Exploration on Optimal Stopping and its Variations

Joonyoung Bae

University of Hong Kong

n99joon@connect.hku.hk

Abstract

In optimal stopping problems, the aim is to decide when to stop in a sequential process, such that the reward is maximized. In this paper, one variation of the optimal stopping problem called Pandora's Box problem is explored. In his paper, Weitzman [1] proved optimal stopping and selection strategies for the Pandora's Box problem, with the restriction that every box value has to be sampled from independent distributions. When the box value distributions are correlated, every information gain brings extra positive informational externality of the posterior distributions and a simple search strategy is difficult to be applied [1]. In most cases, it is challenging to know how far the strategy is from the best possible strategy. In this paper, Reinforcement Learning (RL) methods are used to produce near-optimal values for both independent and correlated Pandora's Box Problems, which can be used as benchmarks or reference points.

1. Introduction

1.1 Project Background

The optimal stopping problem can be described as the following formulation: given the distributions of the values, what is the best stopping rule that yields the maximum expected reward(or the minimum expected cost). With this minimum structure, many variations have been introduced such as the Pandora's Box problem [1], which is the focus of this paper.

The optimal strategy that yields the best expected reward has already been introduced by Weitzman [1] using the solution concept of the reservation price. It is already proven that given independent box value distributions, the selection strategy(which box to open next) and the stopping strategy(whether to stop or open the next box) can both follow the reservation price in order to guarantee maximum expected reward.

However, many real-life scenarios have value distributions that are correlated. For example, the probability of raining in the afternoon is higher if it was cloudy in the morning. The Pandora's Box problem with correlated distributions is thus being actively studied. The theoretical studies on the correlated Pandora's Box problem is more complicated as the optimal strategy has to take account of the distributions that change every step. A simpler approach is the Partially Adaptive strategy, which gives the best strategy given a fixed order of opening the boxes. The full approach that deals with both the order of opening and the selection of the box is called the Fully Adaptive strategy [2]. The ultimate goal is to explore PA and FA strategies, and make approximation guarantees.

In 2020, Chawla et al. [2] provided Partially Adaptive strategies of the correlated Pandora's Box problem that guarantee constant factors approximation with polynomial sample complexities. In 2022, Chawla et al. [3] suggested constant-factor approximation by reducing the problem into a Min Sum Set Cover with feedback problem and then into a Uniform Decision Tree. In 2023, Gergatsouli and Tzamos [4] proved that it is possible to directly apply the optimal approach by Weitzman [1] in the correlated case as well, which is simpler and with improved approximation guarantees.

1.2 Project Objective

Including the ongoing studies on the correlated Pandora's Box problem, it is academically challenging to determine whether the current best strategy is optimal or not. Many approaches are being taken such as proving the upper and lower bounds or rigorously proving the optimality.

In this paper, the objective is to provide a new insights on the Pandora's Box problems by using the Reinforcement Learning with deep neural networks. Different models of reinforcement learning were utilized to provide a reference point of how optimal can the result be of the problems. Even though it is hard to retrieve or analyze what strategy is being used by the model, the average return that the trained agent gets can be used as a good benchmark. For example, if there is a trained deep learning agent that can get higher rewards than the current best strategy on average, it can be a sign that there are more improvements to be made.

There are two main deliverables of the paper:

- Analysis of independent Pandora's Box problem using Deep Q-Learning (DQN) and Deep State-Action-Reward-State-Action (Deep SARSA) models.
- Analysis of correlated Pandora's Box problem using DQN and Deep SARSA models.

2 Preliminaries

2.1 Optimal Stopping

As the topic of Optimal Stopping has been studied from early 1900s, the definitions may vary across literatures. The formal definition of Optimal Stopping in this paper follows the notation from the paper by Ferguson [5].

