Memorial University

REINFORCEMENT LEARNING:

ASSIGNMENT 1

(Spring 2025)

by

Hoang Phi Yen Duong - 202396817

Tinh Thanh Bui - 202398612

(PhD students)

St. John's, June 18, 2025

Outline

Part 1: Stationary environments	
1. Greedy method with non-optimistic initial values	
2. Epsilon-greedy method with different choices of epsilon	
3. Greedy method with optimistic starting values	
4. Gradient bandit algorithm	
5. Comparison of four bandit algorithms	
Part 2: Non-stationary environments	
1. Gradual changes: Drift	
2. Gradual changes: Mean-reverting change	
3. Abrupt changes: Permuting the means	
4. Abrup changes: Reseting all action values	

Code is available on GitHub: Click here

PART 1:

STATIONARY ENVIRONMENTS

In this part, we consider a bandit problem consisting of 10 arms with stationary reward distributions. The means of arms are independently chosen from a N(0, 1) distribution (mean μ_i~N(0, 1)). Additionally, the reward distribution of each arm, arm i, for example, has a N(μ_i, 1) reward distribution (i = 1, ..., 10). We investigate four bandit algorithms, including the greedy method with non-optimistic initial values, the epsilon-greedy method, the greedy method with optimistic starting values, and the gradient bandit algorithm. Each algorithm is run for 2000 time steps with 1000 different simulations.

1. Greedy method with non-optimistic initial values

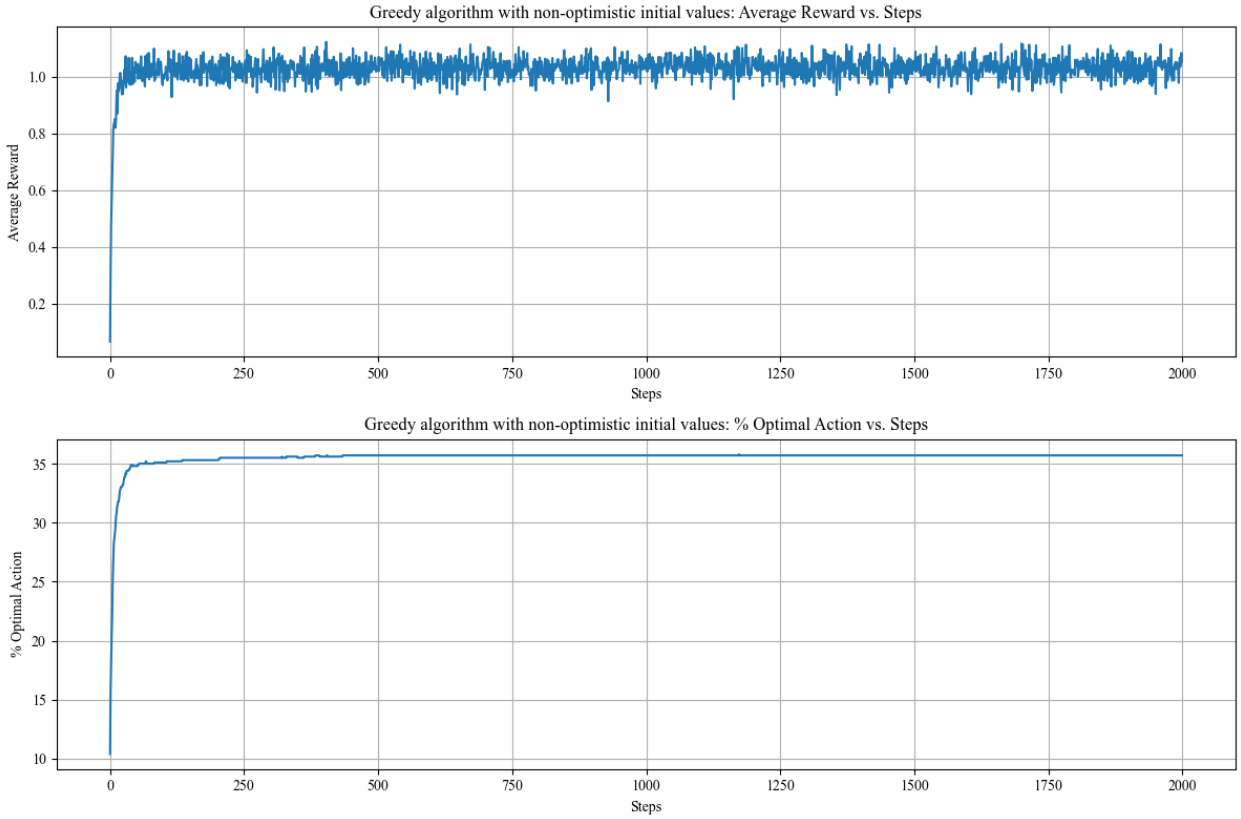

Fig. 1. *Greedy method with non-optimistic initial values*

In this simulation, we start by setting up the non-optimistic initial action values of all arms to 0. The reward plot in Fig. 1 shows that the average reward of the greedy method improves quickly

within the first 50 steps and stabilizes at nearly 1 after 100 steps. After 100 steps, the average rewards are not improved noticeably since the method gets stuck at local optimal actions. On the other hand, at stabilization, the method chooses the optimal action with a probability of 35%. We can conclude that the greedy method with non-optimistic initial values has good exploitation by exploiting the best current actions efficiently, but exploration cannot be used efficiently.

