Part II Project Progress Report: Reinforcement Learning meets Balls into Bins

Andor Vári-Kakas (av543@cam.ac.uk)
Primary supervisor: Thomas Sauerwald
Secondary supervisor: Dimitrios Los
Director of Studies: Dr John Fawcett
Overseers: Alan Blackwell, Srinivasan Keshav

2 February 2021

1 Overview of the project

My project is about optimising free parameters of different stochastic load balancing protocols. The project studies a well-known theoretical abstraction of the field called balls into bins: m balls are sequentially placed into n bins according to a parametric, randomised protocol, and the goal is to optimise an objective function of the (final) load distribution ("how balanced it is"), by choosing suitable parameters for the protocol. An example setting is the Two Thinning protocol, where for each ball, a primary bin is offered, and according to the parameters of the protocol (this is what has to be optimised), it either accepts the bin, or chooses to have the ball placed in a completely random bin instead. The focus of the project is evaluating how well reinforcement learning (RL) based methods can optimise these free parameters, and compare them to classical approaches for choosing the parameters, such as dynamic programming, or greedy algorithms.

2 Progress

I consider the project to be on schedule, but the order of the work items have been exchanged at places, and the emphasis has been shifted occasionally. According to the schedule I should be starting the evaluation, and I should have finished the implementation part. While working on the implementation, it turned out that interleaving parts of implementation and evaluation is better, e.g. because to decide whether to further finetune an algorithm, or move to a new setting, non-trivial evaluation was necessary. Hence, the implementation is not yet fully finished (I would like to finish implementing one more setting), but the basis for evaluation is set up already.

3 Difficulties

We initially thought that the emphasis will be on comparing (mostly ready-made) reinforcement learning algortihms. First of all, it quickly turned out that non-neural network based algorithms are infeasible for most of the settings I wanted to consider, due to the large state space. For reasons such as robustness and good fit to combinatorial optimisation problems (such as this one), I decided to start with Deep Q-Learning (DQN), a version of neural network based RL. Starting out with the seemingly simple DQN, I later realised that to achieve reasonable performance, there are more than a dozen hyperparameters (in the neural network, and the RL framework) that I have to set suitably. Hence, we decided to restrict our focus mostly to DQN (from the family of RL algorithms), and look at its rich nature in more detail. Apart from this unexpected difficulty in adjusting the hyperparameters (we expected it to take some time, but much less), the other challenges have been accounted for in the proposal.

4 Work completed

Now I give a brief summary of my work so far. First, I implemented some non-parametric protocols, such as Two-Choice. Then, I implemented some classical methods for finding the optimal parameters in Two-Thinning, including dynamic programming, that gives a perfectly optimal solution but only for small values of n and m. After that, I moved on to the heart of the project, which is the DQN algorithm for Two-Thinning. This involved experimenting with the hyperparameters, e.g. reward shaping for RL, and trying different neural network architectures. I then created a general OOP framework for comparing different strategies. Most recently I have been working on settings other than the vanilla Two-Thinning. First, I restricted the amount of information the protocol has about the current load distribution, and investigated how it influences the performance of different algorithms for choosing the parameters. Then, I extended Two-Thinning to K-Thinning, where there are not 1, but $K-1$ bin offers that can be rejected by the protocol. Finally, currently I am working on a graph based variant of Two-Thinning, where the bins offered for a ball are chosen randomly only from a proper subset of all pairs. This is more challenging to turn into an RL problem, partly because it is not optimal to simply choose a “threshold” to make the decisions.