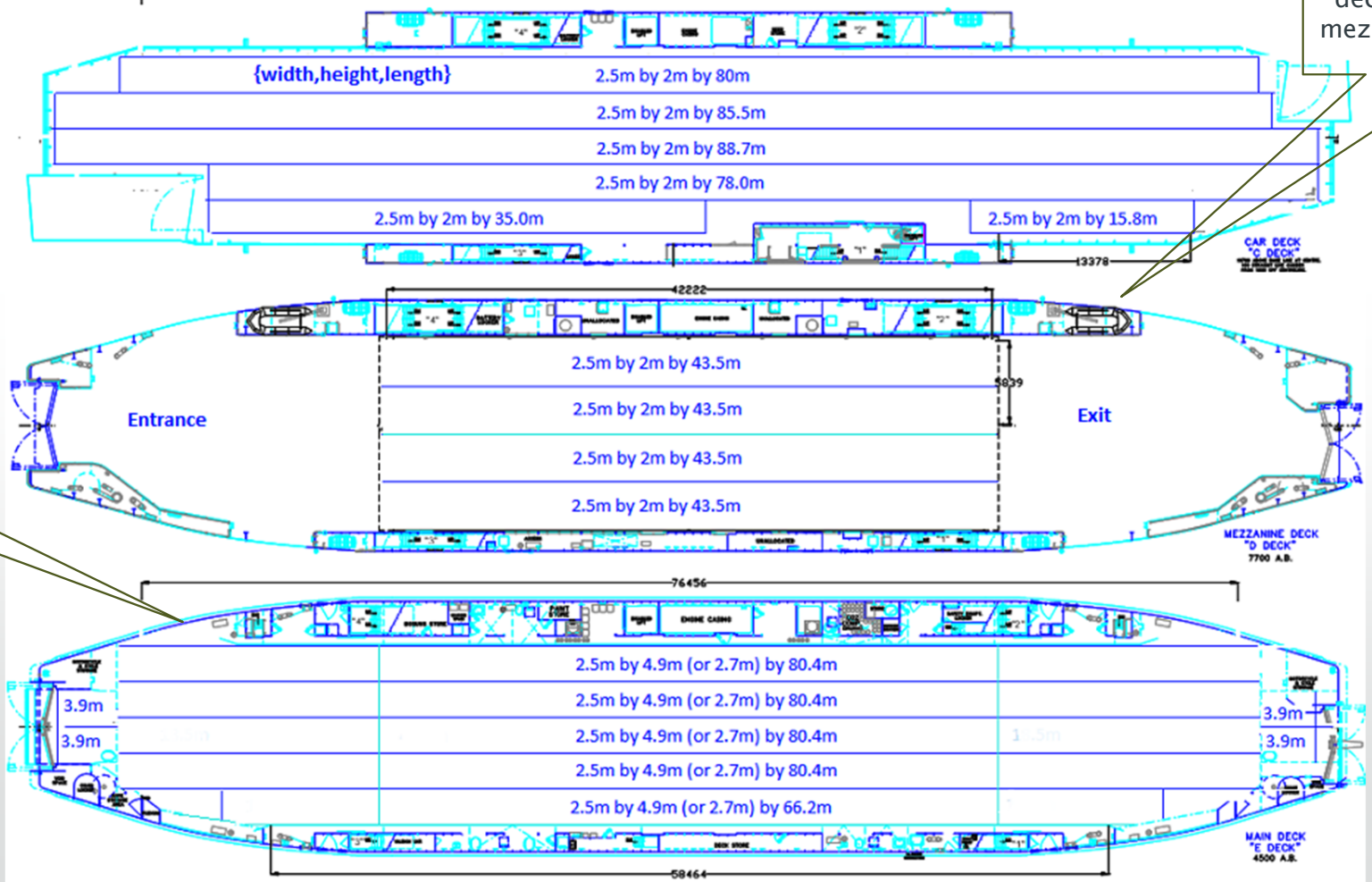
- Red Funnel's terminal in Southampton
- Ro-ro ferries (roll on roll off)

UNIVERSITY OF Southampton
School of Mathematics

Ferry dimensions

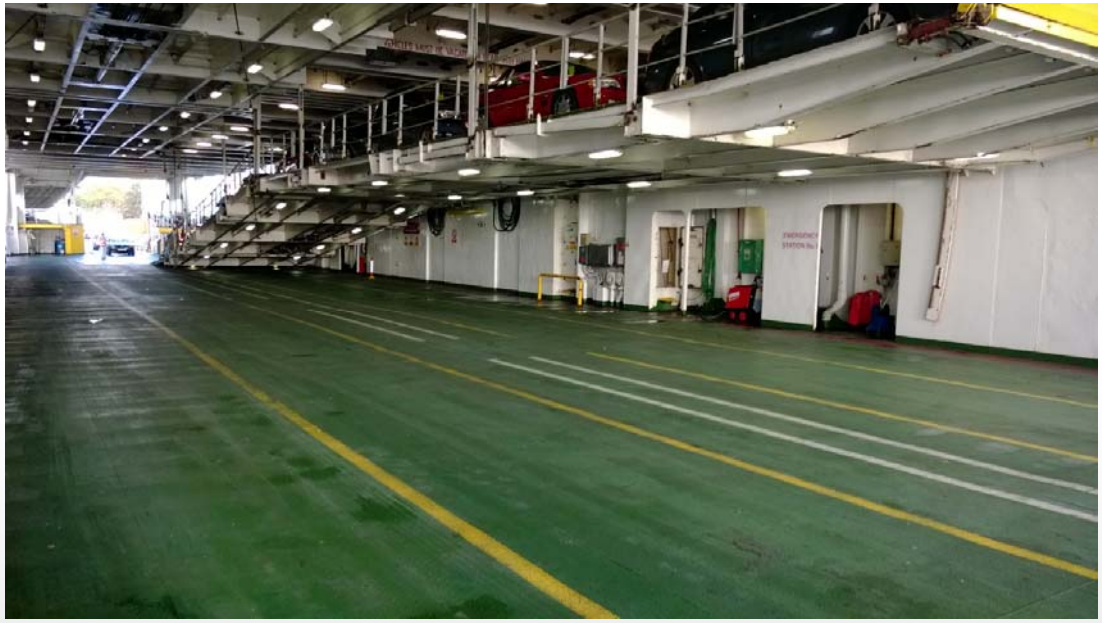
Lane parking of cars and motorcycles on the upper deck and cars only on the mezzanine decks when they are in operation

Multiple vehicle types are parked on the main deck. 2D packing Is allowed



UNIVERSITY OF Southampton
School of Mathematics

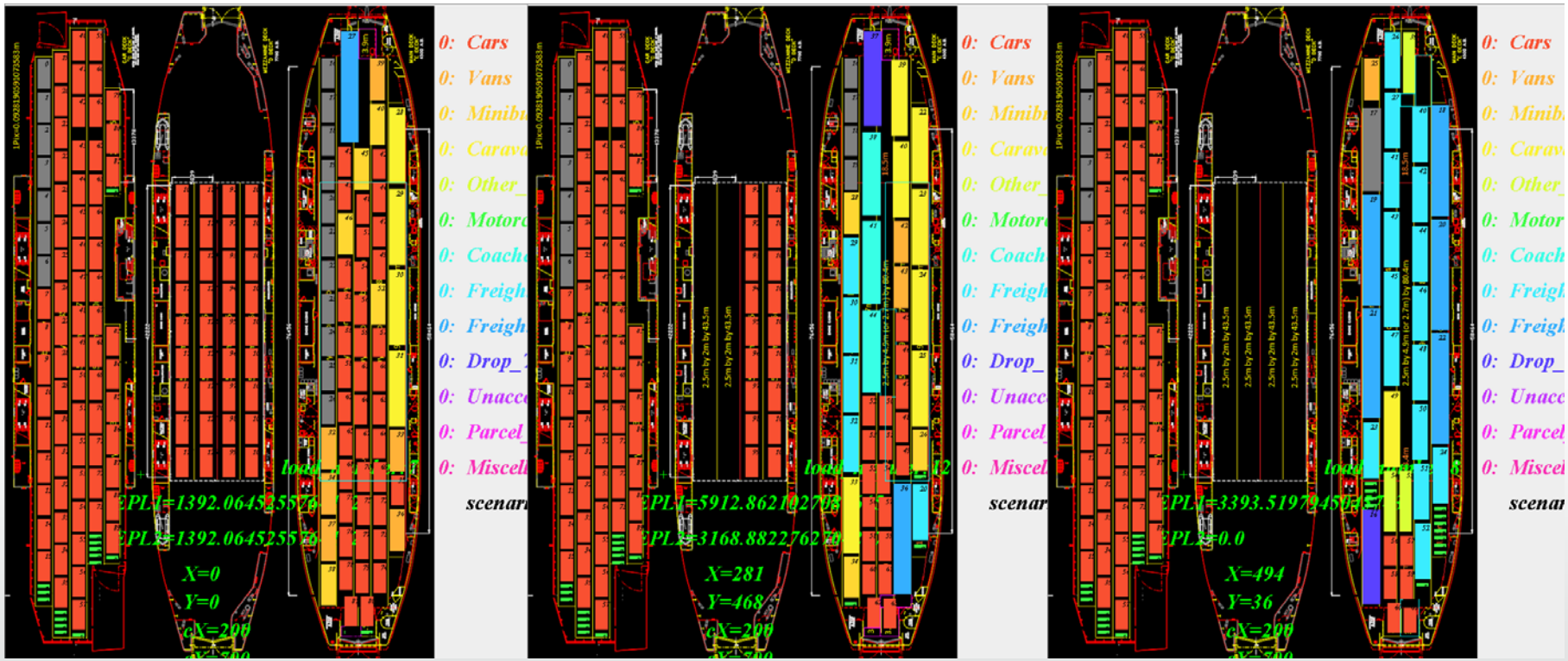
Mezzanine decks



The use of mezzanine decks increase the capacity of the ferry for low vehicles whilst reducing the capacity for high vehicles