2. Epsilon-greedy method with different choices of epsilon

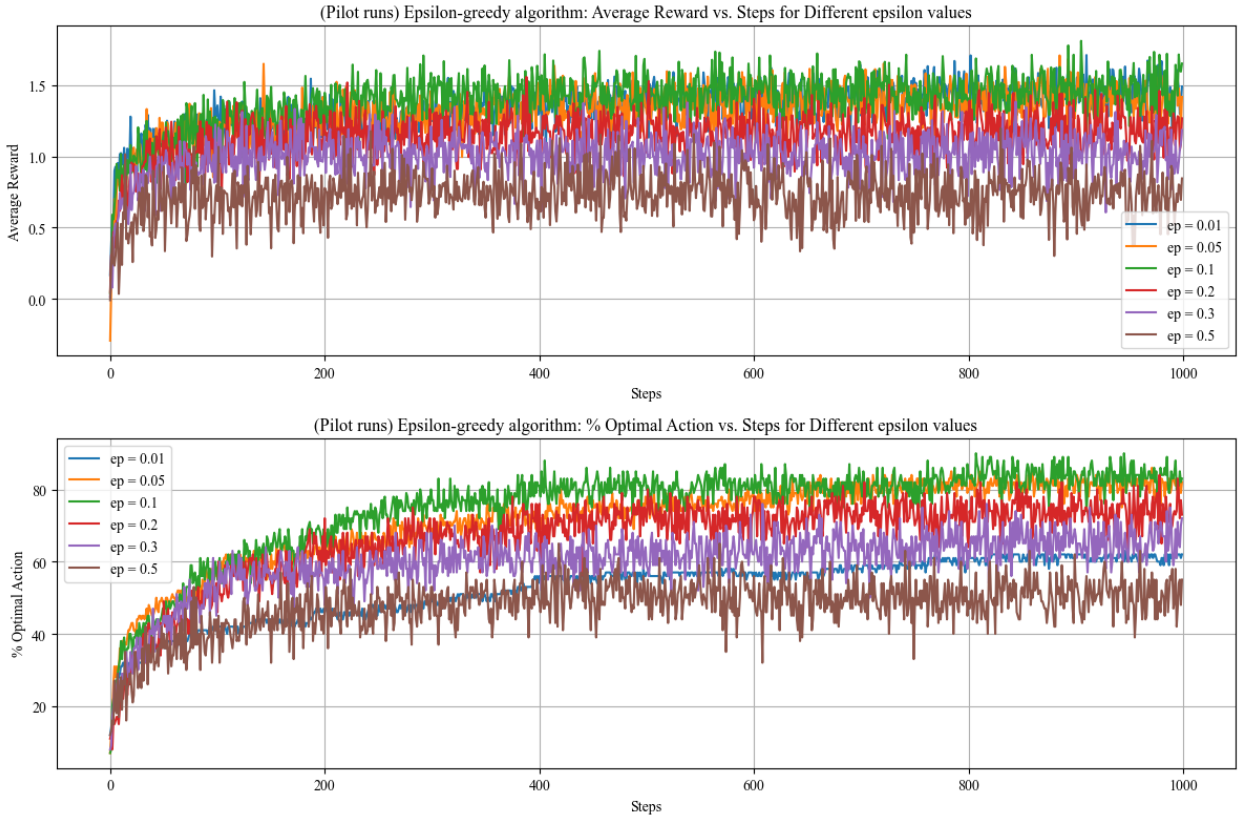

Fig. 2. *Epsilon-greedy method with different choices of epsilon (pilot runs)*

In Fig. 2, we use pilot runs with a preliminary experiment with the setting of 100 simulations and 1000 time steps for each problem. The values of epsilon consist of 0.01, 0.05, 0.1, 0.2, 0.3, 0.5. Fig. 2 shows that the epsilon-greedy method with epsilon = 0.1 (10% exploration) has the highest performance both in the average reward and the probability of choosing the optimal action. Additionally, the average reward of the methods with epsilon = 0.05 and 0.01 is also higher than those with epsilon =0.2, 0.3, and 0.5.

