UNIVERSITY OF Southampton
School of Mathematics

Genetic algorithms to solve a constrained 2-d rectangle packing problem with applications in the vehicle ferry industry

C. Bayliss, A. Martinez-Skyora, C. Currie, M. So, J.A. Bennell

ESICUP Liege May 2017

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

UNIVERSITY OF Southampton
School of Mathematics

Contents

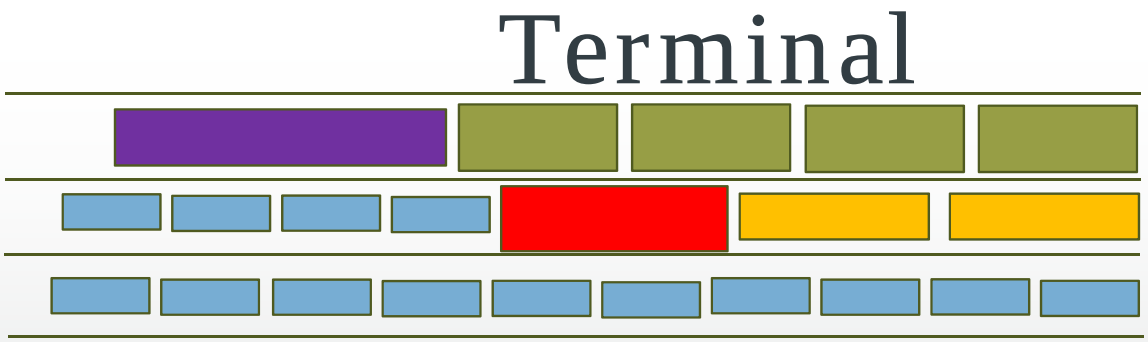
UNIVERSITY OF Southampton
School of Mathematics

Problem description

- Vehicle ferries transport private and commercial vehicles alike
- The variety of vehicle sizes makes the consideration of 2-d packing an attractive proposition
- The set of vehicles to be loaded is known but their arrival times at the terminal are not
- The vehicle ferry loading problem is constrained by:
  - Queues at the ferry terminal mean that only vehicles at the front of queues can be loaded next
  - Roll-on-roll-off ferry means orientations are fixed and parking positions must be reachable from the entrance

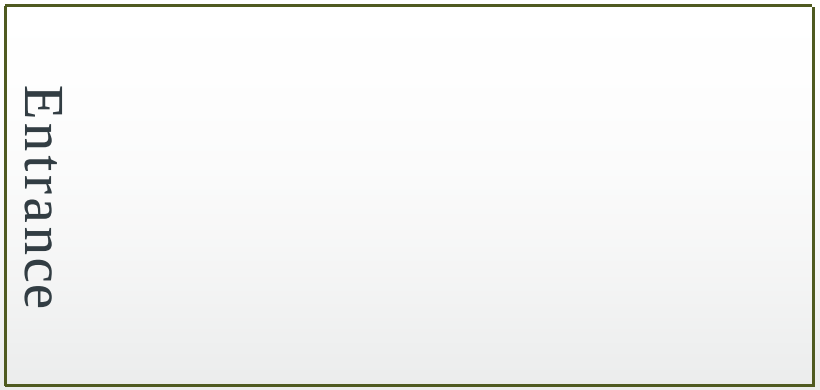
UNIVERSITY OF Southampton
School of Mathematics

Problem description



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

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



Placement order respects queue orders

UNIVERSITY OF Southampton
School of Mathematics

Problem formulation

- Objective:
  - Find a ferry terminal lane allocation policy (yard policy) such that the average space utilisation is maximised under arrival time uncertainty.
  - Alternatively, maximise (ticket revenue/claimed utilisation (first stage)-(refund+compensation for unpacked vehicles (recourse actions)) under arrival uncertainty)
- Constraints:
  - In every arrival scenario
    - Queue orders are respected
    - Loading order respects queue orders
- Recourse problem: Solve the packing problem for a given set of queues