The Optimal Stopping problems are defined by

(i) (**observation**) A sequence of random variables, $X_1, X_2, ...$, with known joint distributions, and

(ii) (**reward**) A sequence of real-valued reward functions,

$$y_0, y_{1(x_1)}, y_{2(x_1,x_2)}, \dots, y_\infty(x_1, x_2, \dots).$$

Given the above objects, the stopping problem is defined as follows.

The player can decide $n = 0,1, \dots, \infty$, which stands for the number of terms of $X_1, X_2, ...$, that will be observed. The reward $y_n(x_1, \dots, x_n)$ (possibly negative) is known to the player at every n and the possible actions are to accept the current reward y_n or to continue observing the next realization of X_{n+1}. It is possible to not observe any term and accept a constant reward y_0.

The problem is to choose the best N that has the best reward y_N. The probability of stopping after observing n terms is defined as $\varphi_n(x_1, \dots, x_n)$. A randomized **stopping rule** is the sequence of these probabilities:

$$\boldsymbol{\varphi} = (\varphi_0, \varphi_1(x_1), \varphi_2(x_1, x_2), \dots) \tag{1}$$

It is non-randomized when every $\varphi_n(x_1, x_2, \dots x_n)$ is either 0 or 1.

The aim is to find the strategy to maximize the expected reward , $V(\boldsymbol{\varphi})$, as the optimal N cannot be found without knowledge of all the realizations of $\boldsymbol{X}$.

$$V(\varphi) = E\ y_N(X_1, \dots, X_N) \tag{2}$$

2.2 Pandora's Box Problem

Pandora's Box Problem is a variation of the Optimal Stopping problem. Below is the definition of the Pandora's Box problem follows Weitzman [1].

- (**Box**) There are n boxes, each having unknown rewards inside.
- (**Distribution**) Each box i has its own probability distribution X_i of its **reward.**
- (**Realization with Cost**) The true value in the box can be revealed by paying the opening cost c_i
- (**Sequential-Search**) Sources can be searched sequentially in any desired order.
- (**Recall**) The maximum searched reward so far is selected when it has been decided to stop.
- (**Fallback Reward**) An initial amount x_0 exists which can be collected without any sampling.
- (**Objective**) To find the optimal sequential search and stopping rule/strategy that maximizes the expected present discount value.

In this paper and in many papers, the objective is often modified to maximize the expected payoff without discount, which is (maximum searched x_i) – (total cost paid) for simplicity.

$$y_n = \max_{i \le n} x_i - \sum\nolimits_{i \le n} c_i \tag{3}$$

2.3 Scenarios

Scenarios are useful tools to describe a joint distribution in a correlated setting. The following definition is adopted from Gergatsouli and Tzamos [4].

Each scenario, s ,from the set of all possible scenarios, S , is a possible outcome from a distribution D. In other words, D is supported by $|S|$ numbers of vectors, $(\boldsymbol{v}^s)_{s \in S}$, and sometimes the abuse of notation follows as each scenario being sampled from the distribution D.

For example, assume there are two boxes with correlated distributions X_1 and X_2. They are correlated such that they can only have both zeros or both ones with equal probability.

Then the distribution $D = (X_1, X_2)$ can be simply shown with two scenarios $(0,0)$ and $(1,1)$ each with p=0.5.

For further simplicity, every scenarios are assumed to have equal probability. It is possible to reduce any set of scenarios into set of scenarios with equal probabilities by increasing the number of the scenario with higher probability.

2.4 Deep Q-Learning (DQN)

Deep Q-Learning (DQN) was first introduced by Deepmind in 2013 [6].

The training is done to reduce the difference between predicted and target Q-values. The predicted Q-value is calculated by deep neural networks and the target Q-value is calculated by the Bellman Equation.

The update rule for DQN is

$$\begin{gathered} Q(s,a) \leftarrow Q(s,a) + \\ \alpha[r + \gamma \max_{a'} Q_\theta(s',a') - Q(s,a)] \end{gathered} \tag{4}$$

where r = reward, γ = discount factor,

(s,a) = current state and action pair. In the example of a tic-tac-toe game, state can be the array of all the coordinates of x and o ticks, action can be all the possible moves to be made.