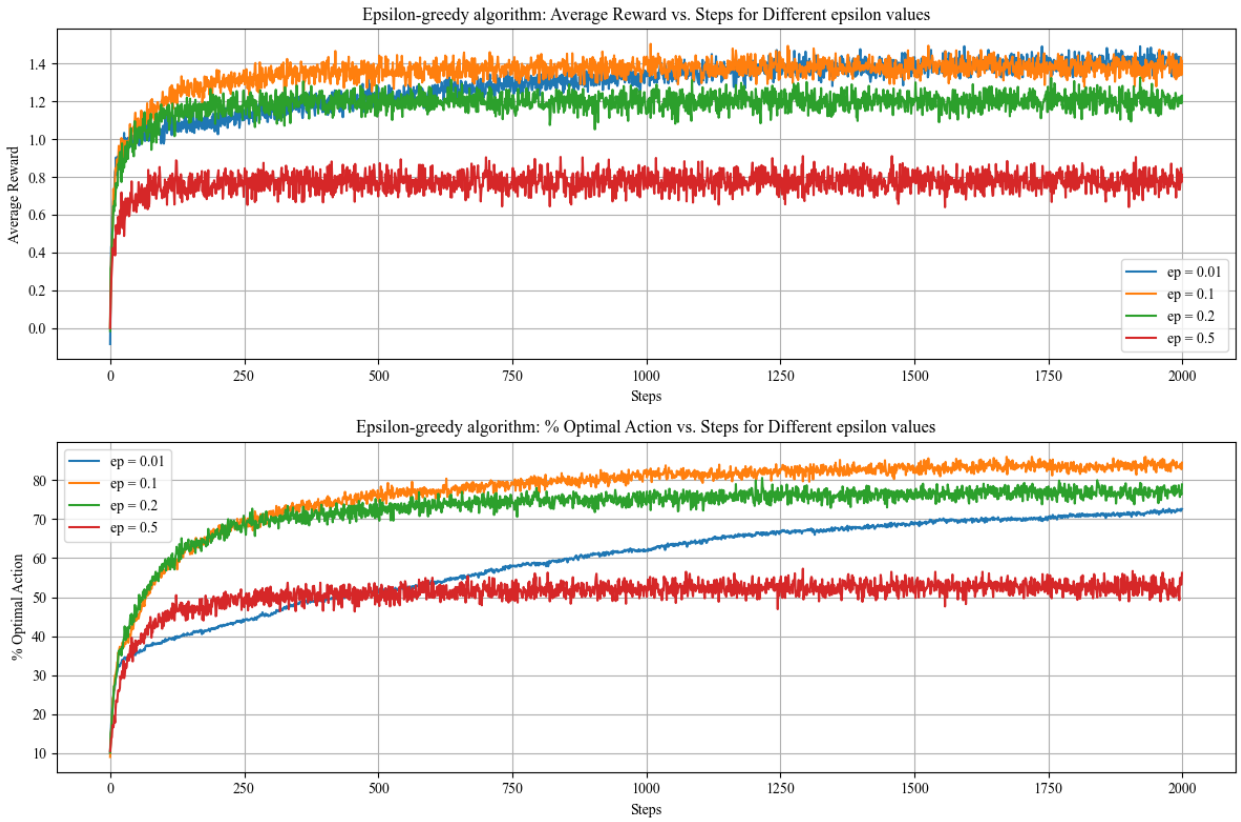

Fig. 3. *Epsilon-greedy method with different choices of epsilon (full runs)*

In Fig. 3, we investigate full runs with 1000 simulations of 2000 time steps. Due to the small size of the simulation, we choose the values of epsilon which have good performance in pilot runs in Fig. 2 (epsilon = 0.01 and 0.1) and others (epsilon = 0.2 and 0.5) for comparison. The methods of epsilon = 0.1 and epsilon = 0.01 have the same average reward at the convergence point (at around 1250 time steps), and converge after approximately 400 time steps, while the method of epsilon = 0.01 converges after around 1250 time steps. The highest epsilon value of 0.5 results in the lowest average reward due to excessive exploration. On the other hand, the epsilon-greedy method of epsilon = 0.2 has a higher probability of choosing the optimal action than the one of epsilon = 0.01, but has a lower average reward. The reason can be that the best actions from exploitation give more benefits than the best actions from exploitation. Overall, the epsilon-greedy method with epsilon of 0.1 provides the best performance, which balances exploration and exploitation.

3. Greedy method with optimistic starting values

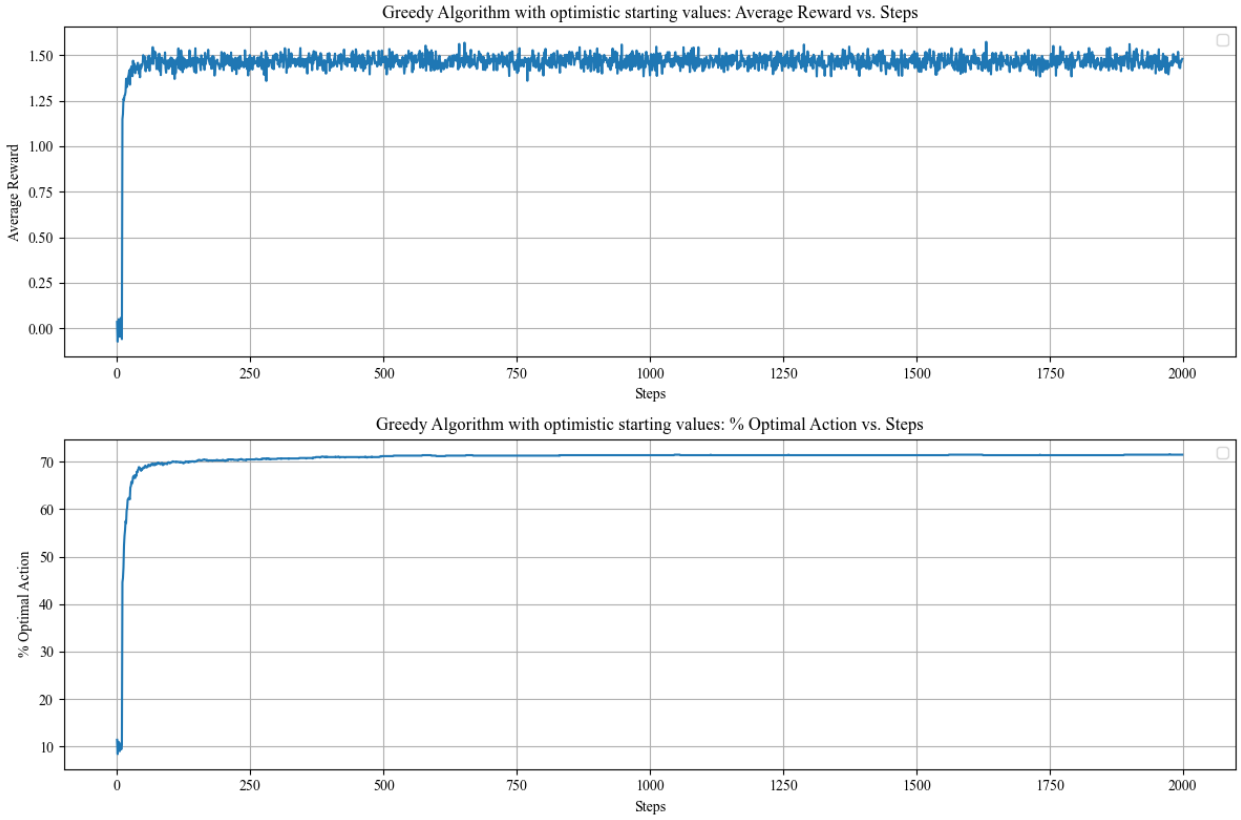

Fig. 4. *Greedy method with optimistic starting values*

In Fig. 4, we investigate the greedy method with optimistic starting action values to the 99.5th percentile of the normal distribution of the highest μ_i. The average reward rapidly reaches a stable point of around 1.5 after only 50 time steps. The probability of optimal action stabilizes around 71%. The greedy method with optimistic starting values has the same behavior as the greedy method in section 1, but higher performance with the reward of 1.5 compared to 1 of the non-optimistic greedy method. This indicates that it can give higher benefits if the exploration can be performed early at the beginning. Overall, the greedy method with optimistic starting values enhances the performance by operating exploration early and then exploiting the environment with a high percentage of optimal values.

