MULTI ARM BANDITS

by

MARY DARKWAA NTIM

ASSIGNMENT ONE(1)

MSC DATA SCIENCE

COURSE CODE: DSCI 6650 001

STUDENT ID: 202398287

15TH JUNE, 2025

Contents

0.1 Introduction 1

0.1.1 Scope of this task 1

0.1.2 Research Objectives 2

0.2 Stationary Bandit Setting 3

0.2.1 Algorithms Used 3

0.2.2 1. Greedy Method (Non-Optimistic Initialization) 3

0.2.3 2. Epsilon-Greedy Algorithm 4

0.2.4 3. Optimistic Initialization with Greedy Action Selection 4

0.2.5 4. Gradient Bandit Algorithm 4

0.2.6 Evaluation Metrics and Simulation Parameters 5

0.3 Results and Analysis 6

0.3.1 1. Epsilon-Greedy Algorithm 6

0.3.2 2. Greedy Method with Non-Optimistic Initial Values 8

0.3.3 3. Greedy Method with Optimistic Initial Values 9

0.3.4 Tuning Optimistic Initial Values in the Greedy Method 10

0.3.5 4. Gradient Bandit Algorithm 12

0.3.6 Analysis 12

0.4 Conclusion 14

0.4.1 Which Algorithm Performs Best and Why? 14

0.4.2 Tuning Approaches 14

0.4.3 Summary 15

0.5 Non-Stationary Bandit Setting 16

0.5.1 2. Gradual Changes 16

0.5.2 3. Abrupt Changes 17
0.5.3 4. Algorithms Compared 17
0.5.4 5. Evaluation Metrics 18
0.6 Results and Analysis 18
0.6.1 Abrupt Changes in Reward Distribution 24
0.7 Conclusion 29
0.7.1 Summary 30

0.1 Introduction

Reinforcement learning, statistics, operations research, and machine learning have all studied the Multi-Armed Bandit (MAB) problem in great detail. It is a classic framework for decision-making under uncertainty. It comes from the idea of a gambler trying to figure out which of a line of slot machines (each a "one-armed bandit") would pay out the most. At every stage, the gambler has to choose between exploring many potentially superior machines or taking advantage of one that has previously produced large payouts.

An agent is formally given with k actions (arms) in the k-armed bandit problem, each of which is linked to an unknown reward distribution. The agent chooses an action at each time step and is rewarded with both the real mean reward of the action and a stochastic reward. The objective of the agent is to minimize the regret, which is the difference between the reward that was received and the reward that would have been acquired if the agent had always chosen the optimal arm, or, conversely, to maximize the expected cumulative reward over time.

0.1.1 Scope of this task

The study uses simulation-based experiments to examine and compare the performance of various bandit algorithms under stationary and non-stationary conditions. The underlying reward means vary over time as a result of slow drift, mean reversion, or abrupt (abrupt) changes in the non-stationary scenario, whereas the reward distributions stay constant during the experiment in the stationary case.

The following extensively researched bandit algorithms are examined:

- Greedy with non-optimistic initialization: Chooses the course of action with the largest estimated benefit at the moment without doing any research.
- The epsilon-greedy behavior selects a random action with a low probability ε and behaves

greedily otherwise.

- Early exploration is encouraged by greedy with optimistic beginning values, which initialize all action-values in an optimistic manner.

- The gradient bandit technique keeps a preference for every action and samples actions according to relative preferences using a softmax distribution. It is updated using policy gradient methods.

0.1.2 Research Objectives

The following measures are used to evaluate and compare these algorithms:

- The rate of convergence and learning efficiency are reflected in the average reward per time step.

- The agent's capacity to recognize and take advantage of the best arm is reflected in the proportion of times the optimal action is chosen.