(s',a') = next state and action pair and s' is the next state resulted by taking action a at state s

$Q_\theta(s',a')$ = Predicted Q-value function, which is the estimated expected cumulative reward by taking the action a' at state s'

α = learning rate.

2.5 Deep State-Action-Reward-State-Action (Deep SARSA)

SARSA algorithm was first introduced in 1994 [7].

It is an algorithm that also uses Q-learning for calculating expected cumulative rewards. The difference is that SARSA explores the environment according to its training policy, while DQN explores by following the maximum predicted Q-value. Deep SARSA model is when the SARSA algorithm uses deep neural networks to approximate the Q-values for state-action pairs.

The update rule for Deep SARSA is

$$\begin{gathered} Q(s,a) \leftarrow Q(s,a) + \\ \alpha[r + \gamma\, \mathrm{Q}(s',a') - Q(s,a)] \end{gathered} \tag{5}$$

The only difference in the update rule is that it learns the Q-value from the next state-action pair from its current policy.

2.6 Reservation Price

The reservation price was introduced as the solution concept to the Pandora's Box problem by Weitzman [1].

The simple definition of reservation price z_i of the box i is as follows.

$$E[(z_i - x_i)^+] = c_i \tag{6}$$

$where\ (x)^+ := \max\ (0,x)$

It means z_i is the threshold value of the maximum value gained so far, at which the player's expected gain of opening the box i equals the opening cost of the box.

A more formal definition with distribution is as follows.

$$\int_{z_i}^{\infty} (x_i - z_i) dX_i(x_i) = c_i \tag{7}$$

2.7 Reservation Price in a Correlated Pandora's Box Problem

In January 2023, Gergatsouli and Tzamos [4] have shown that the reservation price concept can also be applied in the correlated settings to get a good approximation guarantee with slight modifications.

The newly defined reservation price with scenarios is as follows.

$$z_i = \min_{A \subseteq S} \frac{c_i|S| + \sum_{s \in A} x_i}{|A|} \tag{8}$$

It now considers the ratio of the set of current possible scenarios A and the set of all scenarios S, and thus can calculate the expected value x_i from the information of possible scenarios after every step of the problem.

The formal derivation of the equation can be found in the paper [4].

3 Methodology

The reinforcement learning for all problems was done following below procedures.

1) Define a custom environment according to the problem specification in the OpenAI **gym** library
2) Build a deep neural network model using **TensorFlow** and **Keras** by Google. Different dimensions of the neural network are tested for the best output.
3) Build an agent with **Keras-rl2** libraries according to the choice of the RL method.
4) Train the agent and tune hyperparameters with values randomly drawn from specified distributions at every run.
5) Do performance analysis by simulations using **python** code.

Specifically, DQN was chosen for comparison as the use of experience replay buffer greatly increases the efficiency of learning and it has been one of the algorithms to achieve the best outcomes.

Deep SARSA was chosen for comparison as it is suitable for problems where the agent's action has direct impacts on the environment and it prevents overestimation of action values.

The programs were run on Google Colab in Jupyter Notebook files[1].

4 Results

4.1 Pandora's Box Problem : Independent

The original Pandora's Box problem with independent value distributions was first studied with the following specifications.

4.1.1 Specifications

Number of Boxes	6
Cost of Boxes	0.1
Distribution of Box Value	~ Uniform[0,1]
Reservation Price	0.55
State	(1x7) Array with two parameters
Action	7 possible actions

Table 1: Specification of Independent Pandora's Box Problem

The possible box value drawn from uniform distribution of range [0,1] is any real number.

The state of the agent is a coordinate of shape (1x7) with below parameters combined:

- **(Boxes Opened) 1x6 Array** where every index represents whether the box of that index is opened or not
- **(Max Value) A Real Number** which is the best value obtained so far

The possible actions that can be taken at any state is either open any box or choose to stop, which in total are 7 actions.