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

INTEGRATED PRICING AND PACKING

UNIVERSITY OF Southampton
School of Mathematics

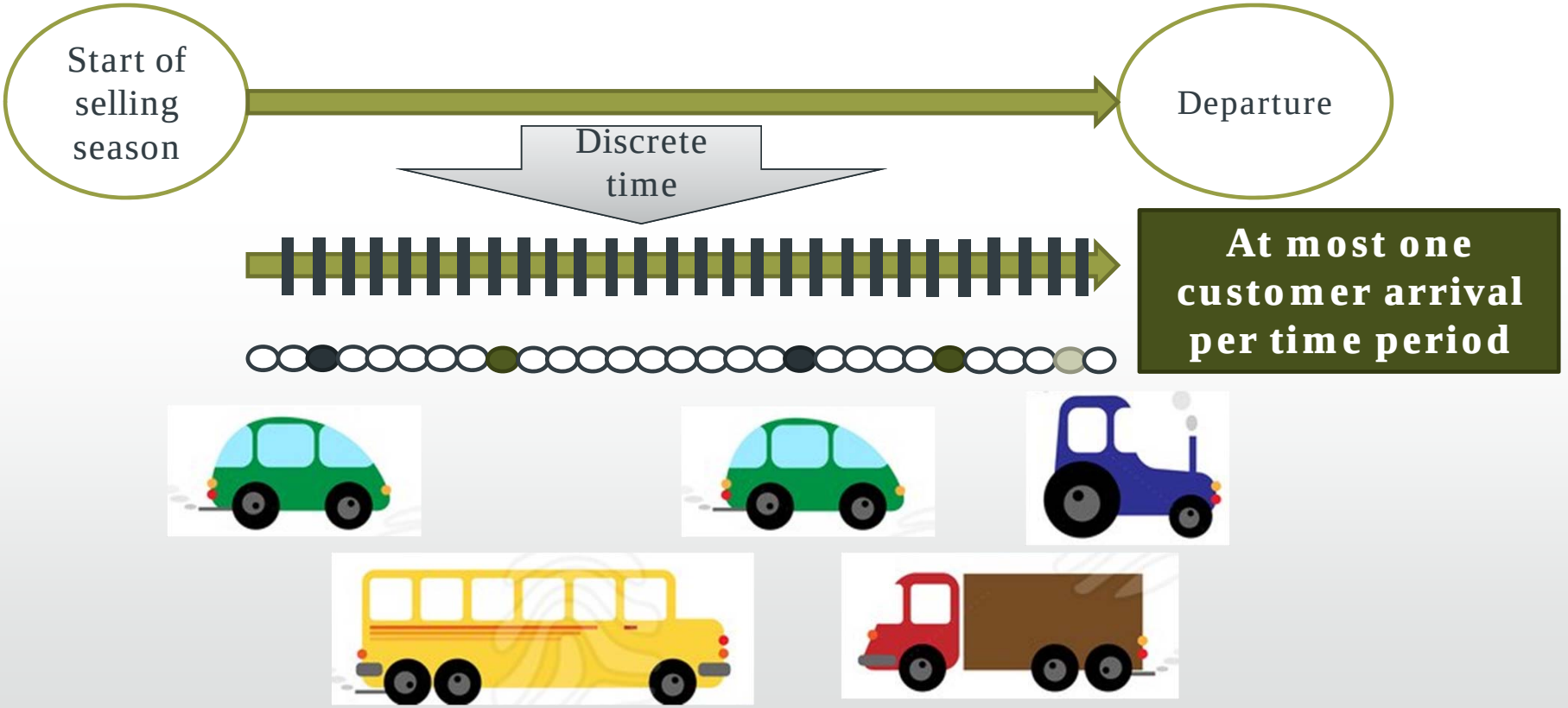
Dynamic pricing problem description

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

- **Ferry capacity depends on packing efficiency**
- Customers
  - Arrive at random times during the **selling season** which begins 6 months before departure
  - Customer **willingness-to-pay** is dependent on time until departure and varies between vehicle types
  - Vehicles vary in **size**

UNIVERSITY OF Southampton
School of Mathematics

Illustration of one selling season

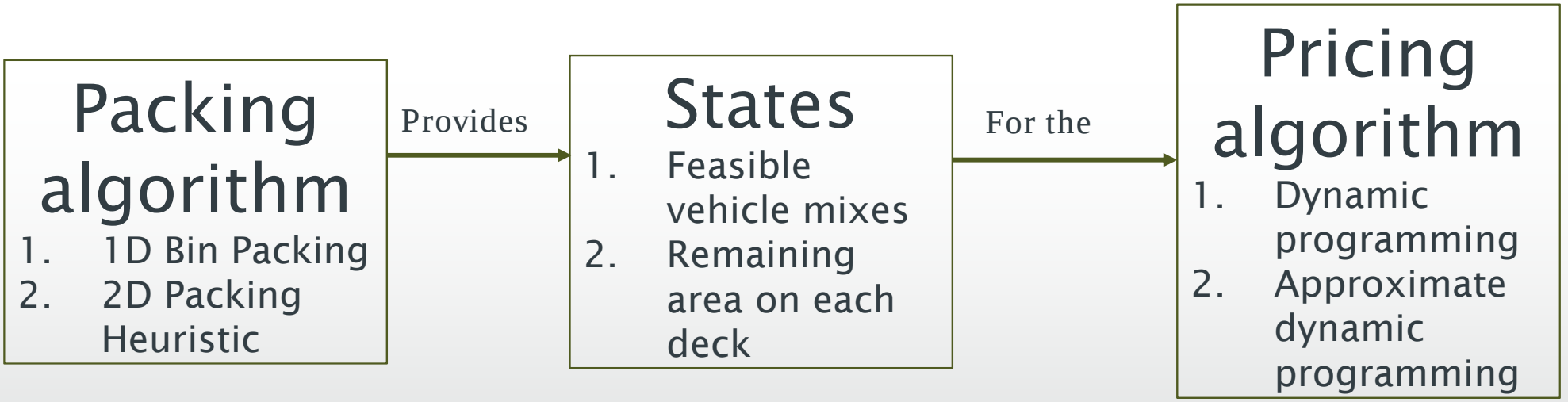


UNIVERSITY OF Southampton
School of Mathematics

Integrating packing and pricing

We have developed two main approaches

- 1. Optimal (**Exact**)
- 2. Heuristic (**Simulation based**)



Capacity model

Capture remaining capacity

Returns optimal dynamic pricing policy

UNIVERSITY OF Southampton
School of Mathematics

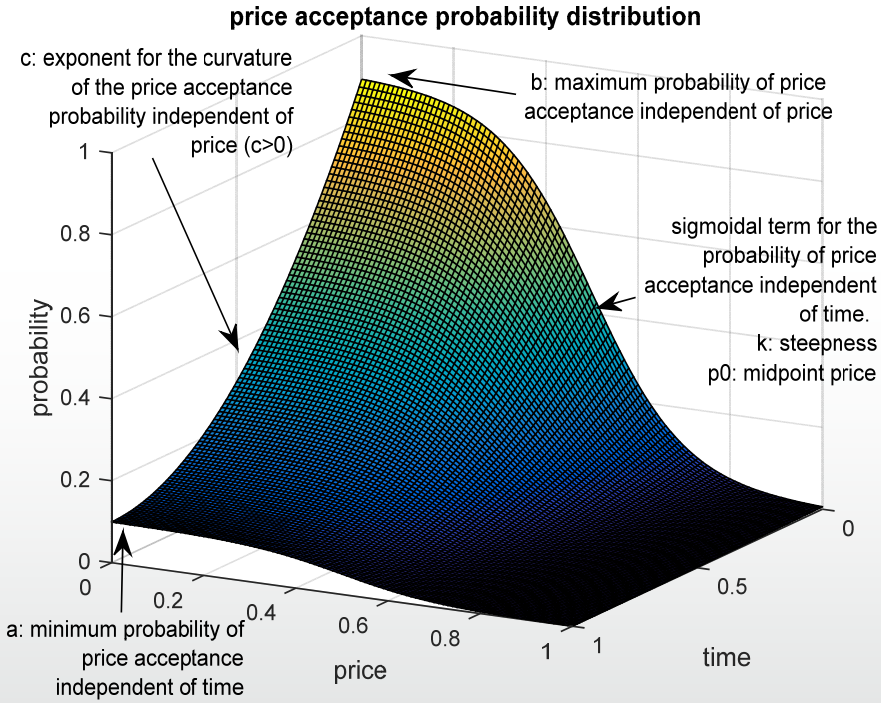
Integrating packing and pricing

- Notation
  - $\boldsymbol{s}$ denotes the state at any given time interval and captures the ferries remaining capacity for vehicles.
  - $\boldsymbol{s'}$ denotes the new state after one sale starting from state $\boldsymbol{s}$
  - $\boldsymbol{V_t(s)}$ denotes the 'revenue-to-go' or the expected future revenue if the state is $\boldsymbol{s}$ at time $\boldsymbol{t}$
  - $\boldsymbol{\lambda_{t,i}}$ denotes the probability that a customer with vehicle type $\boldsymbol{i}$ arrives at time $\boldsymbol{t}$

UNIVERSITY OF Southampton
School of Mathematics

Integrating packing and pricing

- Input functions
  - **Price acceptance function:** $\boldsymbol{\alpha(i,p,t)}$ returns the probability that a customer with vehicle type $\boldsymbol{i}$ will pay a price $\boldsymbol{p}$ at time $\boldsymbol{t}$
  - **Transition function:** $\boldsymbol{f(s,i)}$ returns the new selling season state $\boldsymbol{s'}$ if a customer with vehicle type $\boldsymbol{i}$ purchases a ticket at a time when initially in the state is $\boldsymbol{s}$



UNIVERSITY OF Southampton
School of Mathematics

Integrating packing and pricing

- Dynamic pricing formulation
  - The optimal dynamic pricing look-up-table policy can be derived by solving the Bellman equations by backwards recursion

$$V_t(s) = \max_{p \in P} \left\{ \left\{ \sum_{i \in I} \lambda_{t,i} \{ \alpha(i,p,t)(p + V_{t+1}(f(s,i))) + \big(1 - \alpha(i,p,t)\big) V_{t+1}(s) \} \right\} + \lambda_{t,0} V_{t+1}(s) \right\}$$

(3) (2) (1)

$$\forall s \in S, \forall t \in 1..T-1$$
$$V_T(s) = 0, V_t(0) = 0$$

  - 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

UNIVERSITY OF Southampton
School of Mathematics

One-dimensional bin packing (1DBP)
Two-dimension packing heuristic (2DH)