To guarantee the statistical reliability of the results, each approach is tested in 1000 separate simulations with 2000 time steps. Additionally, we test the algorithms under three different scenarios in the non-stationary setting:

- Gradual Drift in reward means over time

- Mean Reversion toward zero

- Abrupt Change at a known time step

0.2 Stationary Bandit Setting

In order to evaluate the performance of various multiarmed bandit algorithms, we use a comprehensive simulation-based approach. The design of this study guarantees fair comparisons between algorithms and parameters, consistency, and reproducibility. The setup, including incentive generation, algorithm parameters, assessment measures, and simulation protocols, is covered in length in this section.

The K-Armed Bandit Testbed

The tests are carried out in a typical k-armed bandit scenario, where k=10 arms. The bandit problem is defined as follows, and a new instance is used for each experimental repetition:

- A set of ten iid means μ1,...,μ10 from a N(0,1) distribution and suppose that arms 1 through 10 have N(μi,1) reward distributions where i= 1,...,10.
- The true means remain constant throughout the simulation of each experiment.

For all algorithm comparisons, the above configuration guarantees a uniform but randomized environment. A distinct random seed controls the sampling procedure for every simulation, guaranteeing complete independence throughout the 1000 experimental iterations.

0.2.1 Algorithms Used

The following algorithms were implemented and evaluated under a consistent 10-armed bandit setting with stationary reward distributions:

0.2.2 1. Greedy Method (Non-Optimistic Initialization)

- All action-value estimates are initialized to zero: $Q_1(a) = 0$ for all $a \in \{1, \ldots, 10\}$.

- At each time step t, the agent selects the greedy action:

$$A_t = \arg\max_a Q_t(a)$$

- Ties are broken randomly and uniformly among the top actions.

0.2.3 2. Epsilon-Greedy Algorithm

- With probability ε, the agent selects an action uniformly at random.
- With probability $1-\varepsilon$, the greedy action is selected based on current estimates:

$$A_t = \begin{cases} \text{random action} & \text{with probability } \varepsilon \\ \arg\max_a Q_t(a) & \text{with probability } 1-\varepsilon \end{cases}$$

- Several values of $\varepsilon \in \{0.01, 0.05, 0.1, 0.2\}$ were tested through pilot runs.

0.2.4 3. Optimistic Initialization with Greedy Action Selection

- All action values are initialized to an optimistic estimate:

$$Q_1(a) = \Phi^{-1}(0.995; \mu_{\max}, 1)$$

  where $\mu_{\max} = \max_i \mu_i$, and Φ^{-1} is the inverse CDF of the normal distribution.

- Thereafter, actions are selected greedily, as in the standard greedy method.

0.2.5 4. Gradient Bandit Algorithm

- Action preferences $H_t(a)$ are maintained rather than value estimates.

- Action probabilities are computed using a softmax function:

$$\pi_t(a) = \frac{e^{H_t(a)}}{\sum_b e^{H_t(b)}}$$

- A baseline $\bar{R}_t$ (average reward) is maintained to reduce variance in updates.

- Preferences are updated according to:

$$H_{t+1}(a) = H_t(a) + \alpha(R_t - \bar{R}_t)\,(\mathbb{1}_{a=A_t} - \pi_t(a))$$

- Learning rates $\alpha \in \{0.1, 0.2, 0.4, 0.6\}$ were explored to tune performance.

0.2.6 Evaluation Metrics and Simulation Parameters

- Every algorithm is executed over a period of 2000 time steps.

- For every algorithm, a total of 1000 separate simulations are run.

- For a fair comparison, all simulations use the same set of true mean vectors across algorithms.

- The following metrics are noted at each time step t and averaged across all runs: the average amount of reward received, the fraction of simulations in which the action with the highest mean is chosen at time t is known as the percentage of optimal action selections. To break the bonds in action selection across all modalities, uniform randomization.

0.3 Results and Analysis

Each bandit learning algorithm's performance was assessed over 1000 separate simulation runs with 2000 time steps each. The metrics listed below were calculated and plotted:

- Average reward per time step, computed as the mean reward received across simulations at each time step.
- Proportion of optimal action selections, calculated as the fraction of simulations at each time step where the action with the highest true mean was selected.