UNIVERSITY OF Southampton
School of Mathematics

Sequential guillotine-cut-knapsack packing approach

- Packing solutions are encoded as genes which consist of strings of integers
- Each integer defines a guillotine cut orientation and the target dimensions of vehicles loaded from the fronts of queues (a look-up table of orientations and quantiles for target dimensions is used)
- Vertical cuts from the left or right of the remaining space and horizontal cuts of the bottom edge of the remaining space ensure that the entrance is always connected to the remaining space

Problem aspect	Solution approach aspect
Implementability	Packing solutions consist of a sequence of rows and columns of specified vehicles to load onto the ferry
Online/sequential implementability	The entrance edge is never cut off with a guillotine cut
2-d packing solutions	Optimising the sequence of rows and columns provides the benefits of a 2-d packing solution (current practice is based on columns only)

UNIVERSITY OF Southampton
School of Mathematics

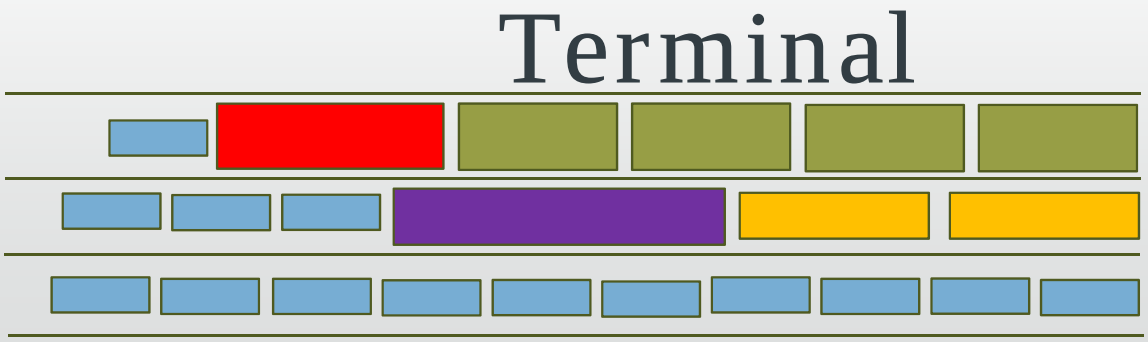
Yard policy solution encoding

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

Example solution={2,5,3}

Yard policy

- Each lane is devoted to vehicles of a target width or a target length
- On arrival vehicles are allocated to the closest matching lane
- Ties can be broken based on:
  - queue length
  - number of different vehicle types in queues



2=Vehicles with a large width

5=Vehicles with a large length

3=Vehicles with a small length

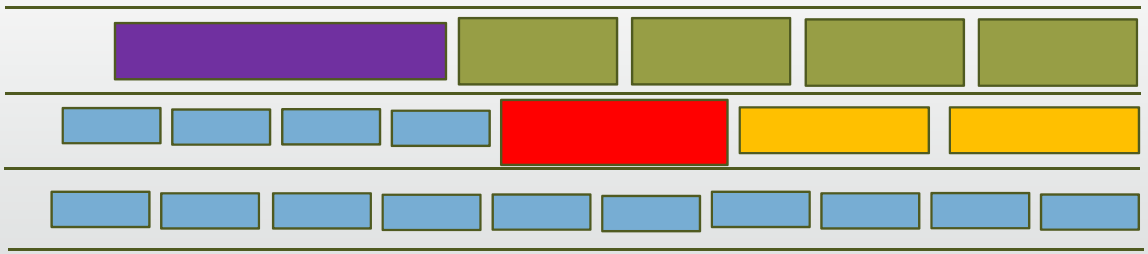
University of Southampton
School of Mathematics