4. Gradient bandit algorithm

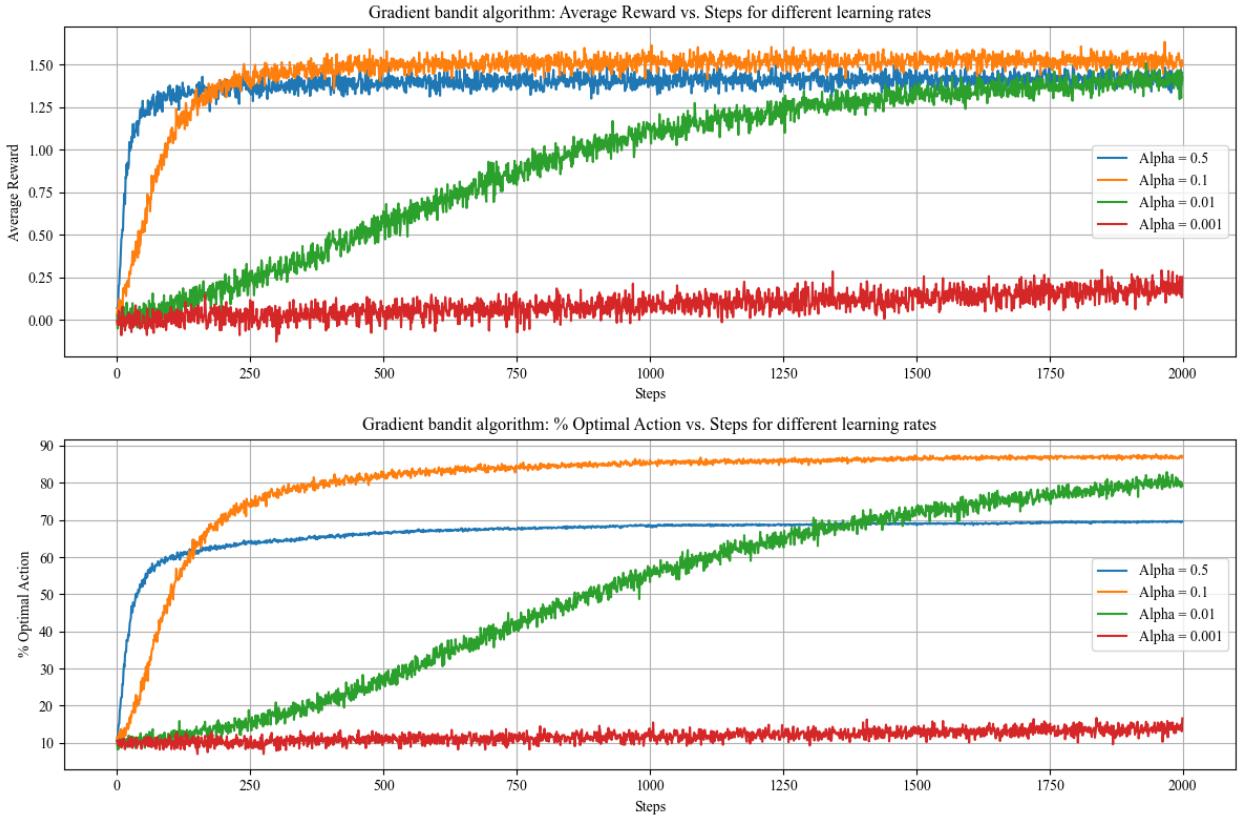

Fig. 5. *Gradient bandit algorithm with different learning rates*

In Fig. 5, we investigate the gradient bandit algorithm with different learning rates of 0.001, 0.01, 0.1, and 0.5. The figure shows that the gradient bandit algorithm with a learning rate of 0.1 has the best performance in terms of average reward and the highest probability of choosing the optimal action. It converges after around 500 time steps, while the one with alpha = 0.01 has a slower increase in average reward (i.e., around 2000 time steps to converge). In contrast, the method with a learning rate of 0.001 has minor improvement throughout the whole 200 time steps and has the lowest probability of choosing the optimal action. With a high learning rate of 0.5, the algorithm quickly converges after around 250 time steps; however, its exploitation is less efficient than that of a learning rate of 0.1, resulting in a lower average reward at the convergence point. Overall, in this case, the gradient bandit algorithm with a learning rate of 0.1 has the highest performance due to learn and selecting optimal actions more effectively.