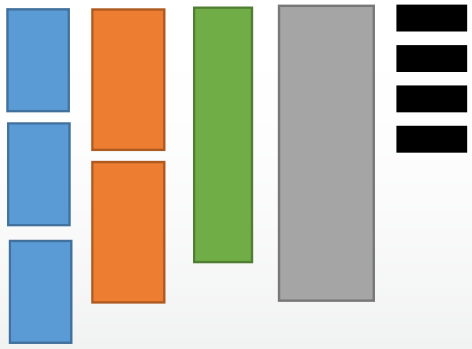
PACKING ALGORITHM DERIVED STATES

UNIVERSITY OF Southampton
School of Mathematics

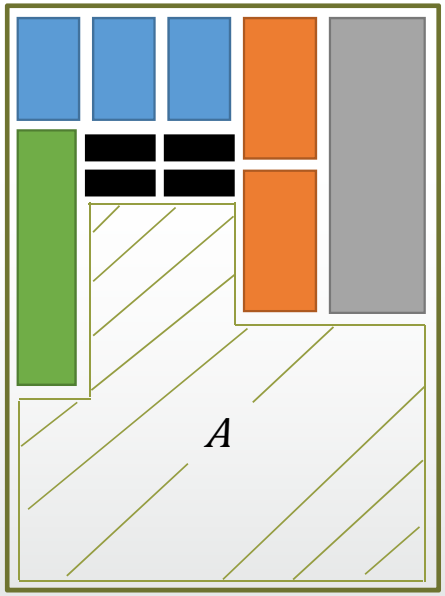
Exact and Simulation based state definitions

Exact

- State=count of vehicles of each type who have purchased tickets
- 1 state dimension per vehicle type



Simulation based



- State=remaining area on each deck
- 1 state dimension per deck region

Transition equations for one vehicle type 0 sale

$state = 3,2,1,1,4$

$f(\{3,2,1,1,4\}, 0) = \{3,2,1,1,4\} + \{1,0,0,0,0\} = \{4,2,1,1,4\}$

$state = A = 950.8, \quad transition\ value = 26.2$

$f(950.8, 0) = 950.8 - 26.2 = 924.6$

UNIVERSITY OF Southampton
School of Mathematics

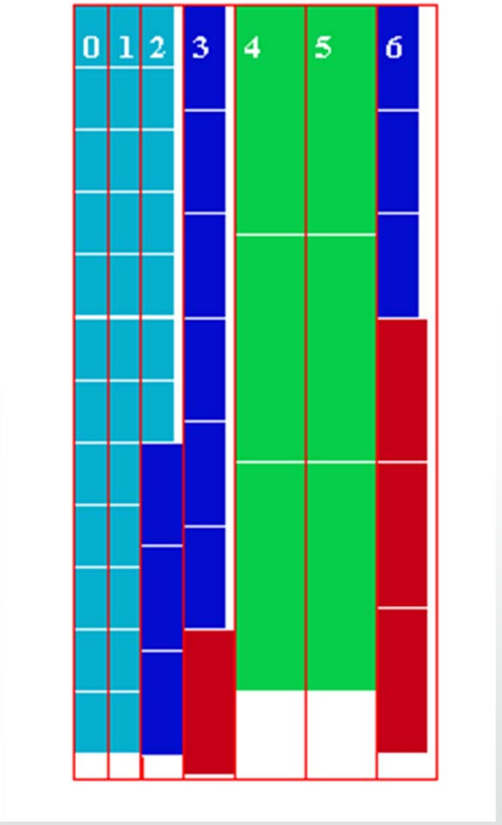
One-dimensional bin packing (1DBP)

- Each deck consists of a set of lanes (bins)
- Each bin has a width, height and length (constraints)
- Maximise the value of the packed vehicles
- $Max \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \sum_{k=1}^{|J_j|} s_{ijk} x_{ijk}$

$$\sum_{j \in J} \sum_{k \in J_j} x_{ijk} \leq d_i \quad \forall i \in I$$

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

$$x_{ijk} \in \mathbb{N} \quad \forall \{\forall i \in I, \qquad \forall j \in J, \qquad \forall k \in J_j\}$$



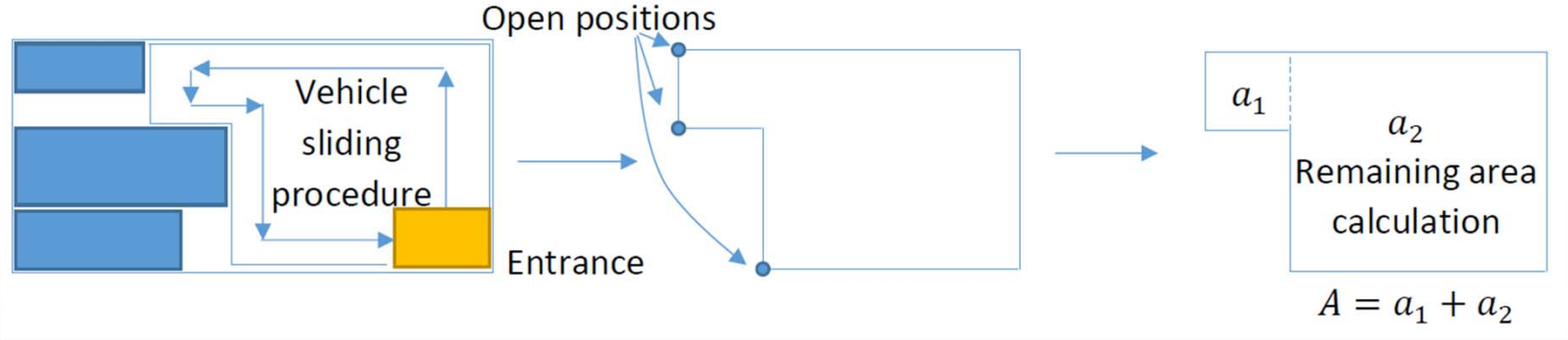
UNIVERSITY OF Southampton
School of Mathematics

One-dimensional bin packing (1DBP)

- Extended to allow:
  - for vehicles **parked across adjacent lanes**
  - for variable **lane and deck configurations**

Two-dimensional packing heuristic (2DH)

- **A vehicle sliding procedure** was used to track the remaining space state and the available vehicle positions

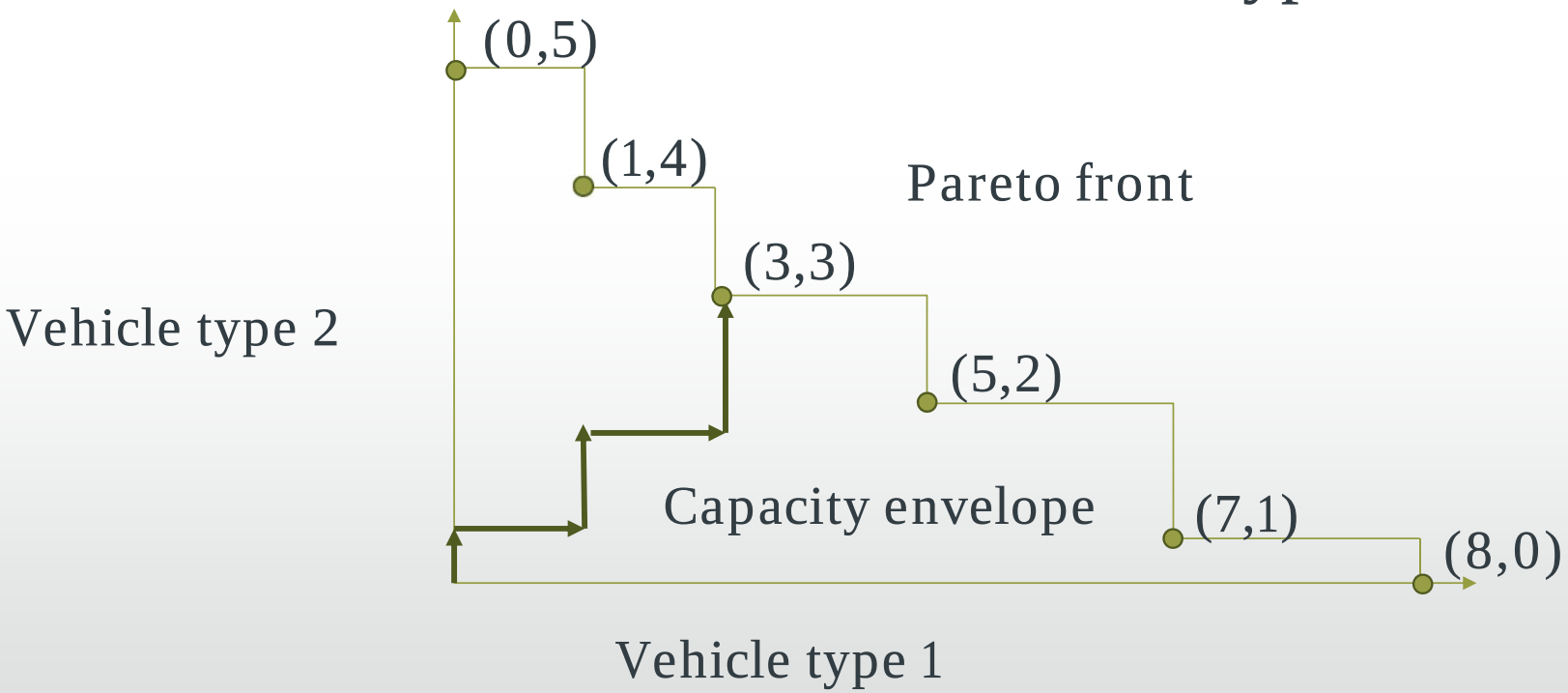


- Loading decisions (vehicle k and its position j) are chosen to maximise a weighted sum of efficiency attribute scores
  - $(j^*, k^*) = argmax\{\sum_{i=1}^{n} w_i a_{ijk}\}$
- A metaheuristic is used to tune the weights to maximise packing efficiency