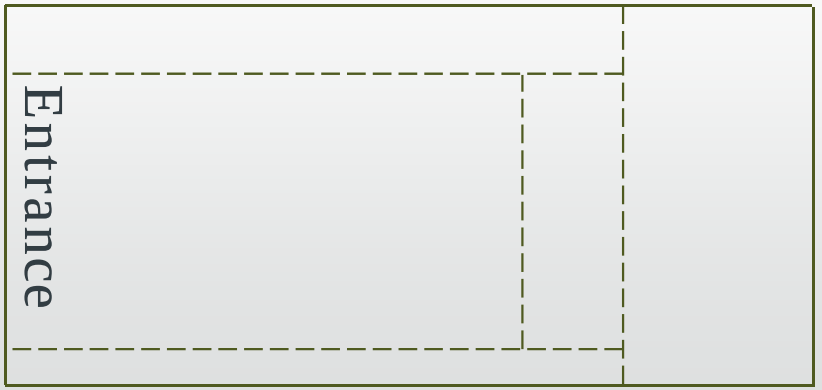
Packing solution encoding

	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

Iterative solution approach

- Since the yard policy and packing solutions are each encoded as strings the same algorithms can be used to solve each problem
- We propose an iterative approach of alternating between packing and yard policy optimisation
- Pseudocode on following slide

UNIVERSITY OF Southampton
School of Mathematics

Iterative genetic algorithms for lane policy and packing optimisation

Algorithm 1 Iterative genetic algorithm for lane policy and packing optimisation under arrival time uncertainty

Generate m random arrival scenarios
Generate and evaluate a **dual population** of solutions (P)
($P_{0,i}$ denotes the packing solution for population member i)
($P_{1,i}$ denotes the yard policy solution for population member i)
Store the best solution ($best\_sol$)
Initialise iteration type $k = 1$
$iteration = 0$
while $iteration < iteration\_limit$ **do**
  $k = 1 - k$ (**alternative iteration** type. Lane policy/packing)
  Set each element of P equal to $best\_sol$
  Randomise a portion q_k of the k type population
  $generation = 0$
  while $generation < generation\_limit_k$ **do**
    Evaluate population in terms of the average utilisation over m arrival scenarios
    update $best\_sol$
    Tournament selection and crossover on P_k
    Mutation applied to P_k (k type strings) with rate r_k
    $generation = generation + 1$
  end while
  $iteration = iteration + 1$
end while

UNIVERSITY OF Southampton
School of Mathematics

Simulated annealing algorithm

- The iterative procedure can be used to find and optimised solution for a single arrival scenario
- The lane policy and packing solutions enable escape from local optima
- We designed problem specific neighbourhoods for this problem
- An alternative approach to the general problem is to solve the packing problem to optimality then reverse engineer the lane policy that maximises the feasibility of this packing solution

Can the vehicles in the yard be assigned to the rows/columns in the packing solution whilst respecting queue order?

University of Southampton
School of Mathematics

Packing solution feasibility under arrival time uncertainty

Inputs

$n_{i,j}$ = the number of vehicle type j in strip i

$p_{l,k,j} = \begin{cases} 1 \text{ if vehicle type } j \text{ occupies position } k \text{ in lane } l \\ 0 \text{ otherwise} \end{cases}$

$|N|$ = number of yard lanes

N_l = number of vehicles in yard lane l

V = set of vehicle types

S = number of strips in packing solution

Decision variables

$\delta_{l,k,i} = \begin{cases} 1 \text{ if the } k^{th} \text{ vehicle in lane } l \text{ is assigned to strip } i \\ 0 \text{ otherwise} \end{cases}$

$b_{l,k}$ = the strip number assigned to the vehicle in the kth position in lane l