5. Comparison of four bandit algorithms

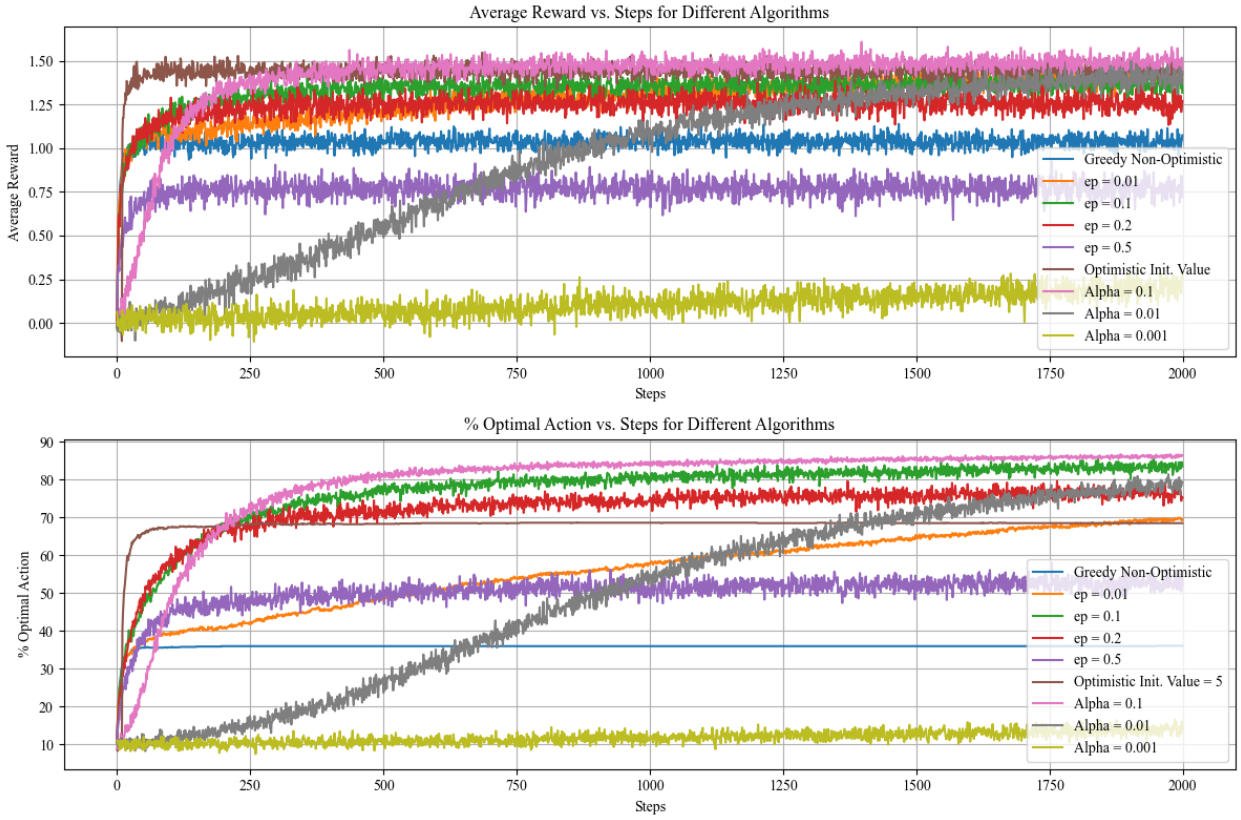

Fig. 6. *The comparison of bandit algorithms in stationary environments*

In this section, we compare bandit algorithms, including the greedy method with non-optimistic initial values, the epsilon-greedy method, the greedy method with optimistic starting values, and the gradient bandit algorithm. Fig. 6 shows that the greedy method with optimistic initial values and the gradient bandit algorithm with a learning rate of 0.1 achieve the highest average reward of around 1.5. The greedy method with optimistic initial values only needs fewer than 100 time steps to converge. This proves that early exploration can generate higher benefits compared to other methods.

PART 2:

NON-STATIONARY ENVIRONMENTS

In this part, we consider a bandit problem consisting of 10 arms with non-stationary environments that change the rewards gradually, abruptly, or a combination of both. There are four types of changes considered, including drift in means, mean-reverting change, permuting means, and a hard reset in all action values. All algorithms are run over 1000 simulations of 2000 time steps. In the simulations, we use the same 10 seeds to keep the same random processes happening in 10 arms.

1. Gradual changes: Drift

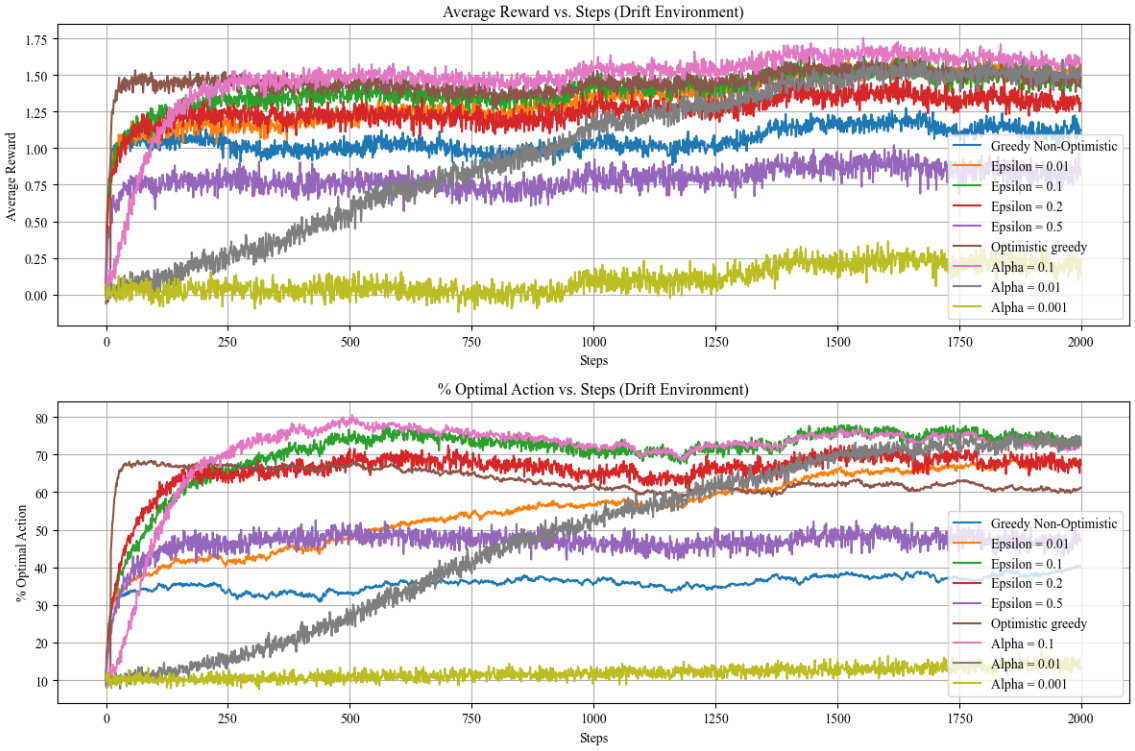

Fig. 7. *The comparison of bandit algorithms in drift environments (line plot)*

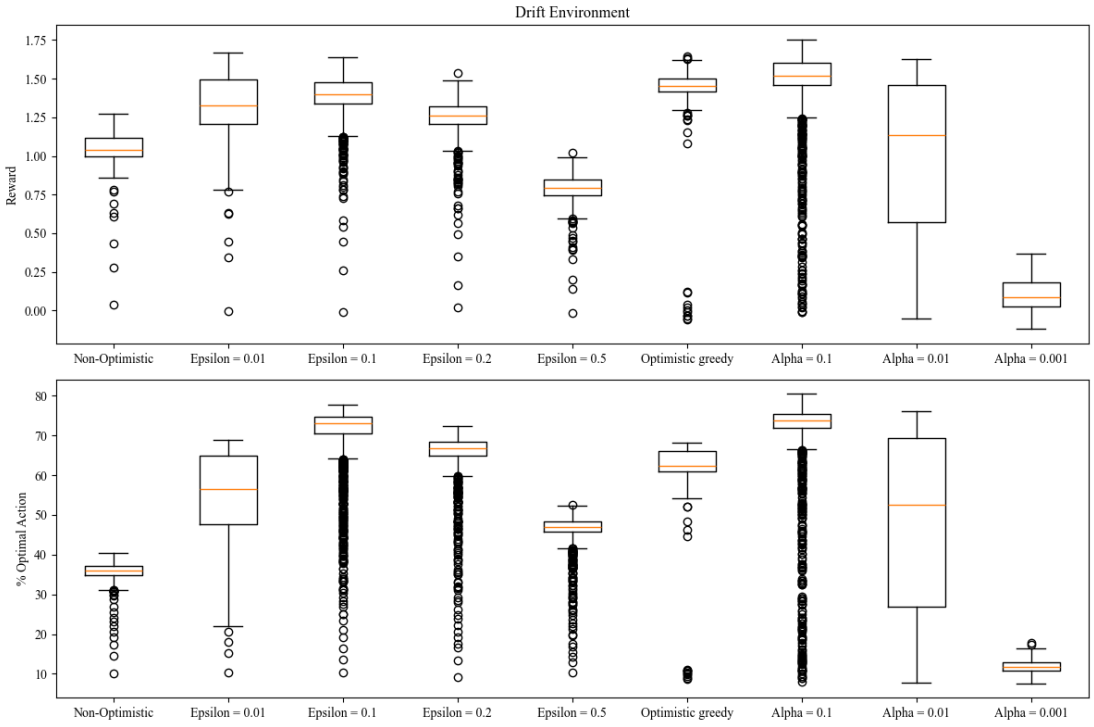

Fig. 8. *The comparison of bandit algorithms in drift environments (box plot)*

In this section, we compare bandit algorithms under a drift change environment as follows