UNIVERSITY OF Southampton
School of Mathematics

Vehicle mix state space

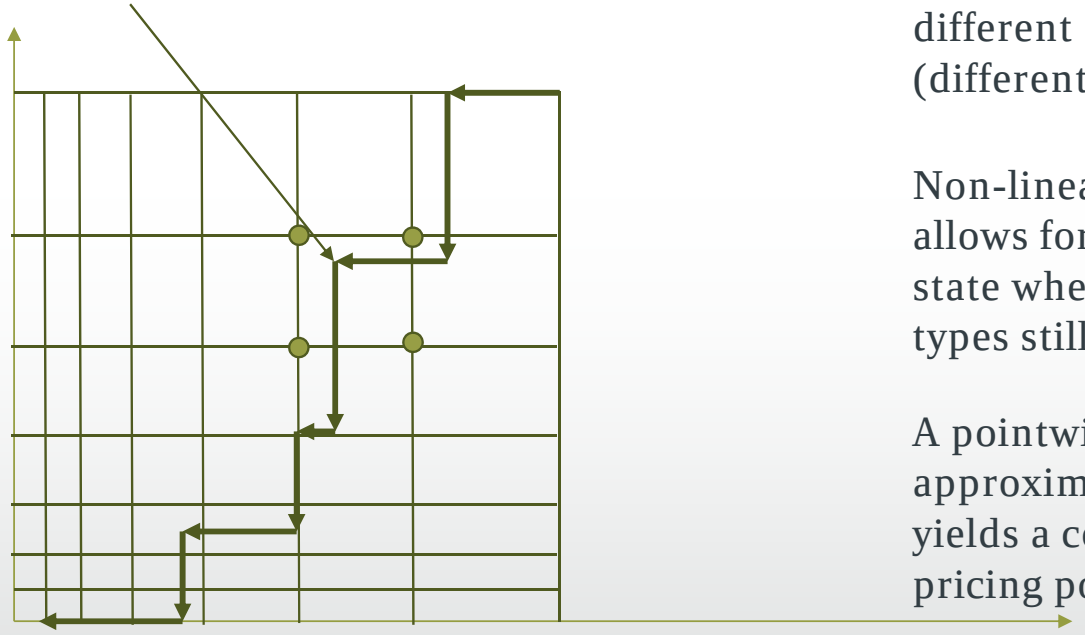
Vehicle type 2 > vehicle type 1



UNIVERSITY OF Southampton
School of Mathematics

Remaining deck area state space

The value of this state has to be interpolated from the neighbouring defined points



Main deck remaining area

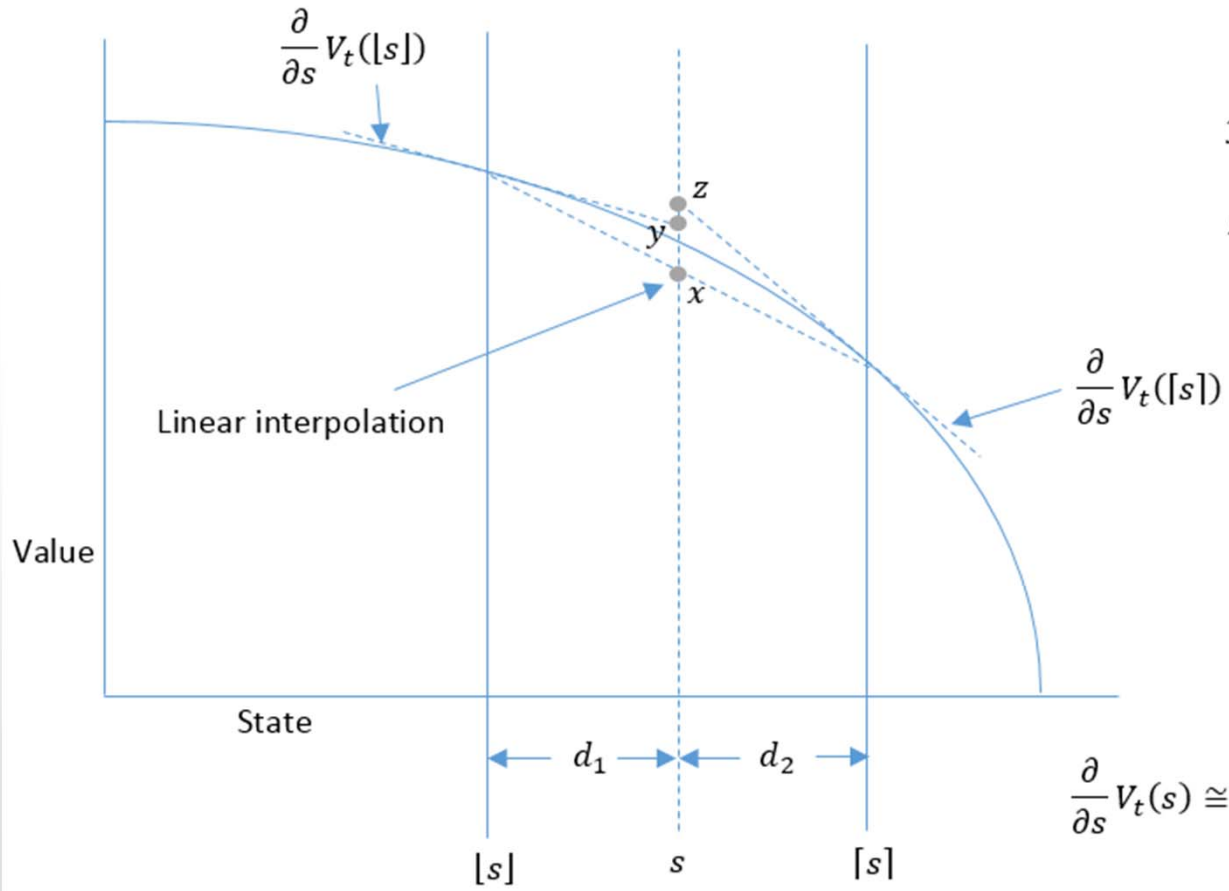
Car deck remaining area

Different vehicle types have different space requirements (different arrow length)

Non-linear discretisation choice allows for low remaining capacity state where only some vehicles types still fit onto the ferry

A pointwise value function approximation is calculated which yields a corresponding dynamic pricing policy

Intermediate state value interpolation (1-d case)



$$x = d_2 V_t(\lfloor s \rfloor) + d_1 V_t(\lceil s \rceil)$$

$$y = V_t(\lfloor s \rfloor) + d_1 \frac{\partial}{\partial s} V_t(\lfloor s \rfloor)$$

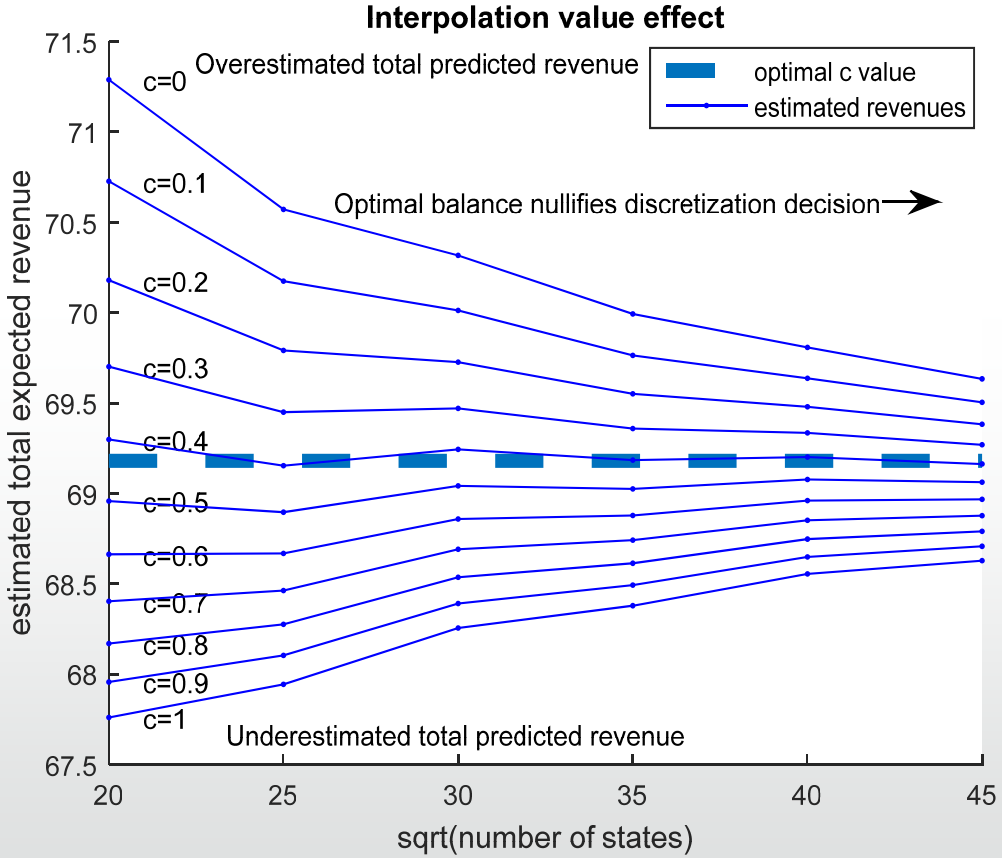
$$z = V_t(\lceil s \rceil) - d_2 \frac{\partial}{\partial s} V_t(\lceil s \rceil)$$

$$V_t(s) = cx + (1-c)(d_2 y + d_1 z)$$

$$\frac{\partial}{\partial s} V_t(s) \cong \frac{\left(V_t(s) - V_t(s-1)\right) + \left(V_t(s+1) - V_t(s)\right)}{2}$$

