Deep Reinforcement Learning for a Multi-Objective Online Order Batching Problem

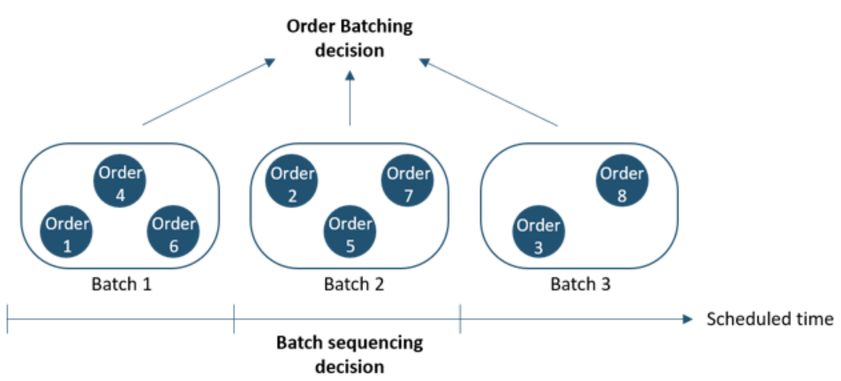
Martijn Beeks, Reza Refaei Afshar, Yingqian Zhang, Remco Dijkman (TU/e)

Claudy van Dorst, Stijn de Looijer (Vanderlande)

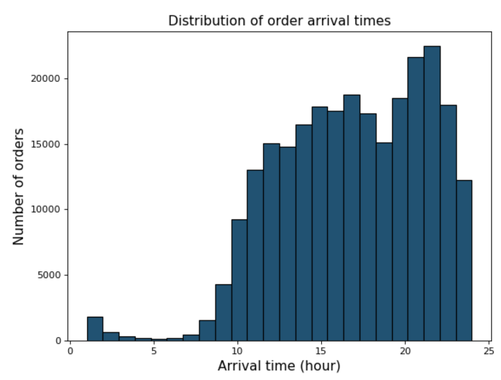
Problem description

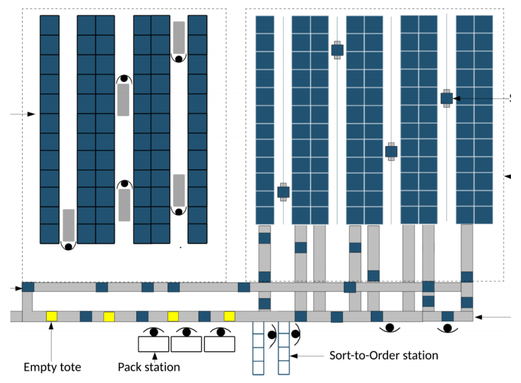
Online order batching problem with two objectives:
>Minimize late orders
>Minimize order picking costs

Real time allocating orders to batches



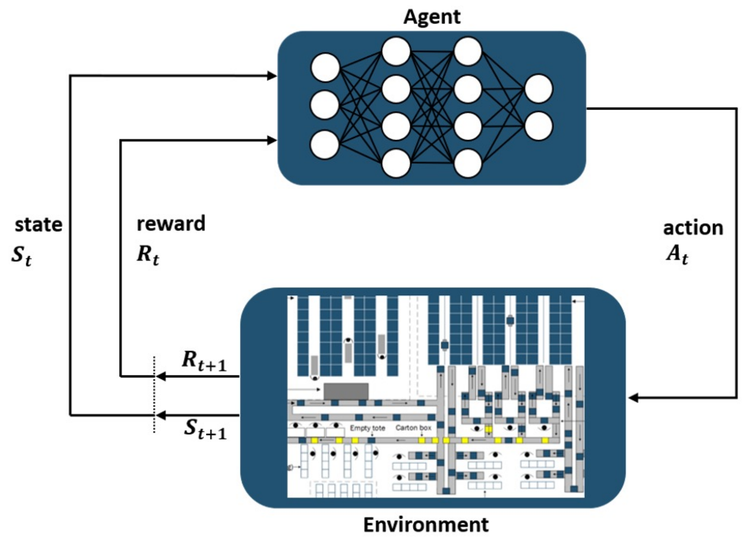
Challenges
>Multi-objective problem
>Large instance size
>Dynamic environments
>Online variant of the order batching problem