A big negative reward has been given if the agent opens a box already opened.

4.1.2 DQN

The neural network for DQN had two ends for input and output and two hidden layers with 24 nodes each.

Below figures show the MAE and Loss during the training of DQN agent. MAE is the difference measured by the validation set during training. Loss is the average squared difference between the predicted and target Q-value during training. The error bound shown in the graphs is inevitable due to the random nature of sampling values from the distribution.

[1] github.com/n99joon/Pandora_Box_RL

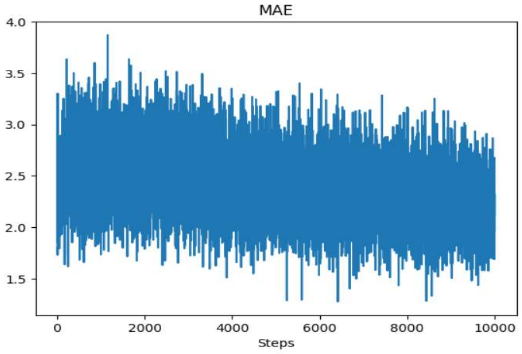


Figure 1: Mean Absolute Error(MAE) of DQN Model Training

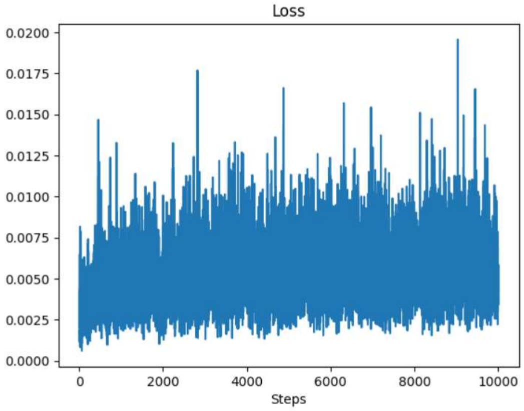
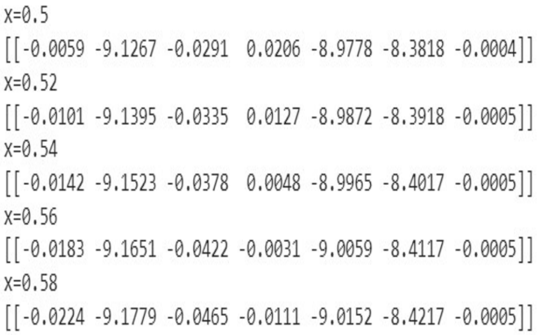


Figure 2: Loss of DQN Model Training



```
x=0.5
[[-0.0059 -9.1267 -0.0291  0.0206 -8.9778 -8.3818 -0.0004]]
x=0.52
[[-0.0101 -9.1395 -0.0335  0.0127 -8.9872 -8.3918 -0.0005]]
x=0.54
[[-0.0142 -9.1523 -0.0378  0.0048 -8.9965 -8.4017 -0.0005]]
x=0.56
[[-0.0183 -9.1651 -0.0422 -0.0031 -9.0059 -8.4117 -0.0005]]
x=0.58
[[-0.0224 -9.1779 -0.0465 -0.0111 -9.0152 -8.4217 -0.0005]]
```

Table 2: Q-Values when Three Boxes are Opened with Different Maximum Values Gained

In the above Table, x represents the highest value gained so far after opening three boxes. The (1x7) array shows the Q-Value of each action at that state, with index 1 to 6 representing the box 1 to 6 and the last index representing the action to stop.

According to the Reservation Value calculated, it is better to keep opening the box if x is less than 0.55 and stop otherwise.

From Table 2, it is noticed that the DQN agent also learned after 0.55, it is better to stop. The agent will choose the highest Q-Value as the next action and at this state, and at $x = 0.54$, it will open the box 4, and at $x = 0.56$ it will choose to stop.