UNIVERSITY OF Southampton
School of Mathematics

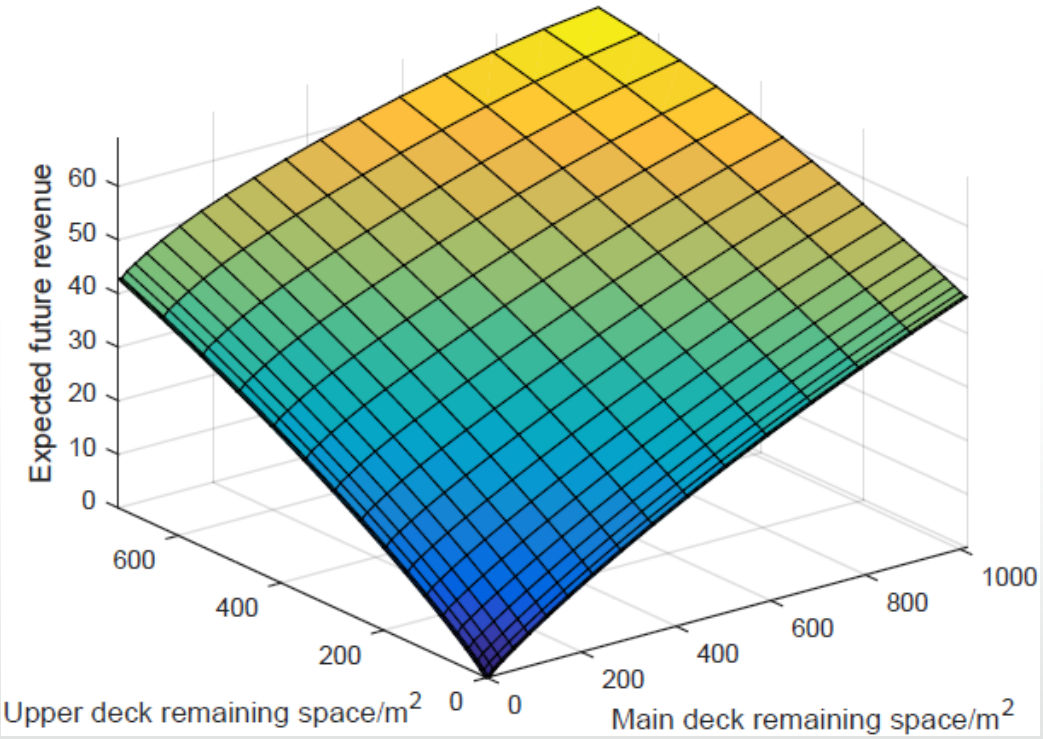
Approximating the value function (Simulation based approach)



$$V_t(s) = c(linear\ interpolation) + (1-c)(gradient\ based\ interpolation)$$

UNIVERSITY OF Southampton
School of Mathematics

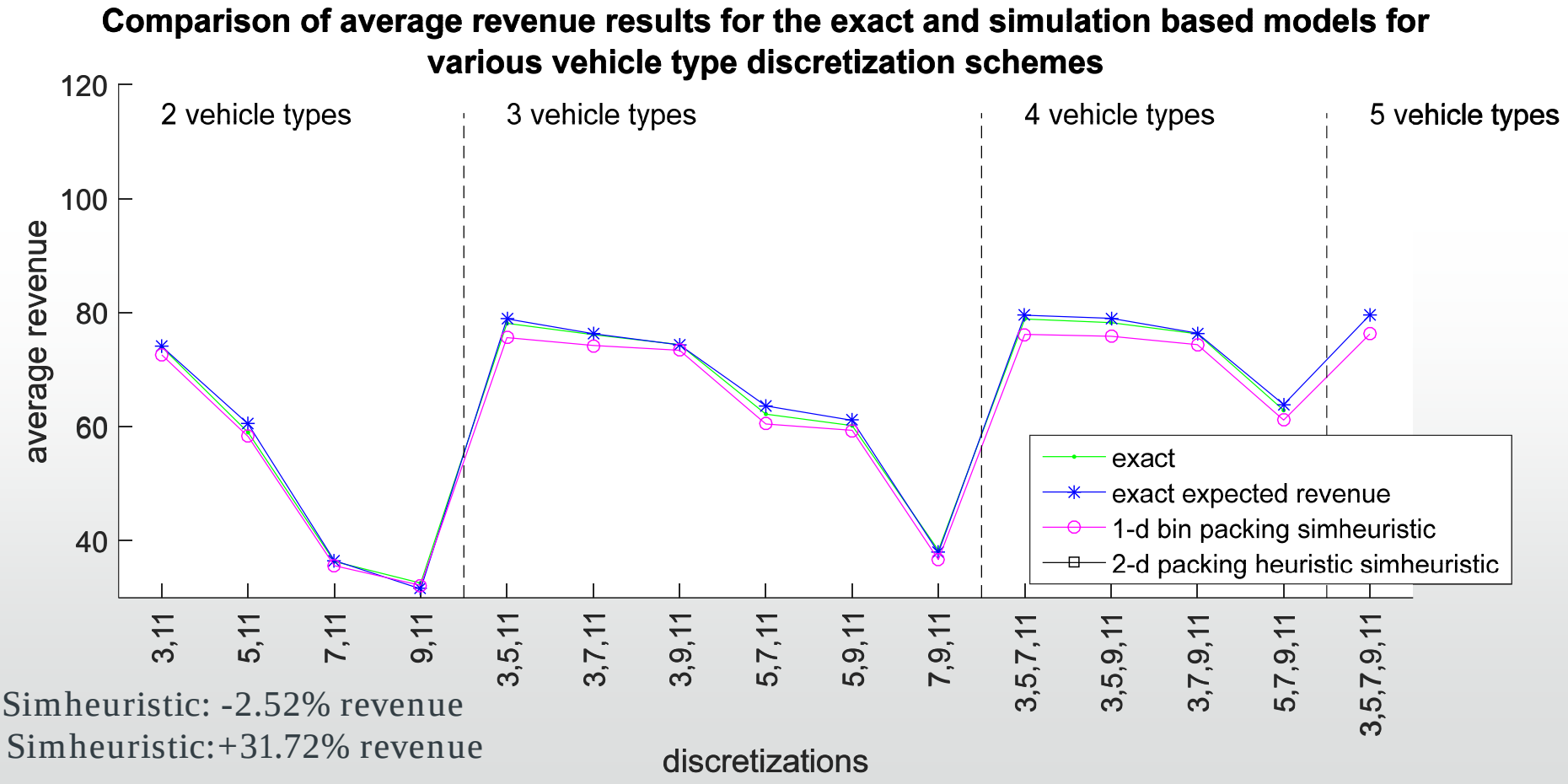
Concave structure of value function



The concave structure of the value function is exploited to speed up the solution time of the dynamic program

UNIVERSITY OF Southampton
School of Mathematics

Simulation based approach compared to Exact

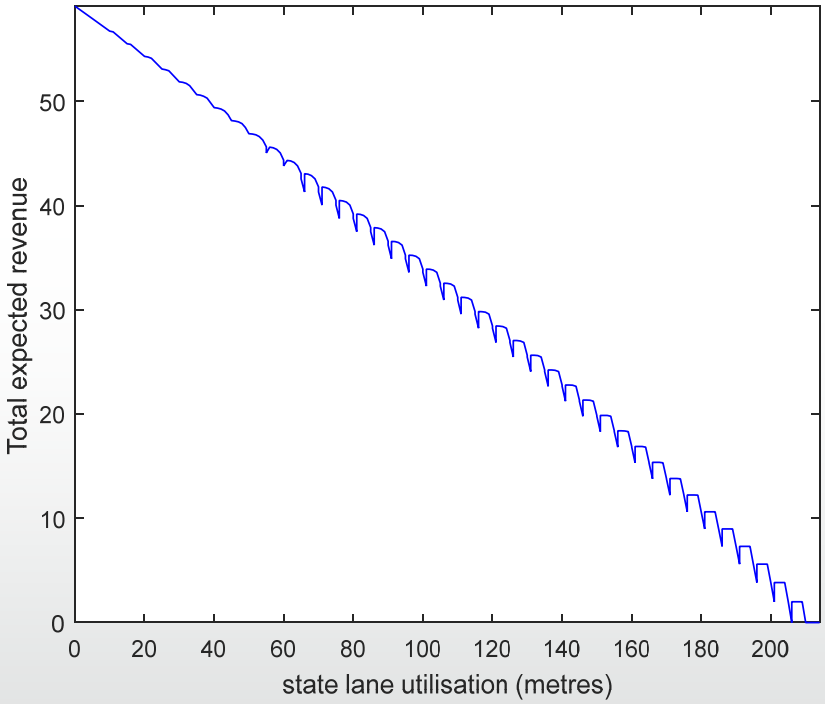


- 1-D Simheuristic: -2.52% revenue
- 2-D Simheuristic:+31.72% revenue

UNIVERSITY OF Southampton
School of Mathematics

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 because vehicle mixes with many large vehicle reduce the flexibility for packing extra vehicles thereafter



UNIVERSITY OF Southampton
School of Mathematics

THE VEHICLE FERRY LOADING PROBLEM

UNIVERSITY OF Southampton
School of Mathematics

- Parallel queues for pre-sorting vehicles

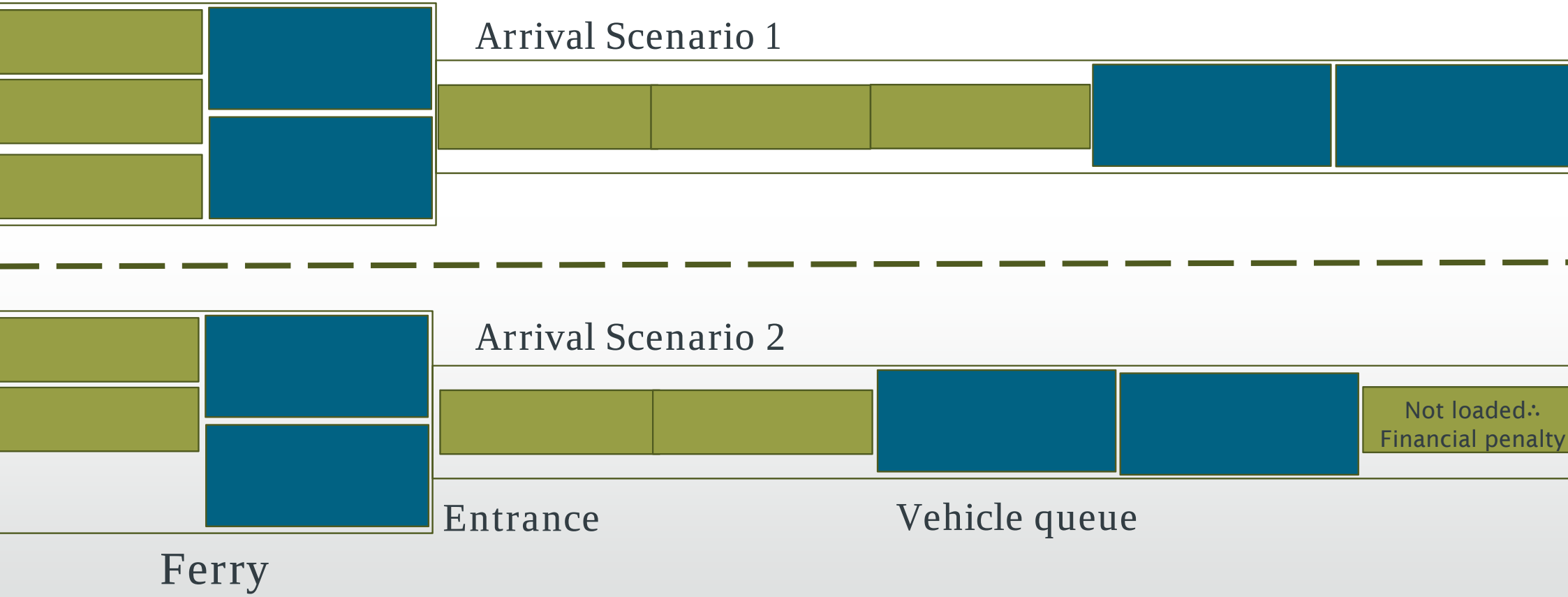
UNIVERSITY OF Southampton
School of Mathematics