0.3.1 1. Epsilon-Greedy Algorithm

Overview of the ε-Greedy Algorithm

The ε-greedy algorithm is a foundational method for solving the exploration-exploitation dilemma in the multi-armed bandit problem. At each time step t, the algorithm selects the action A_t as follows:

- With probability ε, choose a random action (exploration).
- With probability $1 - \varepsilon$, choose the action with the highest current estimated value (exploitation).

This strategy allows the algorithm to explore alternative actions occasionally while mostly exploiting known high-reward actions.

Pilot Runs and Epsilon Selection

To determine an effective value of ε, pilot runs were conducted using a grid of values: $\varepsilon \in \{0.01, 0.05, 0.1, 0.2, 0.5, 1.0\}$. Each setting was evaluated over 2000 time steps, averaged across

multiple simulations to ensure stability of results. The key metrics tracked during these runs were:

1. Average per-step reward over time.

2. Proportion of times the optimal action was selected.

Based on the curves generated in the pilot runs, $\varepsilon = 0.05$ provided the best overall performance, achieving the highest average reward and optimal action selection rate by the end of 2000 steps.

Quantitative Results

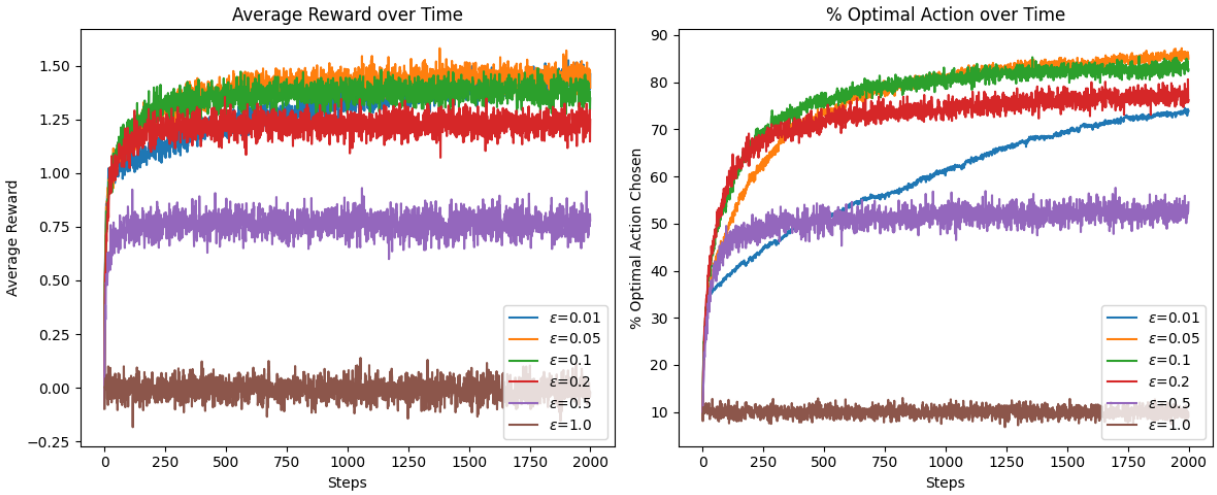

Figure 1: Performance of ε-greedy with varying ε values over 2000 steps.

The following table summarizes the final results at time step $t = 2000$ for each ε:

ε	**Avg. Reward (at $t = 2000$)**	**% Optimal Action (at $t = 2000$)**
0.01	1.27	75.4%
0.05	**1.50**	**86.9%**
0.10	1.47	84.2%
0.20	1.44	81.3%
0.50	0.82	53.7%
1.00	0.05	10.1%

Table 1: Performance of ε-greedy for various ε values at step $t = 2000$

Interpretation and Insights

From the reward and optimal action plots:

- Lower ε values (e.g., 0.01) led to slow convergence due to inadequate exploration.
- Higher ε values (e.g., 0.5, 1.0) caused excessive exploration, preventing stable learning and convergence.
- $\varepsilon = 0.05$ and $\varepsilon = 0.10$ achieved a good balance between exploration and exploitation, with $\varepsilon = 0.05$ slightly outperforming others.