$$\mu_{i,t} = \mu_{i,t-1} + \epsilon_{i,t}$$

where $\epsilon_{i,t}$ is iid N(0, 0.01^2). In general, there are small changes in the average rewards of all algorithms at the convergence state to adapt to changes in the environment, as shown in Fig. 7. The gradient bandit algorithm with a learning rate of 0.1 has the highest average rewards of around 1.5 after convergence. In Fig. 8, although it has several lower outliers with low rewards, the performance is still good in the long run due to high-reward actions (high median of 1.5 and small interquartile range). The performance of the greedy method with optimistic starting values and epsilon-greedy algorithms of epsilon = 0.01 and 0.1 is still good, with a small lower gap compared to that of the gradient bandit algorithm. In terms of stability, the greedy method with optimistic starting values rarely has worse actions, with some lower outliers shown in Fig. 8, while the average rewards are high. Overall, the gradient bandit algorithm and greedy method with optimistic starting values can tackle drift changes in environments efficiently.

2. Gradual changes: Mean-reverting change

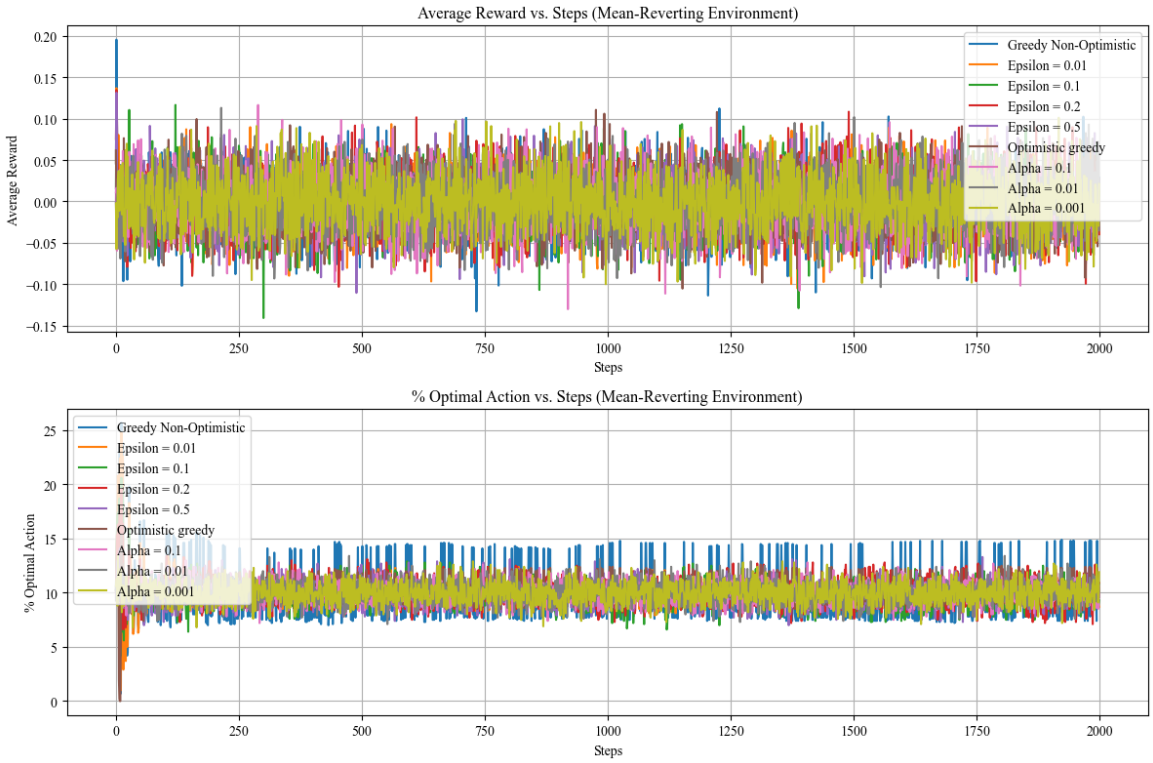

Fig. 9. *The comparison of bandit algorithms in mean-reverting environments (line plot)*