The vehicle ferry loading problem

- On departure day vehicles who have purchased tickets **arrive at random times** before departure
- Upon arrival vehicles are **directed to terminal queues** according to their type and dimensions
- The **queue orders are fixed** and vehicles can only by loaded from the fronts of queues
- Vehicles that **cannot be loaded** receive a refund and compensation, known collectively as a **penalty**

UNIVERSITY OF Southampton
School of Mathematics

Queue constraints can make a packing problem infeasible



This problem can occur whenever it is not possible to keep all vehicle types in separate queues

UNIVERSITY OF Southampton
School of Mathematics

Queue constrained packing problem

Objective: maximise revenue from the sold vehicle tickets minus penalties for failing to load any vehicles

- **Inputs:**
  - Set of vehicles who have purchased tickets, arrival time uncertainty
- **Decisions:**
  - **Yard policy** for arrival vehicles
  - **Packing solution** after a realisation of arrivals

UNIVERSITY OF Southampton
School of Mathematics

GUILLOTINE CUT INSPIRED PACKING METHODOLOGY

UNIVERSITY OF Southampton
School of Mathematics

Sequential **G**uillotine-**C**ut-**K**nap**S**ack packing approach (**SGCKS**)

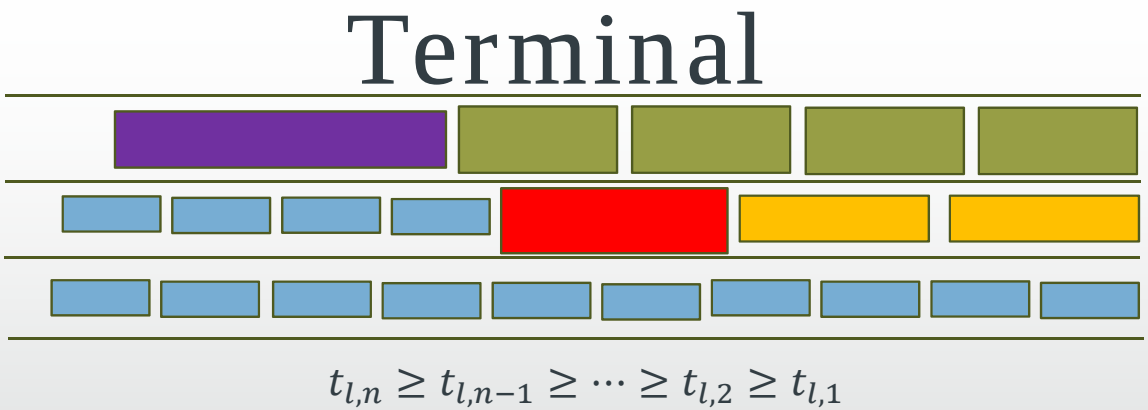
- Models **queue order** and **position reachability** constraints
  - Vehicles are loaded from the fronts of queues first
  - Parking positions have to be reachable from the ferry entrance
- In the SGCKS methodology there are two decision variables
  - Yard policy
  - Packing solution
- The yard policy and packing solutions are each encoded as **integer strings**

UNIVERSITY OF Southampton
School of Mathematics

Yard policy solution encoding

	Quantiles			
Strip type		small	middle	large
	Width	0	1	2
	Length	3	4	5

Example solution={2,5,3}



2=Vehicles with a large width

5=Vehicles with a large length

3=Vehicles with a small length

$$t_{l,n} \geq t_{l,n-1} \geq \cdots \geq t_{l,2} \geq t_{l,1}$$

The queue orders are dependent upon random arrival times and a lane allocation policy

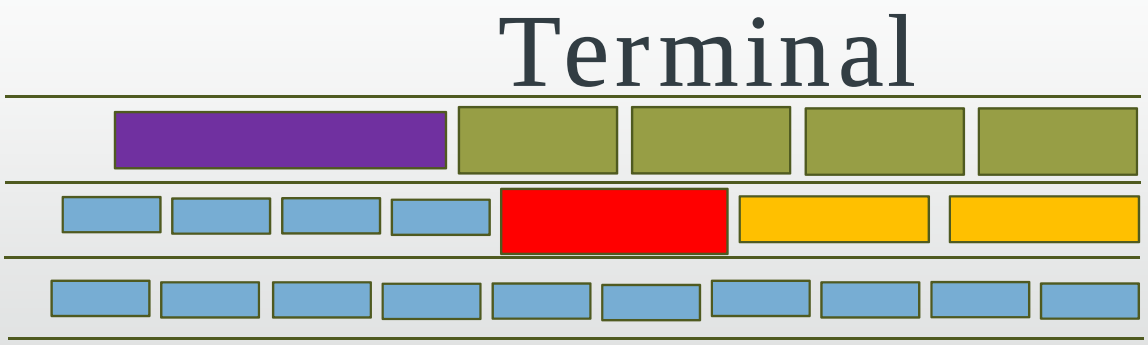
UNIVERSITY OF Southampton
School of Mathematics

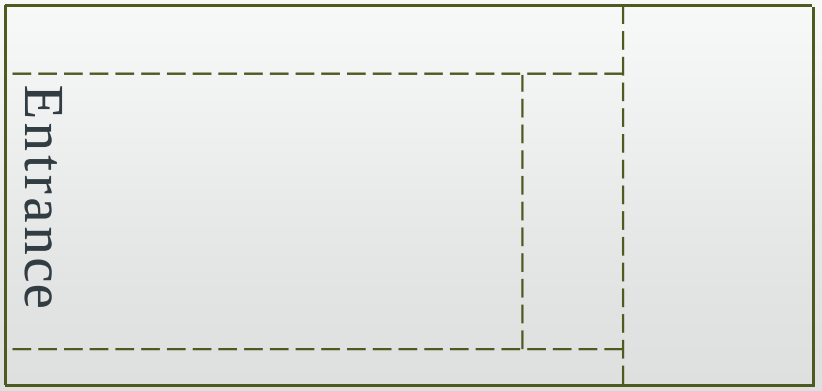
SGCKS packing solution encoding

	Quantiles			
Strip type		small	middle	large
	Bottom row	0	3	6
	Left column	1	4	7
	Right column	2	5	8