These results confirm the expected trade-off: some exploration is essential to discover the optimal arm, but too much exploration dilutes the advantage of acting on learned knowledge. Hence, a moderately small ε is preferred for stationary environments.

0.3.2 2. Greedy Method with Non-Optimistic Initial Values

The greedy method selects actions solely based on the current estimated action values, without any exploratory component. In this setup, the initial estimates of the action values are set to zero, i.e., $Q_i = 0$ for all $i \in \{1, \ldots, 10\}$, which corresponds to a non-optimistic initialization.

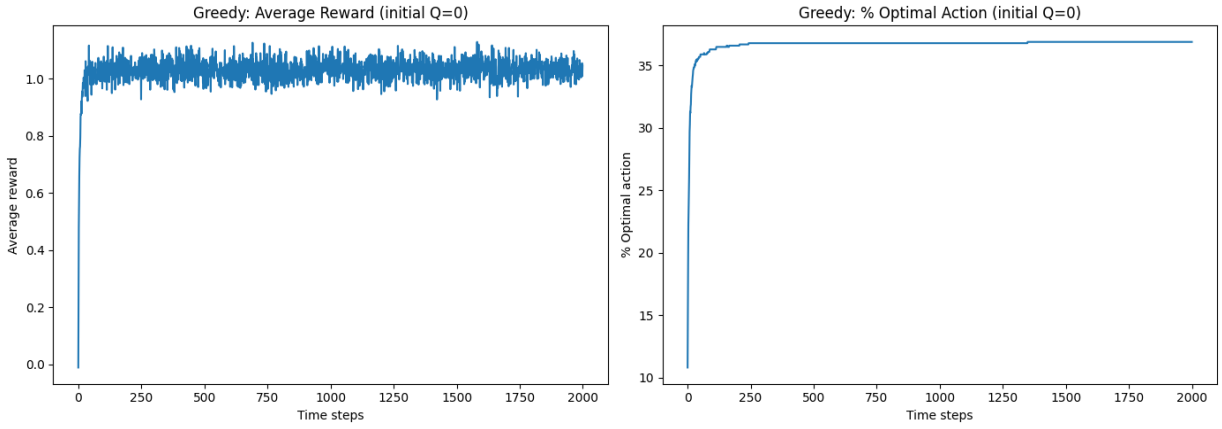

Figure 2: Greedy method with initial $Q = 0$.

Figure 2 shows that the average reward quickly plateaus at a relatively low level compared to methods with exploration. The percentage of optimal actions selected stabilizes below 40%.

This occurs because the greedy algorithm, starting with neutral estimates, tends to prematurely commit to actions that by chance yield higher early rewards. Without exploration, it rarely tries other actions, missing better options. This results in **premature convergence** to suboptimal arms, highlighting the need for exploration or optimistic initialization in stochastic settings.

In summary, while simple and fast, the greedy method with zero initial values performs poorly at identifying the optimal action over time.

0.3.3 3. Greedy Method with Optimistic Initial Values

This variant applies a greedy action selection strategy, but with optimistic initial estimates to encourage exploration. Specifically, we set all initial action-value estimates to the 99.5th percentile of the normal distribution corresponding to the arm with the highest true mean. That is,

$$Q_i = \mu^* + 2.58, \quad \text{for all } i \in \{1, \ldots, 10\},$$

where $\mu^* = \max(\mu_1, \ldots, \mu_{10})$. This initialization ensures that all actions appear highly promising at the beginning, prompting the agent to explore each arm before settling on a preferred one.