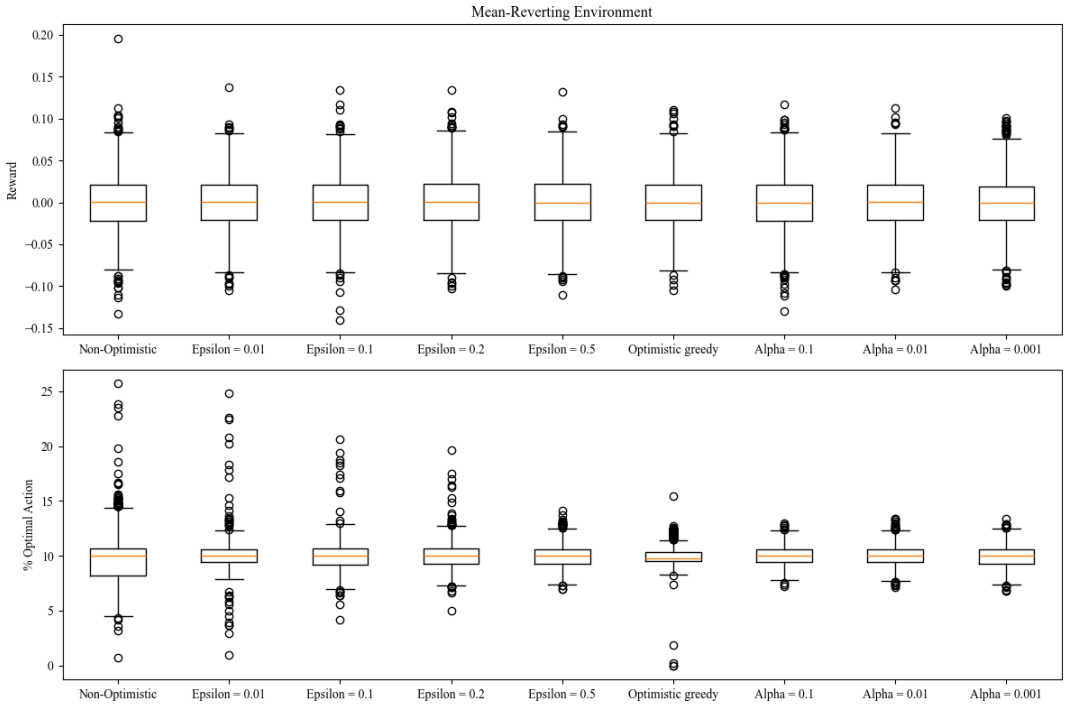

Fig. 10. *The comparison of bandit algorithms in mean-reverting environments (box plot)*

In this section, we compare bandit algorithms under a mean-reverting environment as the equation as follows.

$$\mu_{i,t} = k\mu_{i,t-1} + \epsilon_{i,t}$$

where k = 0.5 and $\epsilon_{i,t}$ is iid N(0, 0.01^2). All algorithms show the same median rewards of around 0 and median probabilities of choosing optimal actions. In this scenario, the environment changes quickly after every time step, which all bandit algorithms cannot adapt to.

3. Abrupt changes: Permuting the means

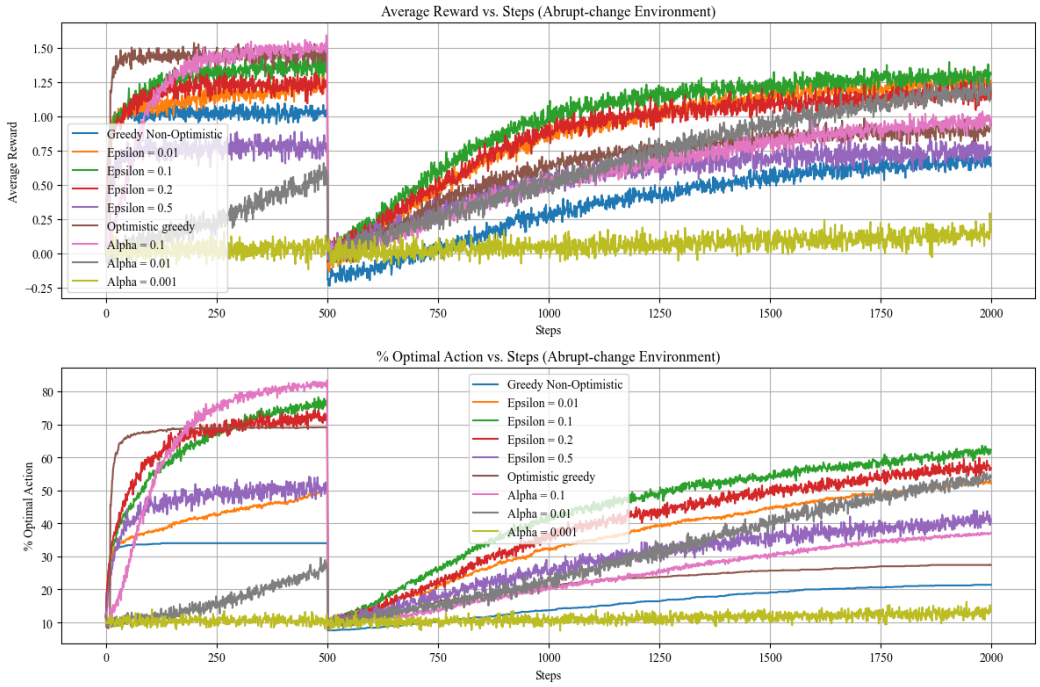

Fig. 11. *The comparison of bandit algorithms in abrupt-change environments (line plot)*

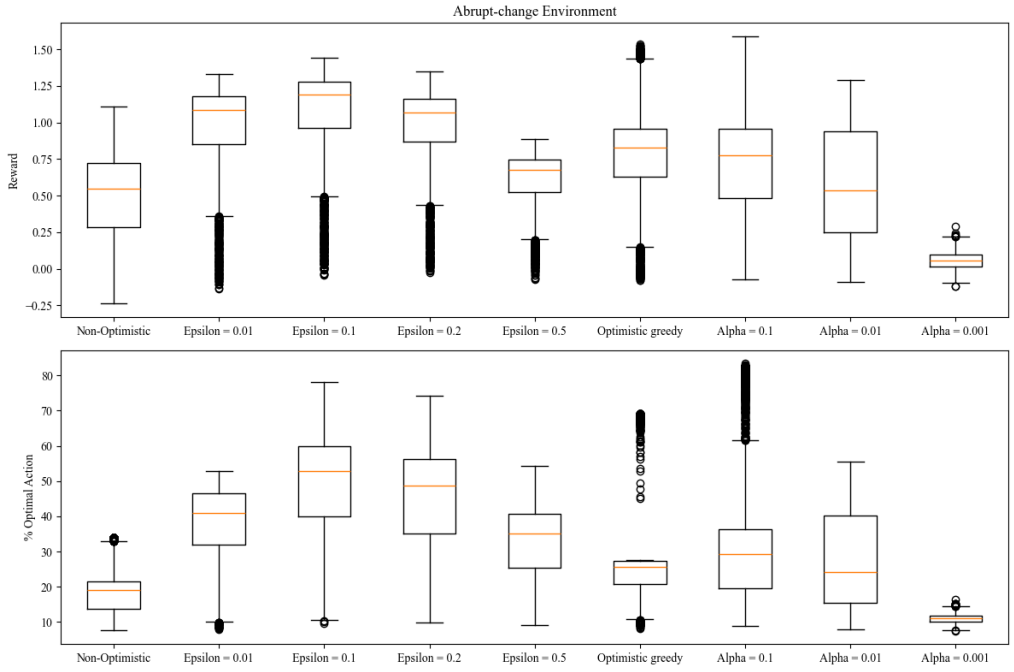

Fig. 12. *The comparison of bandit algorithms in abrupt-change environments (box plot)*

In this section, we assume that at t = 501, the means of reward distributions are permuted between arms. Fig. 11 shows the collapse in performance of all bandit algorithms after t = 501. The epsilon-greedy method of epsilon = 0.1 shows the efficiency in recovery from abrupt change with the highest average rewards. Meanwhile, although the gradient bandit algorithm of learning rate 0.1 has the highest rewards after convergence before an abrupt change, its performance cannot recover quickly and is outweighed by other epsilon-greedy methods of epsilon = 0.01 and 0.2. Overall, to deal with abrupt-change environments, epsilon-greedy methods with appropriate epsilon values should be chosen.