- 3 cut orientations
- 3 rectangle size quantiles

Example solution={3,2,7,0,6}

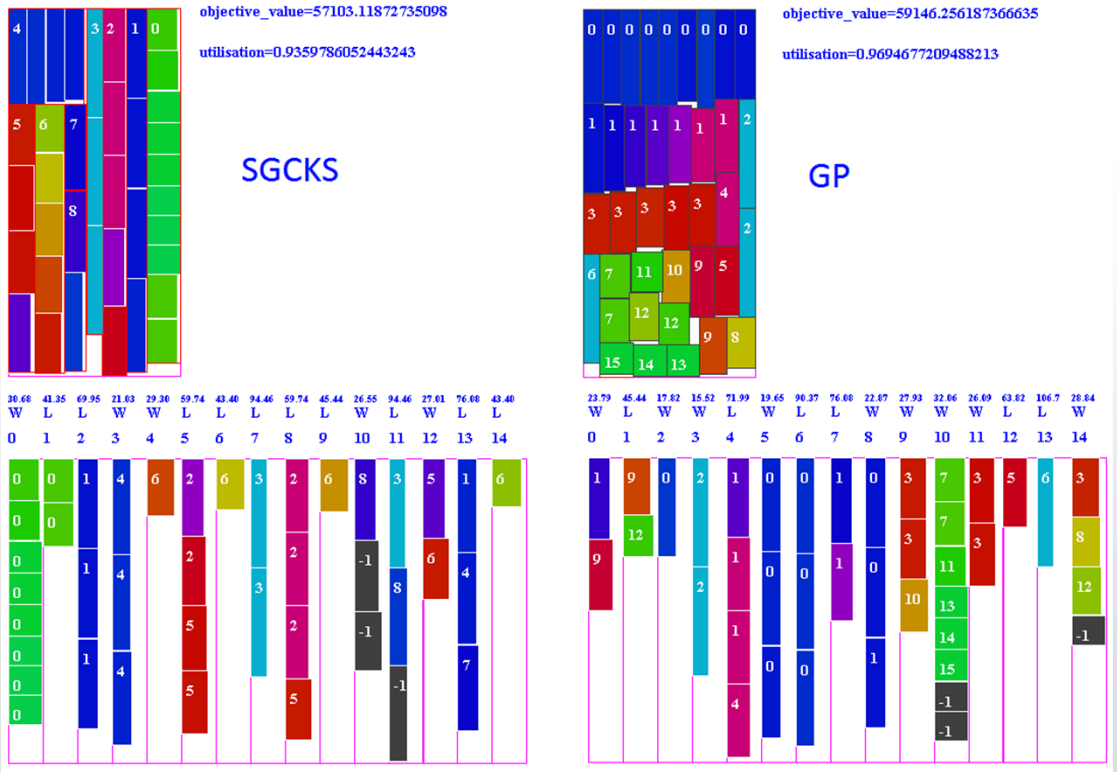




UNIVERSITY OF Southampton
School of Mathematics

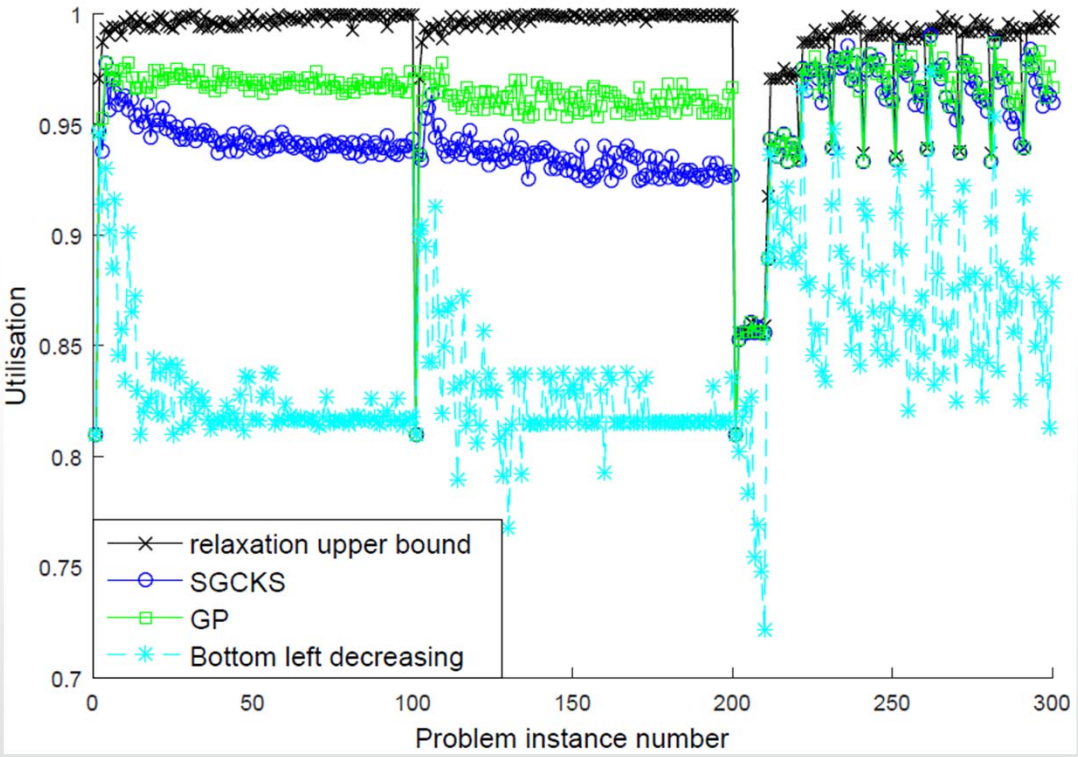
SGCKS relaxation (General Packing or GP)

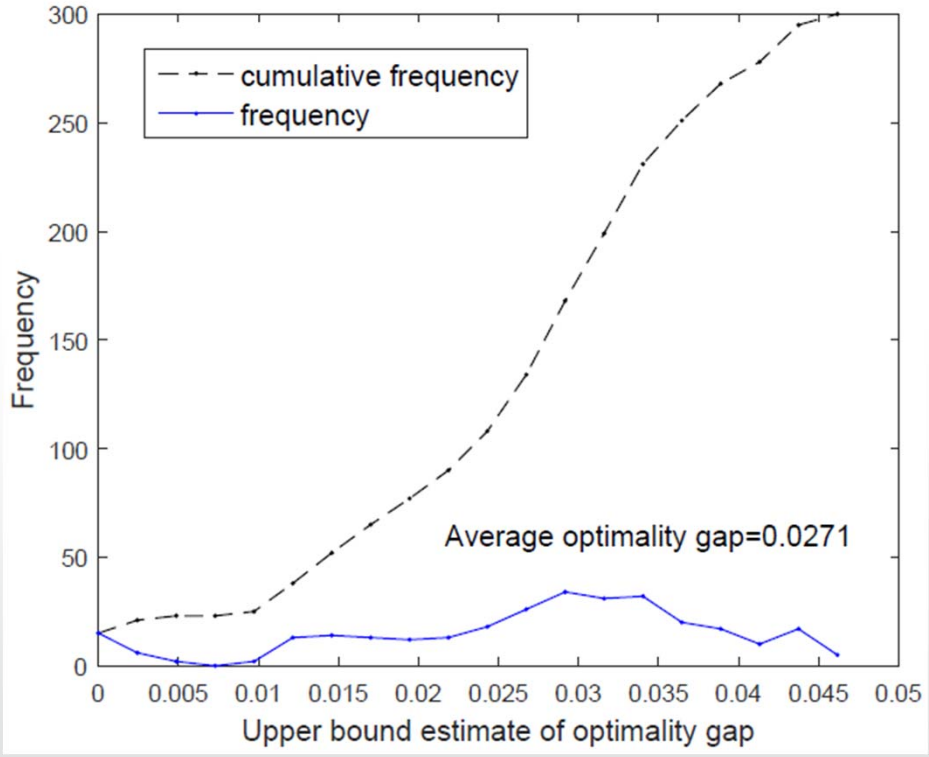
- Instead of strictly obeying the vertical and horizontal cuts use the nearest corner positions, this recovers wasted space within previous cuts



UNIVERSITY OF Southampton
School of Mathematics

Utilisations and optimality gaps for SGCKS approaches





UNIVERSITY OF Southampton
School of Mathematics

Two stage stochastic optimisation formulation

- First stage:
  - Determine a yard policy that will maximise operational revenue
  - Given uncertain arrival orders
- Second stage:
  - Solve the packing solution for a realisation of the arrival process
  - With yard queues built from the first stage yard policy solution
  - The objective is to minimise penalty payments

UNIVERSITY OF Southampton
School of Mathematics

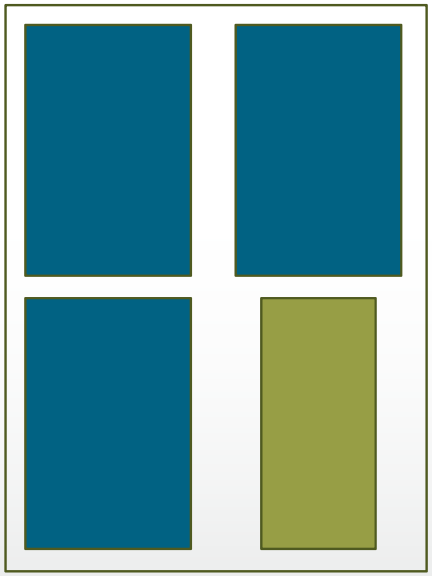
First stage solution strategy

- Use a **random set of arrival scenarios** (S) as the training sample
- Find a yard policy $\boldsymbol{y}$ and a set of packing solutions $\boldsymbol{P}$ (one for each arrival scenario) which maximises the achievable revenue
- **Iterative metaheuristic** iterates between
  - Packing optimisation for each scenario whilst fixing the yard policy
  - Optimise the yard policy whilst fixing the packing solution strings
- Two **candidate objective functions**
  - Expected revenue
  - MAXIMIN

UNIVERSITY OF Southampton
School of Mathematics

Maximise the intersection vehicle mix (MAXIMIN)

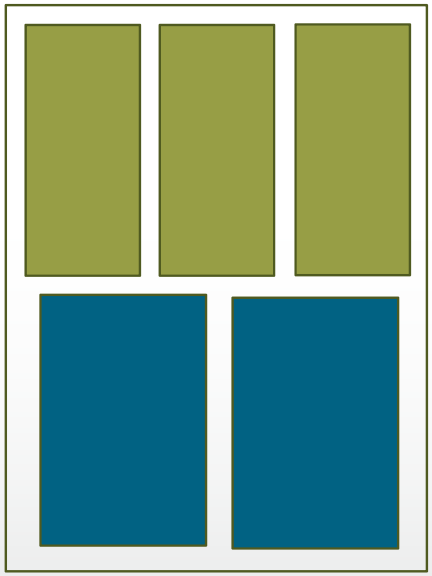
Arrival scenario 1 packing solution



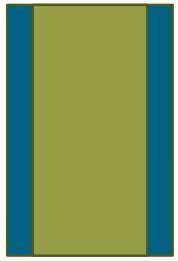
2 small vehicles left off

Tickets sold={3,3}

Arrival scenario 2 packing solution



Nested vehicle size



1 large vehicle left off

{1,2}: Simple intersection (just take the minimum of each vehicle type)
{2,2}: Intersection after accounting for nested vehicle sizes

UNIVERSITY OF Southampton
School of Mathematics

Committing to a subset of the booked vehicles

- The vehicle mix intersection approach provides a single vehicle mix that can definitely be packed in each scenario
- Therefore we could try to rebook vehicles not in the intersection
- Two policies are tested
  - **Commit to the intersection** vehicle mix and pre-emptively refund the rebooked vehicles (under the assumption that this rules out a compensation payment)
  - **Commit all vehicles**, this means do nothing

UNIVERSITY OF Southampton
School of Mathematics

First stage formulation

- $\max_{y,P} \sum_{j \in J} R_j m_j$

-Revenue value of the achievable vehicle mix

- Subject to

1. $v_s \leftarrow g\left(p_s, f(y,s)\right) \quad \forall s \in S$

-SGCKS maps y and P to vehicle mixes

2. $m_j = \min_{b \in B}\left(v_{bj}\right)$ **or** $\left(\sum_{b \in B} v_{bj}\right)/w \quad \forall j \in J$

-achievable vehicle mix (intersection or expected)

3. $B \subset S$

-over a subset of the arrival scenarios

4. $|B| = w$

-the size of the subset (risk parameter)