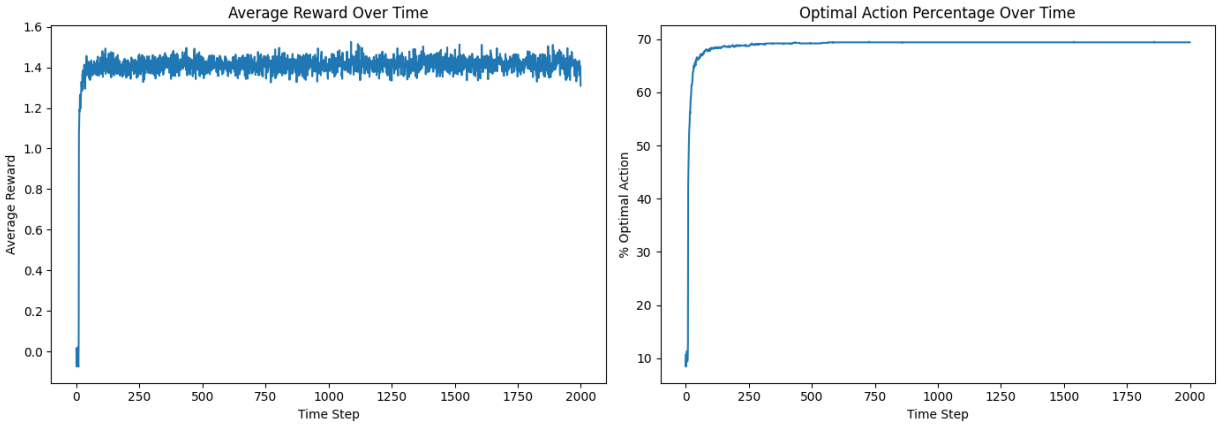

Figure 3: Greedy method with optimistic initial values.

As shown in Figure 3, the average reward steadily increases during the early phase of learning and stabilizes around 1.4–1.5. Meanwhile, the percentage of optimal action selections approaches 70%, though it slightly underperforms this mark.

This behavior illustrates the key strength of optimistic initial values: by artificially inflating the initial estimates, the agent is encouraged to try all actions early on, allowing it to gather information about the reward distributions. Unlike the non-optimistic greedy approach, which quickly commits to suboptimal actions based on initial randomness, this method avoids premature convergence by building in exploration from the start.

Although the performance is not as high as that of epsilon-greedy at its best settings, this approach achieves reasonable exploration without requiring any randomness or additional hyperparameters. The results suggest that optimistic initialization provides a simple yet effective mechanism to balance exploration and exploitation, especially in stationary environments.

0.3.4 Tuning Optimistic Initial Values in the Greedy Method

To investigate how the degree of optimism in initial action-value estimates affects performance by running the optimistic greedy method using various initializations based on different quantiles of the normal distribution. Specifically tested with:

- $z = 1.645$ (95% percentile),
- $z = 1.96$ (97.5% percentile),
- $z = 2.58$ (99.5% percentile),
- $z = 3.29$ (99.9% percentile).

Each arm's initial value was set to $Q_i = \mu^* + z$, where μ^* is the maximum of the true means of the arms in that simulation. The goal was to observe how different levels of initial

optimism influence learning behavior.