4. Abrup changes: Reseting all action values

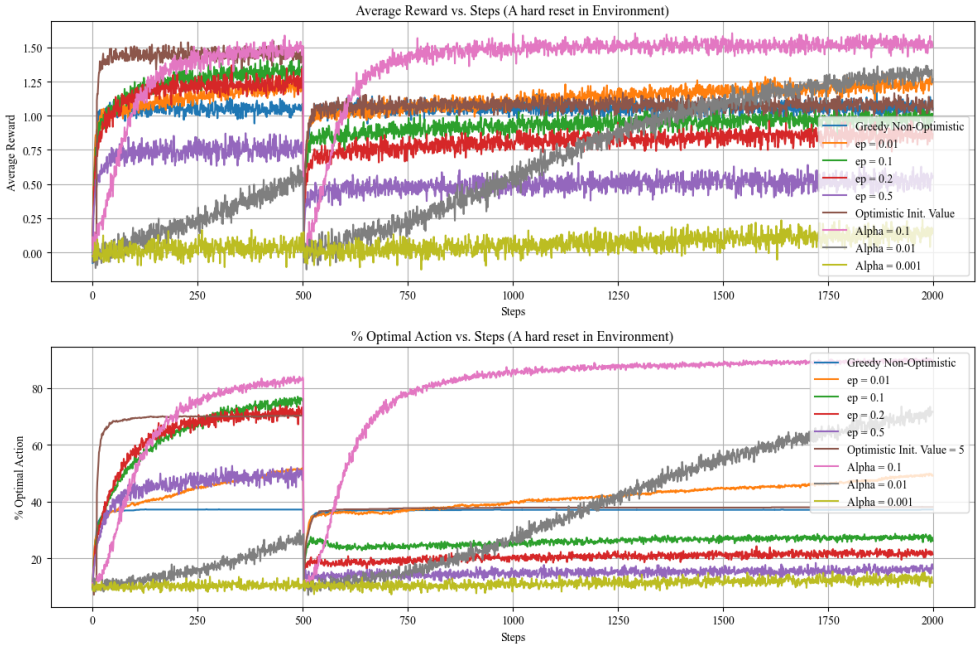

Fig. 13. *The comparison of bandit algorithms in non-stationary environments with a hard reset (line plot)*

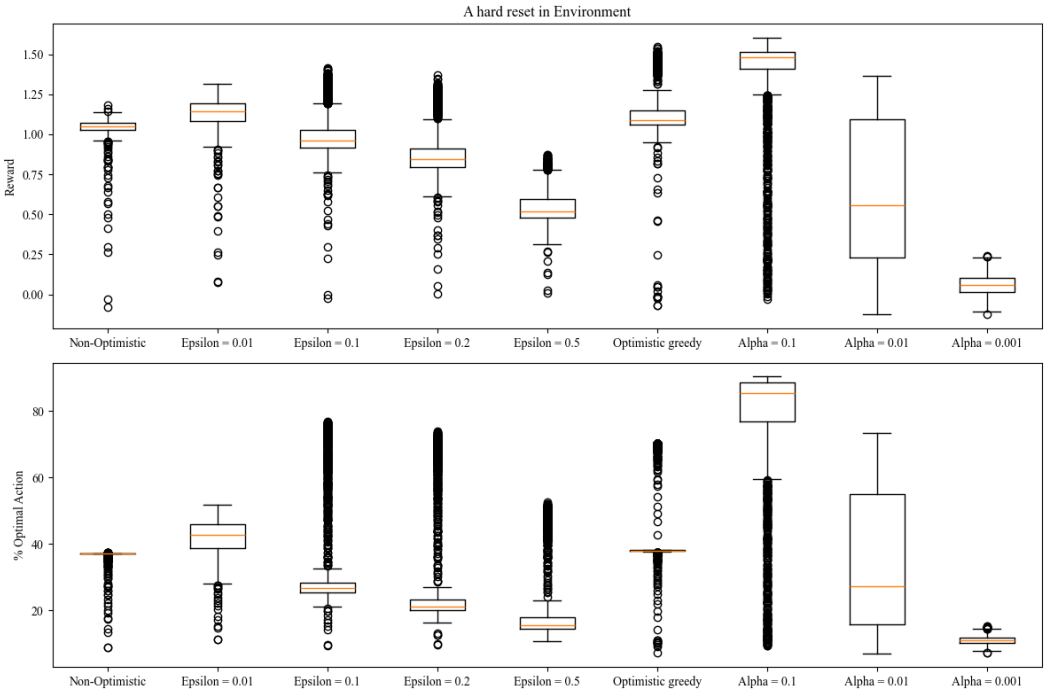

Fig. 14. *The comparison of bandit algorithms in non-stationary environments with a hard reset (box plot)*

In this section, we assume that at t = 501, all action values and preference values of all bandit algorithms are reset to 0. Fig. 13 shows the collapse in performance of all bandit algorithms after

t = 501. After the reset point, the gradient bandit algorithm with a learning rate of 0.1 recovers and has the highest rewards after around 300 time steps. Greedy methods with non-optimistic and optimistic initial values and the epsilon-greedy method with epsilon = 0.01 have good performance with the average reward of around 1.2 and quick recoveries after the hard reset; however, they cannot reach the performance achieved before the hard reset. Overall, to deal with environments with a hard reset, the gradient bandit algorithm with a learning rate of 0.1 should be chosen.