7th Student Symposium on Mechanical and Manufacturing Engineering, 2019

Batching Using Reinforcement Learning

M. Hildebrand, M. Sarivan, J. Frendrup
Department of Materials and Production, Aalborg University, DK

1. Introduction

When performing batching of products that have varying weight, the primary goal is to achieve a desired package weight. In an effort to achieve the desired goal weight, companies utilise rigid distribution rules that require tuning for each specific batch, with different weight distributions. In an effort to reduce the setup time for a new batch, and possibly further reduce giveaway, this article investigates the implementation of reinforcement learning (RL) in a batching environment.

2. Reinforcement learning

The RL method used in this paper is known as Q-learning. The Q-learning algorithm uses an update based adaption of the Bellman equation which is as follows:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma(\max(Q(s',a') - Q(s,a))$$

where the Q-value of a state given an action, is expressed through immediate reward, and a prediction of future Q-values. $Q(s,a)$ denotes the Q-value for the state, s, given action, a. α is the learning rate parameter, r is the immediate reward, and $\max(Q(s',a'))$ is the maximum predicted Q-value, given the next state, s', and the action that will lead to said maximum Q-value, s'.

The environment of the problem is defined as:

- The products which needs batching,
- A conveyor belt transports products into a batching cell,
- Trays which are to hold the products while being batched,
- The target weight of the trays,
- A manipulator for sorting the products is referred to as the agent,
- The giveaway, which is extra, or lack of weight in a tray,
- The distribution of the products with a given mean and span.

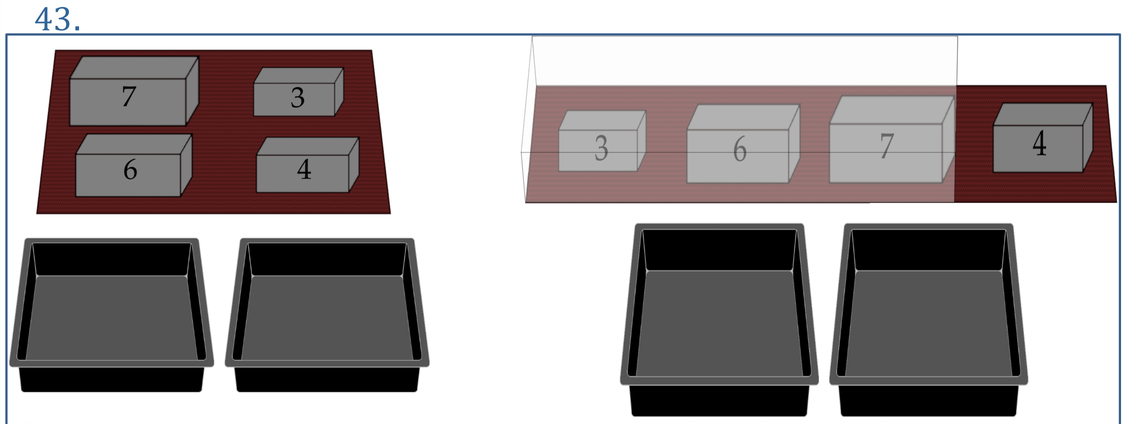
The goal of the Q-learning algorithm is to increase the reward where the reward function is given by the equation:

$$r = \frac{1}{\sqrt{(\text{Target weight} - \text{trayweight})^2}}$$

Iteration 1 consist of 4 products with the weight values $[4, 3, 7, 6]$, which are to be batched into 2 trays, which the lower left figure shows.

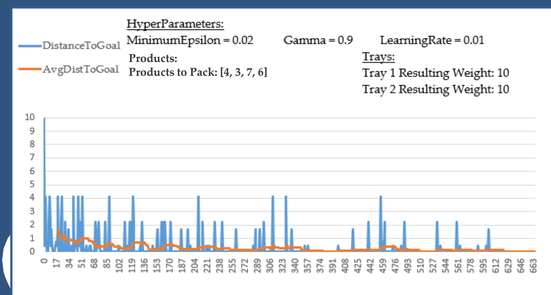
In iteration 2 the products weight values changes from a fixed set to a distribution, which has the mean, $\mu = 5$, and standard deviation, $\sigma = 2$, additionally the environment is adapted to a first in first out system, resulting in a reduction of the agent actions, by limiting the interactable products to one. This scenario is shown in the figure below on the right-hand side.