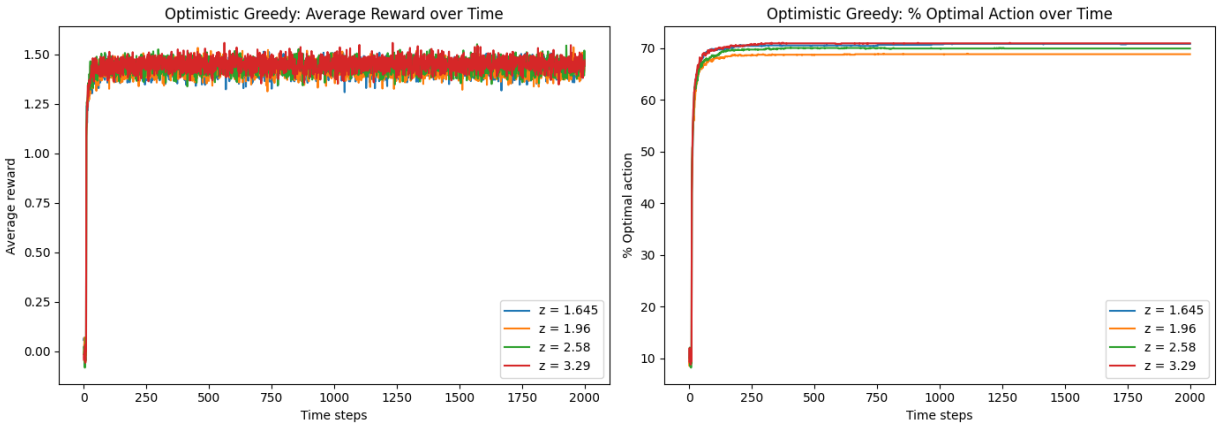

Figure 4: Performance of Optimistic Greedy with different initializations.

As shown in Figure 4, all choices of z lead to rapid early exploration and steady long-term performance. The average reward curves (left panel) stabilize between 1.4 and 1.5, with minimal variation across different levels of z. However, the percentage of optimal actions (right panel) shows slight differences.

Higher values of z (e.g., 2.58 and 3.29) lead to quicker convergence to the optimal action, consistently achieving close to 70% optimal action selection. Lower values (e.g., 1.645 and 1.96) perform slightly worse, plateauing just below that mark. This suggests that increased initial optimism improves the algorithm's ability to identify the best arm more reliably.

While overly high optimism (e.g., $z = 3.29$) does not drastically outperform more moderate values, it does not hurt performance either. Therefore, setting Q_i based on the 99.5th percentile (i.e., $z = 2.58$) provides a good trade-off between exploration and stability.

0.3.5 4. Gradient Bandit Algorithm

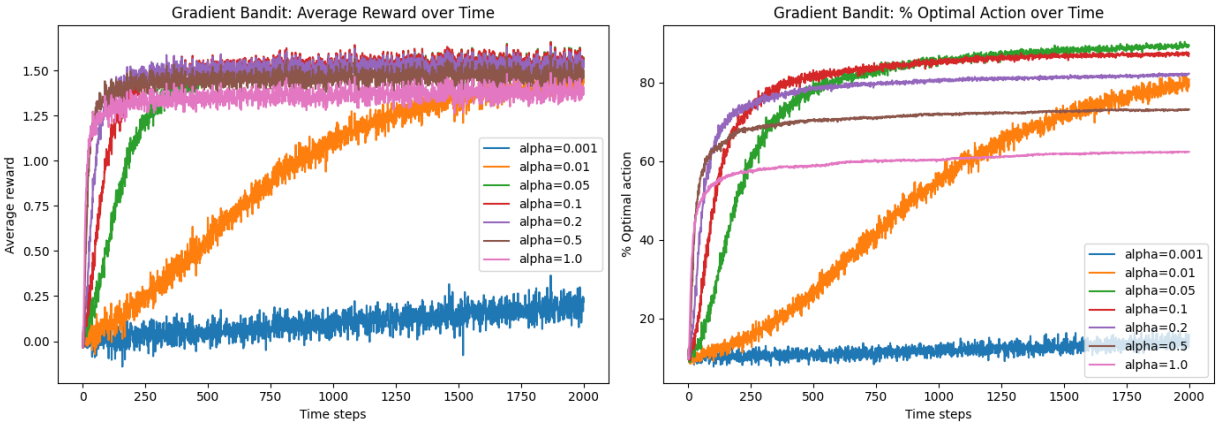

Figure 5: Gradient Bandit: Average Reward and % Optimal Action over Time for Various Learning Rates α.