4.1.3 Deep SARSA

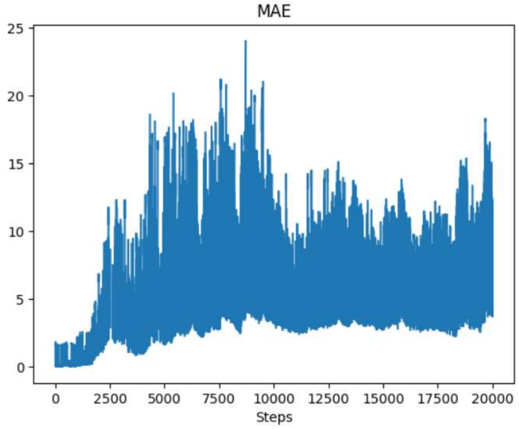


Figure 3: Mean Absolute Error(MAE) of Deep SARSA Model Training

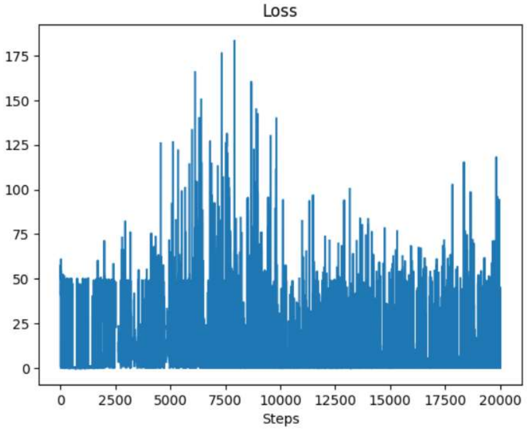


Figure 4: Loss of Deep SARSA Model Training

After training, Deep SARSA agent was not able to make decisions according to the current state. It only had a set number of boxes it will open, regardless of the current maximum value gained. The problem might have been from the inherent randomness of the rewards, which hinders the learning as Deep SARSA learns from the policy it is currently following. No big improvement was made with different neural network structures, hyperparameters, and steps taken during training.

4.1.4 Performance Analysis

In this section, the average performances of opening a set number of boxes (Opening 2 boxes yields the best expected reward)(**S1**), the trained DQN model (**S2**), the strategy with Reservation Price (**S3**), and the best possible reward (best value – opening cost) (**S4**) are compared.

With 100,000 randomly generated box distributions, below are the performances of each strategies.

S1	0.467
S2	0.529
S3	0.55
S4	0.757

Table 3: Average Reward of 100,000 Simulations

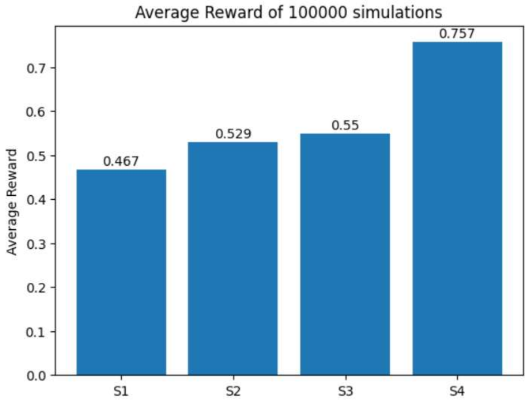


Figure 5: Average Reward of 100,000 Simulations

The trained DQN model has demonstrated the capacity of a Reinforcement Learning algorithm to be used as a good benchmark as the difference with the Reservation Price Strategy which guarantees maximum average reward is trivial. Further hyperparameters and model structure tuning can also increase the performance of the DQN model.

4.2 Pandora's Box Problem : Correlated

For the Pandora's Box problem with correlated value distributions, discrete value distributions were chosen to design it as a set of scenarios. Also, the number of boxes are reduced to 3, so as to keep the number of scenarios small.