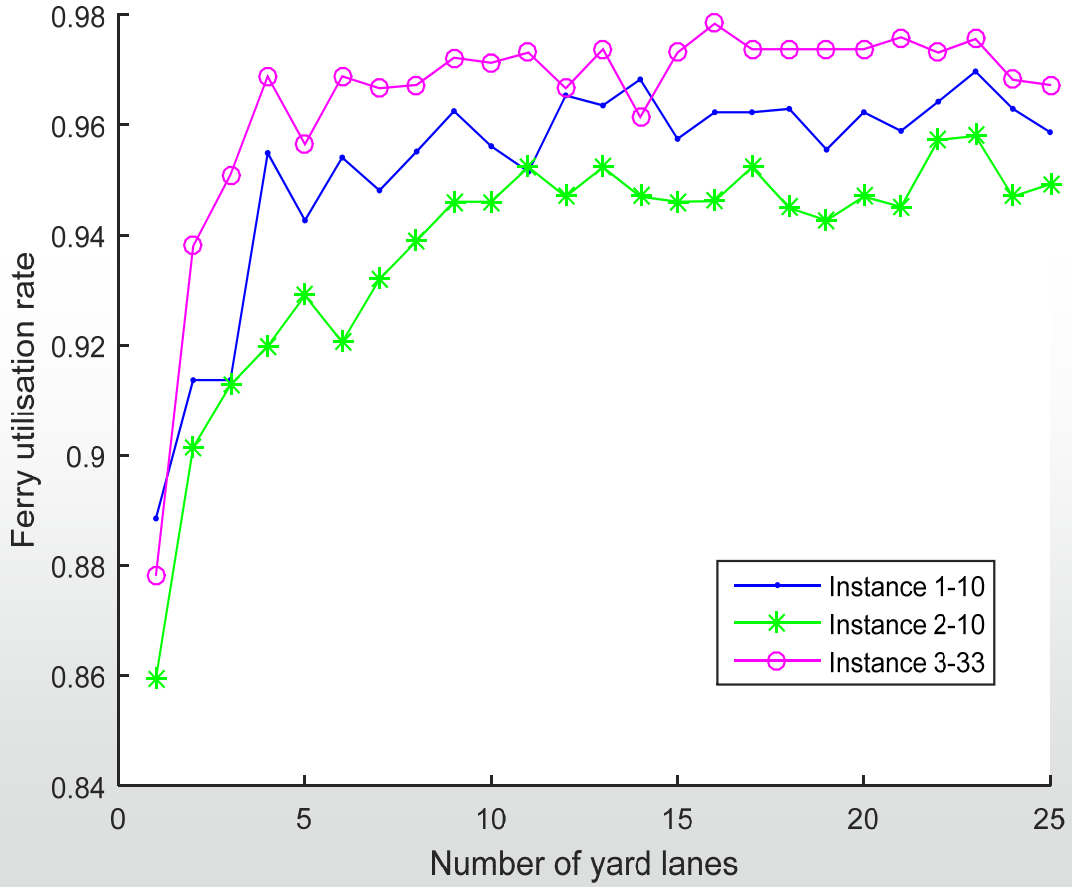
UNIVERSITY OF Southampton
School of Mathematics

Experimental results

- Graph 1: The effect of the number of terminal queues on packing efficiency
- Graph 2: Objective function comparison and committed vehicle mix policy comparison
- Graph 3: The effect of the size of number of scenarios (S) and subset size (w)

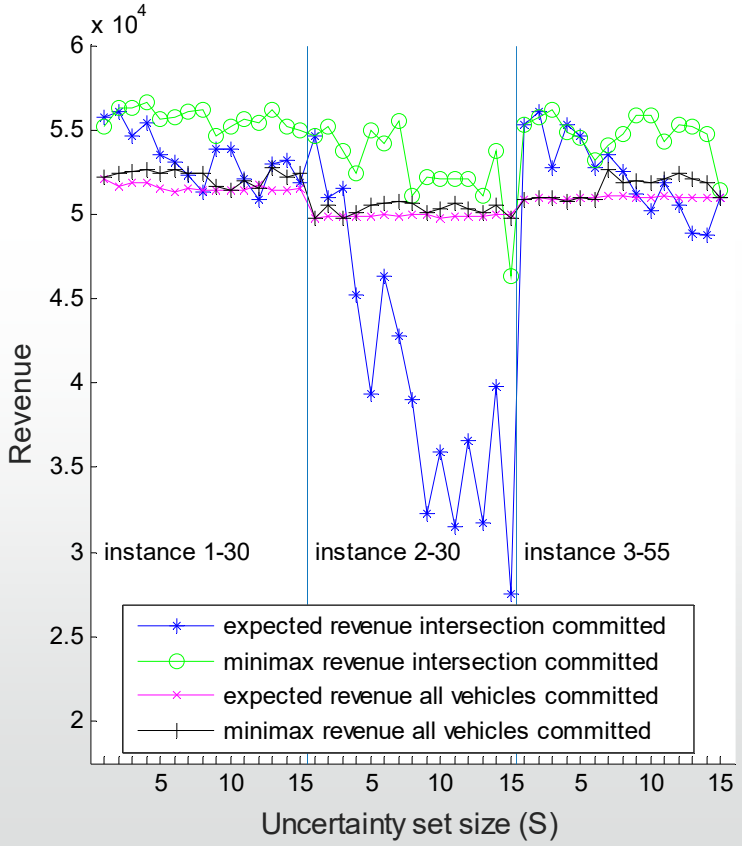
UNIVERSITY OF Southampton
School of Mathematics

The effect of the number of queues



UNIVERSITY OF Southampton
School of Mathematics

Expected revenue Vs MINIMAX



The effect of |B| and |S|

Revenue		Subset size (B)														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Uncertainty set size (S)	1	55330														
	2	56240	56778													
	3	55781	56352	56921												
	4	55214	57029	56680	55262											
	5	55494	56607	56827	56595	54862										
	6	55373	56767	56851	**57188**	56129	54721									
	7	55682	56731	57021	55847	55667	55940	54776								
	8	56225	56404	56524	56504	56548	54828	54825	54252							
	9	55516	56700	57148	56933	56277	55411	54828	53878	52969						
	10	55918	56289	56343	56980	56482	56080	54756	53807	55349	54014					
	11	55992	56779	56849	56925	56377	56206	55543	54554	54557	54414	52922				
	12	55909	55892	57075	56919	56891	54781	56172	54915	53882	54889	53272	52820			
	13	54907	56373	56769	56728	56157	55482	56015	56173	54727	54154	53622	52169	52448		
	14	56092	56052	56562	56993	56236	56396	54889	55169	53342	54600	54171	53668	54411	52105	
	15	55392	56118	56205	56980	56832	55514	56236	55900	55572	54672	53312	52191	53878	54460	53652

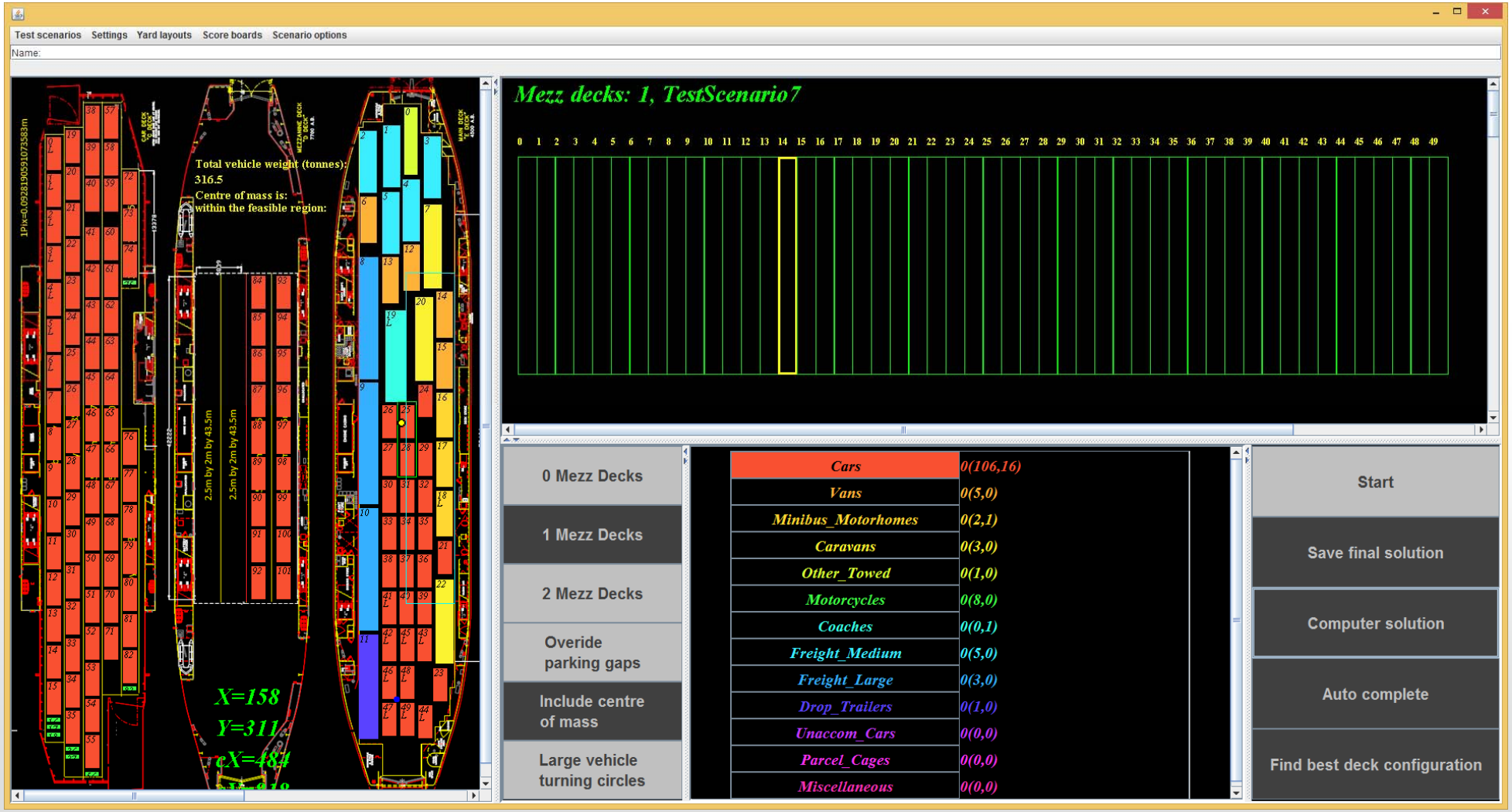
- For this fully nested 30 vehicle category example revenue was maximised by setting |S| and |B| as high as possible

- When there is no nested vehicle type structure moderate |S| and |B| values prevent us from committing to over cautious vehicle mixes

UNIVERSITY OF Southampton
School of Mathematics

LOADING SIMULATOR DEMO

UNIVERSITY OF Southampton
School of Mathematics



UNIVERSITY OF Southampton
School of Mathematics

Conclusion

- Two approaches to integrated packing and pricing for dynamic pricing of vehicle ferry tickets were developed
- Main insights:
  - Packing interactions preclude a strictly optimal formulation based on a remaining space state in favour of vehicle mix states
  - However close to optimal solutions can still be derived using a remaining space state definition

UNIVERSITY OF Southampton
School of Mathematics

Conclusion

- The loading simulator implemented as a training tool has gained the interest of Red Funnel
- In an investigation of the effect queue constrained packing—a problem that Red Funnel will face to a greater extent in the future revealed that:
  - Fewer queues does (as expected) significantly reduce packing efficiency
  - A MaxiMin objective function reduces overfitting to a small sample of training scenarios
  - The number of scenarios and subset parameter choices depend on the degree of nested vehicle size relations

UNIVERSITY OF Southampton
School of Mathematics

Thank you