Iteration 3 increases the complexity by having 4 trays and 500 products, where the weight is described as having $\mu = 253$, and $\sigma = 43$.

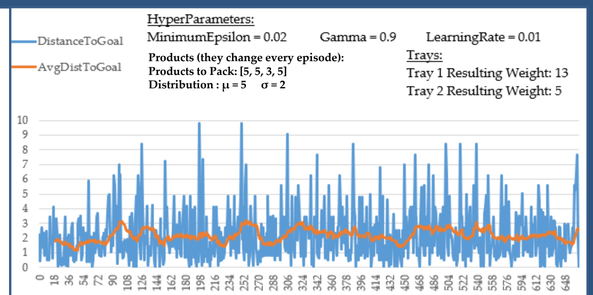


On the left-hand side is the scenario for iteration 1 shown, and the scenario of iteration 2 on the right-hand side.

3. Development results

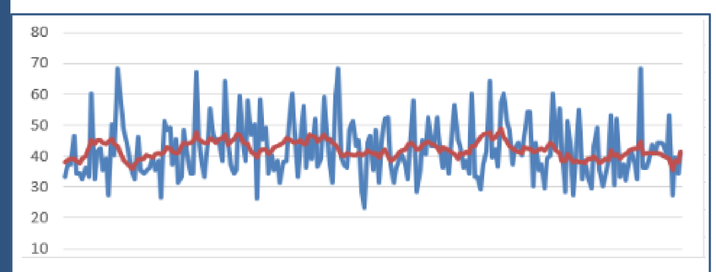


The result of iteration 1 with affixed tuning parameters.



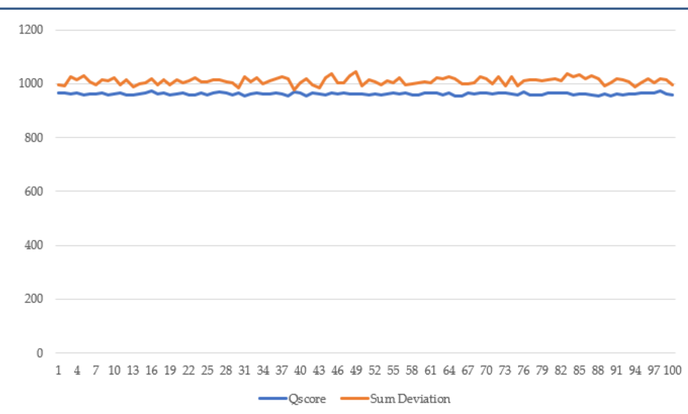
Plot showcasing the behaviour of QL in the context of Iteration 2

In iteration 1, an episode of training consists of batching the 4 available products. The optimal solution of the environment, is an equal distribution of 10 in each tray, and this was achieved using the tuning parameters shown on the left figure. The plot also shows the convergence towards zero giveaway when using Q learning. Due to large size of weight distribution in product weight, as described in Iteration 2, Q learning is deemed ineffective as it does not converge to a satisfactory result (top right figure). Deep Q-Network (DQN) algorithm is employed to fix this problem from Iteration 2. DQN is a combination between Q learning and artificial neural networks. The DQN was trained for 1.5 million time-steps, but gross overfitting occurred, and the best results were achieved after 100.000 time-steps. 200 were selected for the plot on the left.



Giveaway (red) and reward (blue) when using DQN in Iteration 3, over 200 episodes

4. Conclusions



A comparison of the Qscore from the e-weighing regulations and the giveaway of a random sample, across 100 episodes.

The final test results can be seen in the figure above, which clearly show that all sampled trays pass according to the e-weighing regulations, forming the basis of the conclusion that the current environment setup of iteration 3 is sufficiently solved. The average giveaway per episode during testing was found to be ~1000 and considering that each episode consists of approximately 126 finished trays, makes the average giveaway equal to less than 10 per tray, per episode. From this, it is concluded that the application of RL in a batching environment, holds the potential of improving or surpassing current batching methods.

Acknowledgement

The authors of this work gratefully acknowledge Grundfos for sponsoring the 7th MechMan Symposium