The Pandora's Box problem with correlated value distributions was studied with the following specifications.

4.2.1 Specifications

Number of Boxes	3
Cost of Boxes	1.2
Distribution of Box Value	~Uniform{1,2,3,4,5}
Reservation Price	2
State	(1x7) Array with three parameters
Action	4 possible actions

Table 4: Specification of Correlated Pandora's Box Problem

The possible box value drawn from discrete uniform distribution from set {1,2,3,4,5}.

The state of the agent is a coordinate of shape (1x7) with below parameters combined:

- **(Expected Values) 1x3 Array** which demonstrates the expected value of boxes based on the current possible scenarios. If the box is already opened, it is set as -10.
- **(Opened Boxes) 1x3 Array** which shows which box is opened
- **(Max Value) An Integer** which is the best value obtained so far

The possible actions that can be taken at any state is either open any box or choose to stop, which in total are 4 actions.

A big negative reward has been given if the agent opens a box already opened.

The scenarios, or the correlated value distributions, were randomly generated at every iteration, in a way that they keep individual average as 3 if sampled independently.

4.2.2 DQN

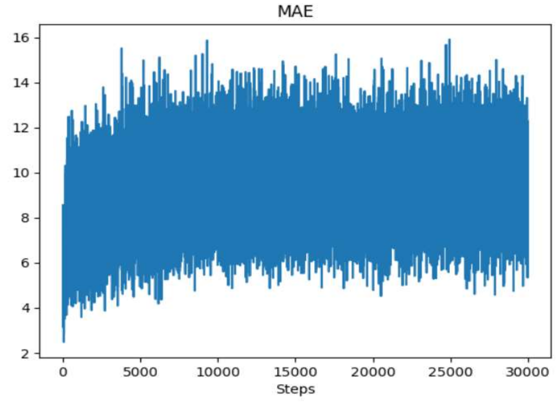


Figure 6: Mean Absolute Error(MAE) of DQN Model Training

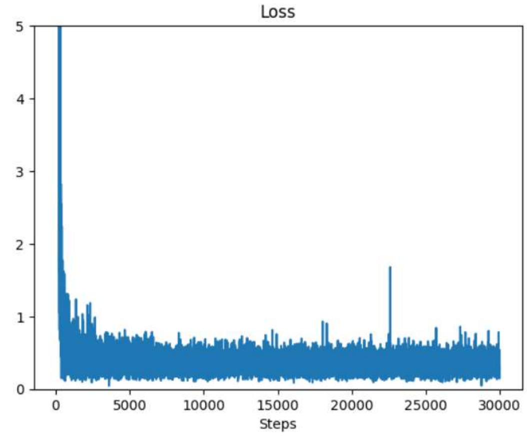
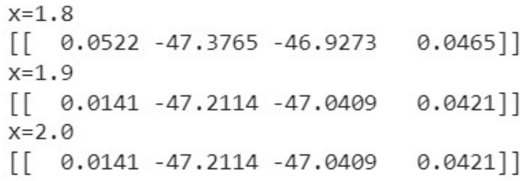


Figure 7: Loss of DQN Model Training

It is clear that the correlated Pandora's Box problem still contains uncertainties in the distribution so that the error bound exists in the training. However, the loss function settles quick as shown in Figure 7, because the DQN agent is calculating the expected value of each box after every step from the information gained in previous step. It rules out all the scenarios that do not agree with the value revealed in the previous step, which makes a more accurate prediction on expected values of unopened boxes possible. It is using the *Fully Adaptive* approach which makes a tree of possible sequence of events [4].



```
x=1.8
[[  0.0522 -47.3765 -46.9273   0.0465]]
x=1.9
[[  0.0141 -47.2114 -47.0409   0.0421]]
x=2.0
[[  0.0141 -47.2114 -47.0409   0.0421]]
```

Table 5: Q-Value Table with only First Box Unopened

The state of the above Q-Value table is when only the first box is unopened and its expected value is 3, with the highest value sampled so far is 1.8, 1.9, and 2.0. In this model, the reservation price is shown as 1.9, which is close to the actual reservation price 2.0 of this setting.