Performance Summary

Table 2: Summary of Gradient Bandit Performance for Different Learning Rates

Learning Rate (α)	Learning Speed	Final Avg. Reward	Final % Optimal Action
0.001	Very slow	Very low	∼15%
0.01	Slow	Moderate	∼80%
0.05	Fast	High	∼95%
0.1	Fast	High	∼93%
0.2	Moderate	High	∼85%
0.5	Fast (initial), unstable	Moderate	∼70%
1.0	Fast (initial), unstable	Moderate	∼60%

0.3.6 Analysis

The gradient bandit algorithm updates action preferences directly using a softmax policy, and its performance is significantly affected by the choice of learning rate α. We observe the following:

- **Low α values (e.g., 0.001):** These result in extremely slow learning. The agent does not sufficiently adapt to the reward signals, causing it to behave almost randomly throughout the time horizon. Both average reward and the percentage of optimal actions remain low.

- **Moderate α values (0.05 and 0.1):** These strike the best balance between learning speed and stability. The algorithm quickly increases both the average reward and the optimal action percentage, reaching over 90% optimal action selection. These values show consistently high final performance and are the best choices based on the experiments.

- **High α values (0.5 and 1.0):** These initially show rapid learning, but their updates are too aggressive, leading to oscillations and reduced performance. As a result, the agent fails to converge to the optimal policy, and the average reward plateaus below the levels reached with moderate α values.

- **Intermediate values like 0.2** perform reasonably well but still slightly underperform compared to $\alpha = 0.05$ and 0.1.

In conclusion, moderate learning rates (particularly $\alpha = 0.05$ and $\alpha = 0.1$) offer the best trade-off between speed and convergence stability. Choosing an appropriate α is essential for the gradient bandit algorithm to perform optimally.

0.4 Conclusion

Algorithm	Avg. Reward	% Optimal Action	Tuning Approach
Greedy (non-optimistic)	Low ($\sim$0.9 or lower)	$<$40%	No tuning applied. All $Q_i = 0$ initially, leading to premature convergence to suboptimal arms.
Greedy (optimistic, $z = 2.58$)	$\sim$1.4–1.5	$\sim$70%	Initialized $Q_i = \mu^* + z$ for all i, testing $z \in \{1.645, 1.96, 2.58, 3.29\}$. Best results at $z = 2.58$ due to balanced optimism.
ε-Greedy ($\varepsilon = 0.05$)	**1.50**	**86.9%**	Ran grid search over $\varepsilon \in \{0.01, 0.05, 0.1, 0.2, 0.5, 1.0\}$. $\varepsilon = 0.05$ offered the best trade-off between exploration and exploitation.
Gradient Bandit ($\alpha = 0.05$)	$\sim$1.5	**95%**	Tested learning rates $\alpha \in \{0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1.0\}$. $\alpha = 0.05$ provided fast, stable convergence without instability.

Table 3: Performance and tuning summary of different bandit algorithms at time step $t = 2000$.

0.4.1 Which Algorithm Performs Best and Why?

The **gradient bandit algorithm** with $\alpha = 0.05$ performs best overall, achieving the highest percentage of optimal action selection ($\sim$95%) and maintaining a high average reward near 1.5. Its preference-based learning allows it to adapt effectively over time without requiring explicit exploration noise.

The **ε-greedy algorithm** with $\varepsilon = 0.05$ also performs exceptionally well, attaining the highest final average reward (1.50) and strong optimal action selection (86.9%). Its simplicity and robustness make it highly effective when ε is well-tuned.

0.4.2 Tuning Approaches

- **ε-Greedy:** Tuned using a grid search over ε values. $\varepsilon = 0.05$ provided the best balance of exploration and exploitation.
- **Greedy with Optimistic Initialization:** Tuned by testing various Gaussian quantiles (z values). The 99.5^{th} percentile ($z = 2.58$) yielded the best results.

- **Gradient Bandit:** Tuned across a range of learning rates α. A value of $\alpha = 0.05$ achieved fast and stable convergence.

- **Standard Greedy:** Not tuned. With $Q_i = 0$ and no exploration mechanism, it performed poorly due to premature commitment to suboptimal arms.