Methodology

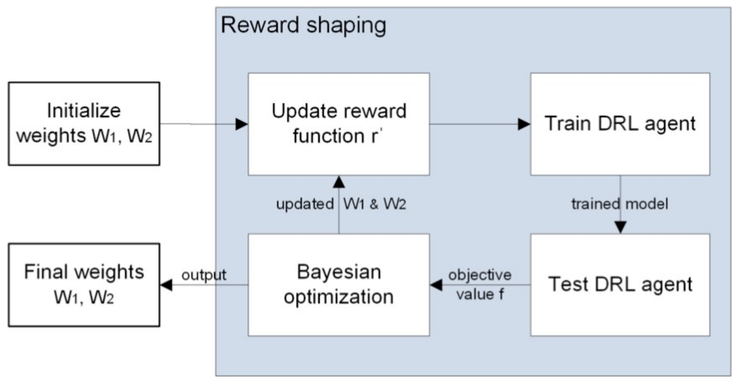
Literature review
>Exact methods are computational too complex
>Most heuristic methods consider single objective
>Deep Reinforcement Learning demonstrates interesting behavior (Cals et al., 2020)



1. Deep Reinforcement Learning
>Actions: pick-by-order / pick-by-batch
>State information: orders, time
>Reward function →

$$r'(s,a,s') = \begin{cases} W_1 \times (-1.5) & \text{if} tardy_o = 1 \\ W_1 \times (1 - \frac{u}{|O|})^2 & \text{If episode terminates} \\ -0.5 & \text{if infeasible action} \\ W_2 \times (-\frac{(1-v/N)}{50}) & \text{If order is picked} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

2. Deep Reinforcement Learning with reward shaping (RS)
>Reward shaping is a difficult process in multi-objective DRL
>Finding a reward function is presented as a Hyper-optimization problem
>This is solved using a Bayesian Optimization problem

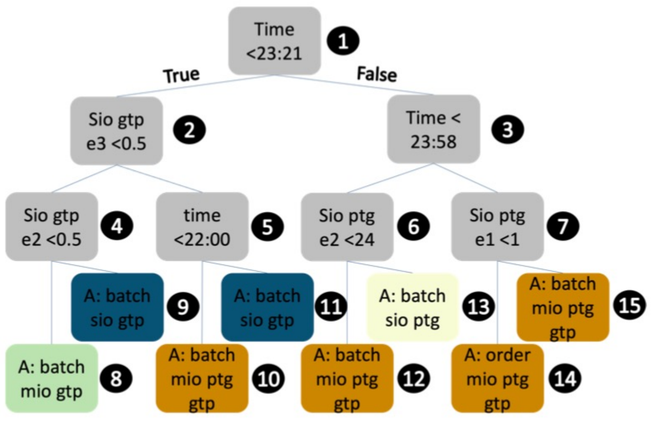


Results and Discussion

>BOC heuristic: batches based on similarity of orders (Huang et al., 2017).
>DRL + Reward Shaping (RS) outperformed other methods significantly (p=0.05)

Model	Setting A		Setting B		Setting C		Setting D	
	Tardy orders (%)	Picking costs	Tardy orders (%)	Picking costs	Tardy orders (%)	Picking costs	Tardy orders (%)	Picking costs
BOC	3.17 (1.1)	46.80 (0.8)	6.24 (1.4)	49.00 (0.9)	11.80 (1.8)	45.42 (0.9)	23.91 (0.3)	35.40 (0.2)
LST	17.6 (0.5)	65.12 (0.7)	18.7 (0.6)	68.83 (0.8)	21.15 (0.9)	67.21 (0.9)	25.76 (0.3)	53.34 (0.3)
GVNS	6.82 (0.2)	47.41 (0.9)	13.05 (0.5)	48.92 (0.9)	18.78 (1.5)	45.48 (1.0)	26.46 (0.2)	35.52 (0.2)
DRL	2.03 (0.8)	48.65 (0.8)	4.18 (0.8)	50.70 (1.0)	6.65 (0.8)	**42.22 (0.7)**	13.39 (0.4)	34.15 (0.2)
DRL + RS	**1.68 (0.7)**	**43.02 (0.6)**	**2.60 (0.7)**	**46.87 (0.8)**	**5.53 (1.1)**	42.90 (0.6)	**12.40 (0.2)**	**33.96 (0.2)**

Analysis of learned policy
Using a Decision Tree, logic from the DRL approach is derived
- During a day, focus on minimizing order picking costs
- End of a day, focus on minimizing tardy orders



Conclusion
"DRL with RS has found to be well capable to learn to address the multi-objective online order batching and sequencing problem

Future work
>Automate state reward action formulation
>Do we need deep NN's?