4.2.3 Deep SARSA

Deep SARSA again could not produce a meaningful model after training. The training progress showed that the model was falling into states that guaranteed big negative rewards and the policy was not updated in a meaningful way.

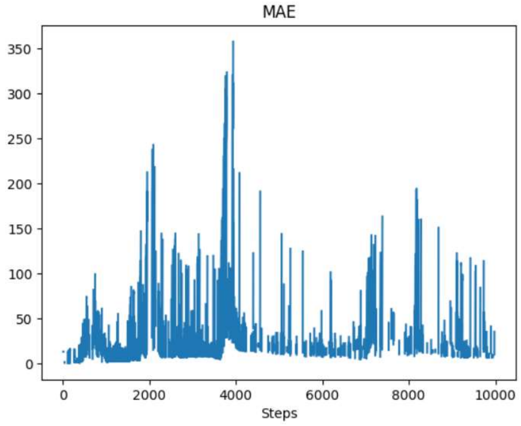


Figure 8: Mean Absolute Error(MAE) of Deep SARSA Model

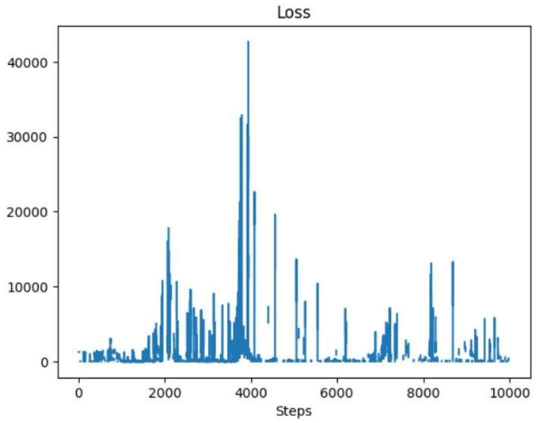


Figure 9: Loss of Deep SARSA Model

After training, the Deep SARSA model chose to not open any box in the initial stage and it indirectly shows that the best policy the model learned is to quit right away. It is true that quitting without opening any box guarantees non-negative reward, and Deep SARSA might have trained to gain the best reward by choosing it. During training, it was frequently seen that the model received large negative rewards by opening the same box again, and it could have adjusted the policies to open any box to decrease and always choose a safer reward. Many different epsilon values have been tried to make the agent try different paths, but the result showed no improvements.

4.2.4 Performance Analysis

In this section, the average performances of opening set number of boxes (Opening one box is of maximum expected reward) (**S1**), the trained DQN model (**S2**), the *Fully Adaptive* algorithm

[4] (**S3**) and the best possible reward (best value – opening cost) (**S4**) are compared.

S1	1.81
S2	2.07
S3	2.035
S3	3.386

Table 6: Average Reward of 100,000 Simulations

The average reward of the DQN model proved to outperform the strategy of choosing set number of boxes to open. The *Fully Adaptive* algorithm is proven to be 5.828-Approximation to the cost minimization version of the correlated Pandora's Box problem [4]. The *Fully Adaptive* algorithm works by using the new Reservation Price shown in equation (8) for deciding which box to choose and whether to stop. The difference is that after opening every box, all the scenarios that do not agree with the realized value are removed from the set of possible scenarios to be considered in the next step. The algorithm in full detail can be found in Gergatsouli and Tzamos [4].