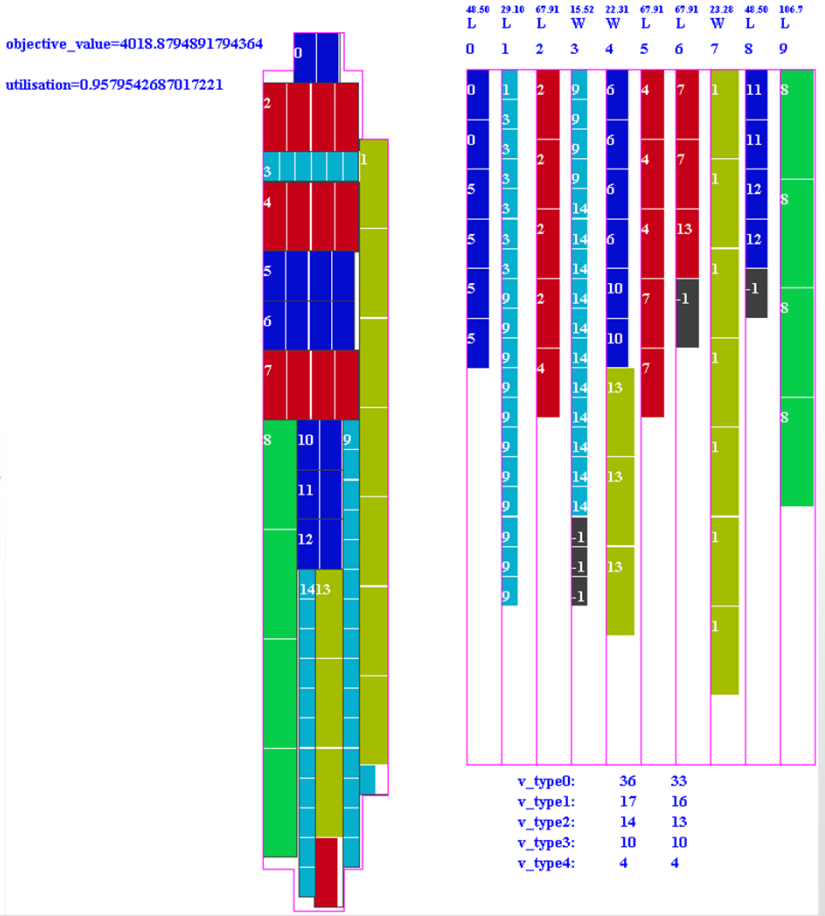
$$max: \sum_{l=1}^{|L|} \sum_{k=1}^{L_l} b_{l,k} \qquad (1)$$

s.t.

$$b_{l,k+1} \geq b_{l,k} \quad \forall l \in \{1..|N|\}, \forall k \in \{1..N_l - 1\} \qquad (2)$$

$$\sum_{l=1}^{|N|} \sum_{k=1}^{N_l} p_{l,k,j}\, \delta_{l,k,i} = n_{i,j} \quad \forall i \in S, \ \forall j \in V \qquad (3)$$

$$\delta_{l,k,i} \cdot i \geq b_{l,k} \quad \forall i \in S, \forall l \in \{1..|N|\}, \forall k \in N_l \qquad (4)$$