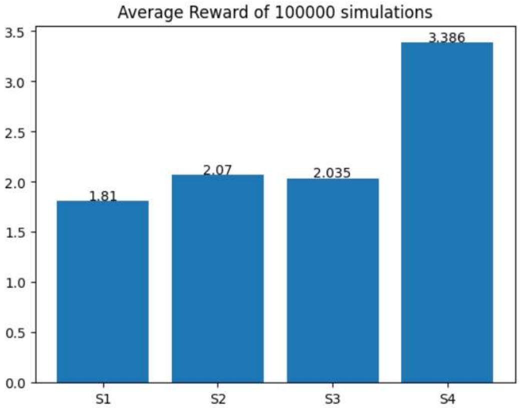


Figure 10: Average Reward of 100,000 Simulations

It is noted that the trained DQN model outperformed the *Fully-Adaptive* strategy that was recently studied to have a good performance guarantee [4]. This can be a great indicator of any improvements to be made to the algorithm or any existence of substitute algorithm that might have better performance guarantees. It is also possible that the marginal difference came from the uncertainties of the value realizations from distributions.

5 Conclusion

The application of Reinforcement Learning, specifically DQN, has been proven to be successful in providing a meaningful benchmarks to theoretically complicated problems. In correlated setting where many on-going researches are happening, the DQN outperformed the *Fully Adaptive* algorithm by a small margin. It thus proved that the Reinforcement Learning has a high potential to be used as the benchmark or performance goal when deriving new algorithms for complicated problems.

The trained Reinforcement Learning agents are shown to be able to derive the behaviors of optimal strategies or provide near-optimal reward predictions for comparison.

Thus, in many currently studied fields, such as the correlated Pandora's Box problem, Reinforcement Learning can be a good guidance if the problem can be meticulously formulated into Reinforcement Learning settings.

6 Future Work

The algorithms of Reinforcement Learning (RL) are improving rapidly. For example, many new variations of DQN algorithm, such as Double-DQN and Duel-DQN, have been proposed with better performance guarantees and less bias. Also, for specific settings such as continuous environment, DQN is not good at fully exploring the states and thus alternatives such as the Soft Actor-Critic algorithm can yield better results. In future works, comparison with more diverse models may improve the overall performances and thus provide better reference points for future researches.

One of the limitations of this project was that Reinforcement Learning becomes computationally heavy in an exponential rate if the number of boxes or distributions increases. It is true in scenarios that are more complicated and it contradicts the main purpose of using empirical approaches at theoretically complicated problems. More researches are to be followed on exploring various methods to formulate an established way of reducing an algorithm problem to a Reinforcement Learning problem.

The analysis of trained model can also be improved with further researches. The performance guarantees of RL models are good benchmarks, but it is challenging to understand the decision making logic or the strategy used by the model. The difficulty stems from the way those models are designed with neural networks and different propagating algorithms. It is often hard to derive the optimal strategy the model is following by observing the weights of the nodes in the neural networks, because the number of them is too big and the nodes are interconnected heavily to observe individual's impacts on the decision making process.

Acknowledgements

I would like to acknowledge and give deepest thanks to my academic supervisor, Prof. Huang Zhiyi, for his insights and guidance throughout the project.

References

[1] Weitzman, M. L. (1979). Optimal search for the best alternative. Econometrica: Journal of the Econometric Society, 641-654.

[2] Chawla, S., Gergatsouli, E., Teng, Y., Tzamos, C., & Zhang, R. (2020). Pandora's box with correlations: Learning and approximation. In 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS) (pp. 1214-1225). IEEE.

[3] Chawla, S., Gergatsouli, E., McMahan, J., & Tzamos, C. (2021). Approximating Pandora's Box with Correlations. arXiv preprint arXiv:2108.12976.

[4] Gergatsouli, E., & Tzamos, C. (2023). Weitzman's Rule for Pandora's Box with Correlations. arXiv preprint arXiv:2301.13534.

[5] Ferguson, T. S. (1967). Probability and mathematical statistics.

[6] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

[7] Rummery, G. A., & Niranjan, M. (1994). On-line Q-learning using connectionist systems (Vol. 37, p. 14). Cambridge, UK: University of Cambridge, Department of Engineering.

.