UNIVERSITY OF Southampton
School of Mathematics

Lower bounds for utilisation

UNIVERSITY OF Southampton
School of Mathematics

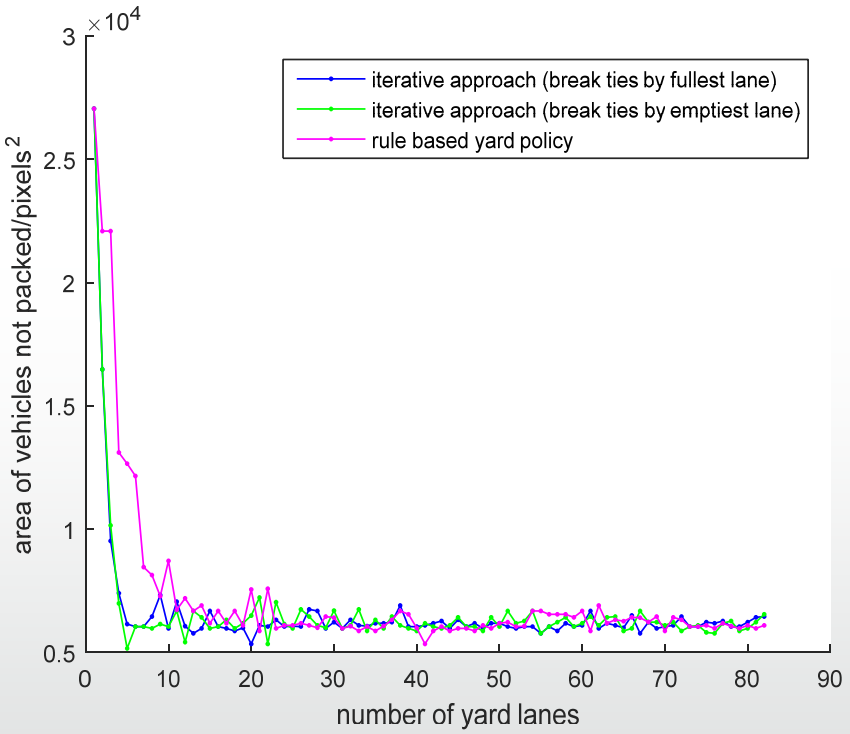
Experiments

- We compare the proposed algorithm with:
  - Bottom left
  - A simulated annealing algorithm

UNIVERSITY OF Southampton
School of Mathematics

The effect of the number of lanes

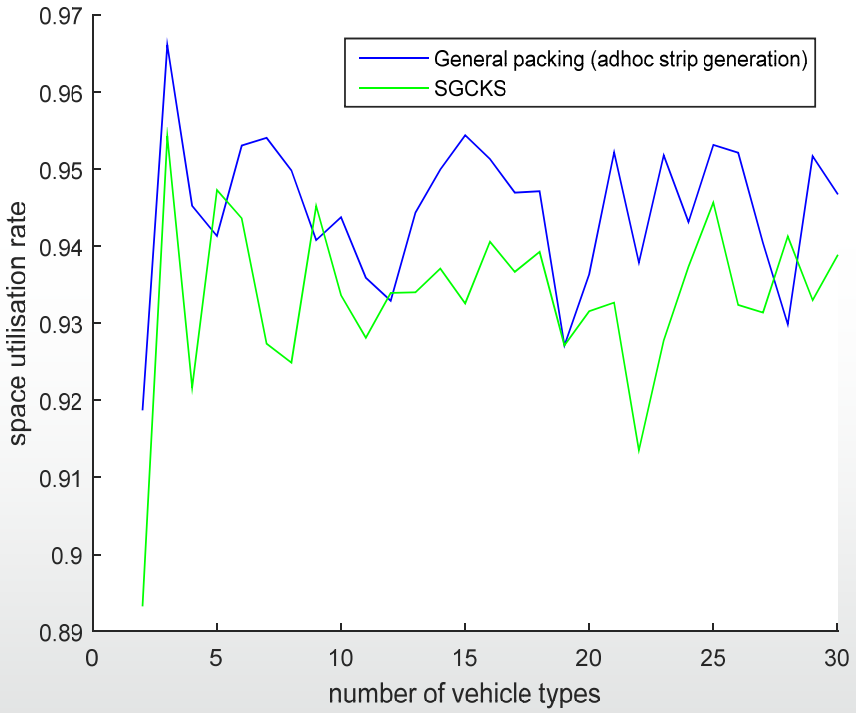
- Repeat for numbers of vehicle types other than 5
- Plus general packing



UNIVERSITY OF Southampton
School of Mathematics

The effect of the number of vehicle types

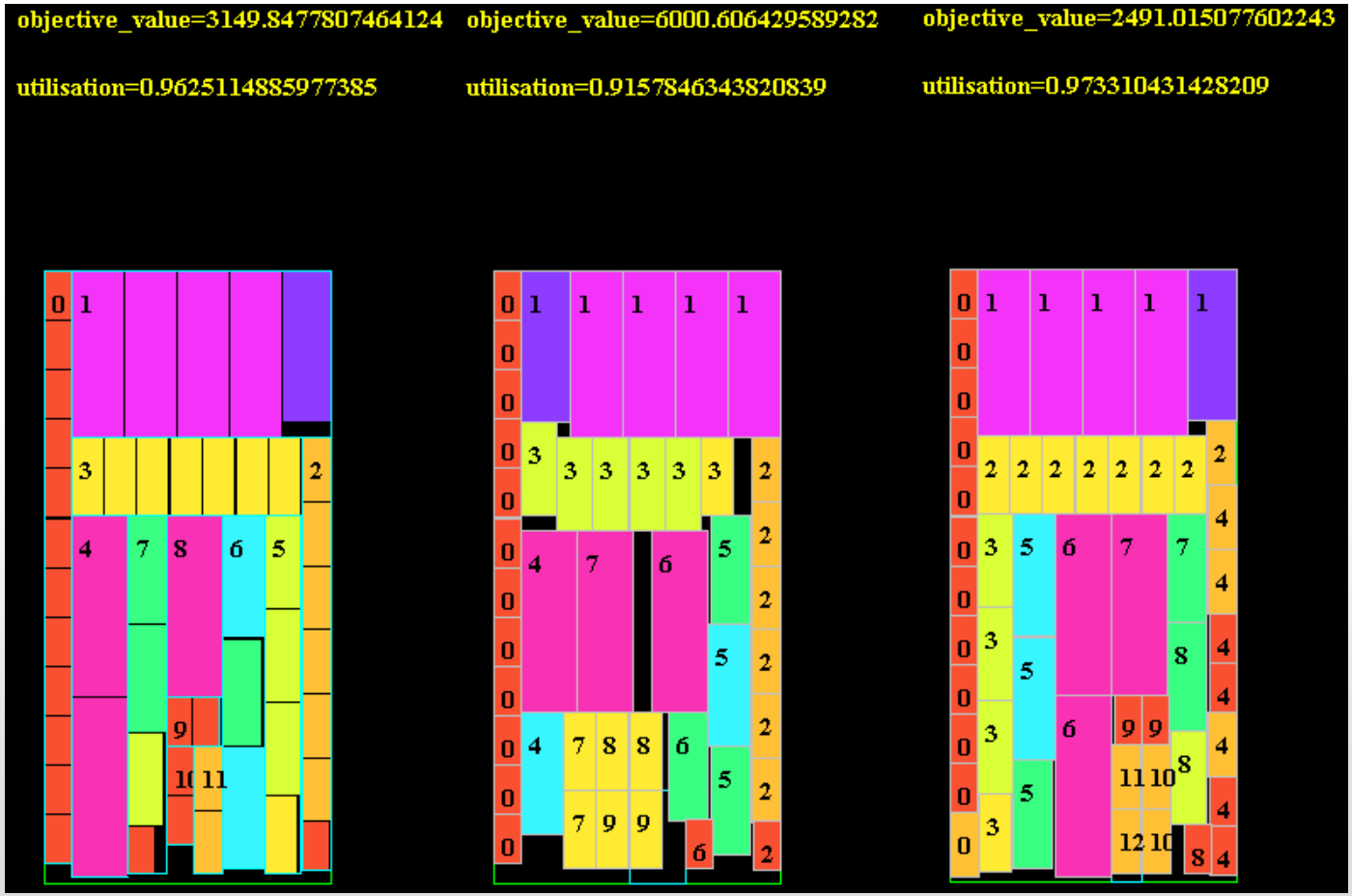
- The adhoc strip generation approach enables the exploitation of compacting that is precluded in the standard SGCKS approach
- Repeat this graph with SOCKS initial solutions in GP



UNIVERSITY OF Southampton
School of Mathematics

Combining guillotine cut and adhoc strip generation

- 1) Generate an initial solution using the fast SGCKS
- 2) solution is translated into adhoc approach (different gene interpretation)
- 3) Adhoc approach finds solution (which retains and improves upon the initial solution)



UNIVERSITY OF Southampton
School of Mathematics

SA approach iterations tradeoff

UNIVERSITY OF Southampton
School of Mathematics