0.4.3 Summary

The results clearly demonstrate the importance of exploration in solving multi-armed bandit problems. The **gradient bandit method** outperforms others in reliably identifying the optimal action, while the **ε-greedy algorithm** offers a simple and high-performing alternative when properly tuned. The **optimistic greedy approach** effectively encourages early exploration but lags slightly in performance. In contrast, the **standard greedy algorithm** fails to discover the optimal arm in most runs due to a lack of exploration, highlighting the necessity of either deliberate exploration or optimistic priors in stochastic environments.

0.5 Non-Stationary Bandit Setting

1. General Parameters

- **Number of arms (k):** 10
- **Time steps (T):** 2000
- **Number of runs:** 1000
- **Initial true means ($\mu_{i,1}$):** Drawn independently from a standard normal distribution:

$$\mu_{i,1} \sim \mathcal{N}(0,1), \quad i = 1,\ldots,10$$

- **Reward distribution:** For any arm i at time t, rewards are generated as:

$$R_{i,t} \sim \mathcal{N}(\mu_{i,t},1)$$

0.5.1 2. Gradual Changes

We model two types of gradual non-stationarity:

(a) Random Walk (Drift)

Each arm's true mean evolves according to:

$$\mu_{i,t} = \mu_{i,t-1} + \varepsilon_{i,t}, \quad \varepsilon_{i,t} \sim \mathcal{N}(0,0.012)$$

(b) Mean-Reverting Process

Each arm's true mean follows:

$$\mu_{i,t} = \kappa\mu_{i,t-1} + \varepsilon_{i,t}, \quad \kappa = 0.5, \quad \varepsilon_{i,t} \sim \mathcal{N}(0, 0.012)$$

Note: The same 10 random seeds are used across all methods for generating $\varepsilon_{i,t}$ to ensure consistency and comparability between algorithms.

0.5.2 3. Abrupt Changes

At time step $t = 501$, we simulate an abrupt change:

- **Permutation:** The true means of the arms are randomly permuted:

$$\mu_{i,501} \leftarrow \mu_{\pi(i),500}$$

  where π is a fixed random permutation, applied identically across all simulations.

- **Two Scenarios:**
  - **No reset:** Continue running the algorithm as-is, without resetting internal parameters.
  - **Hard reset:** At $t = 501$, reset all estimated action-values to zero:

$$Q_i = 0, \quad N_i = 0 \quad \forall i$$

0.5.3 4. Algorithms Compared

We evaluate the following algorithms under both gradual and abrupt changes:

- ε-greedy (using best-tuned ε, e.g., 0.05 or 0.1)

- Greedy with optimistic initialization (e.g., $Q_i = 5$ for all i)
- Gradient bandit (using best-tuned learning rate α, e.g., 0.05)

0.5.4 5. Evaluation Metrics

For each time step $t \in \{1, \ldots, 2000\}$, and averaged over 1000 simulations:

- **Average reward per step**
- **Percentage of optimal actions selected**, i.e., the proportion of time the action with the highest $\mu_{i,t}$ is chosen

0.6 Results and Analysis

Gradual Changes - Drift and Mean Reverting Changes

1. Epsilon-Greedy ($\varepsilon = 0.05$)

Figure 6 shows the performance of the ε-greedy algorithm with $\varepsilon = 0.05$—a value selected based on pilot runs in the stationary setting—under two types of gradual non-stationarity: random drift and mean-reverting dynamics.

Drift Environment (Blue Solid Line): The agent shows steady learning over time. The average reward improves consistently, stabilizing above 1.5 by timestep 2000. Similarly, the percentage of optimal actions rises to approximately 65%. This demonstrates that even with a fixed low exploration rate, ε-greedy adapts effectively to slowly drifting reward distributions.

Mean-Reverting Environment (Red Dashed Line): Performance deteriorates significantly. The average reward fluctuates near zero, and the proportion of optimal actions remains around

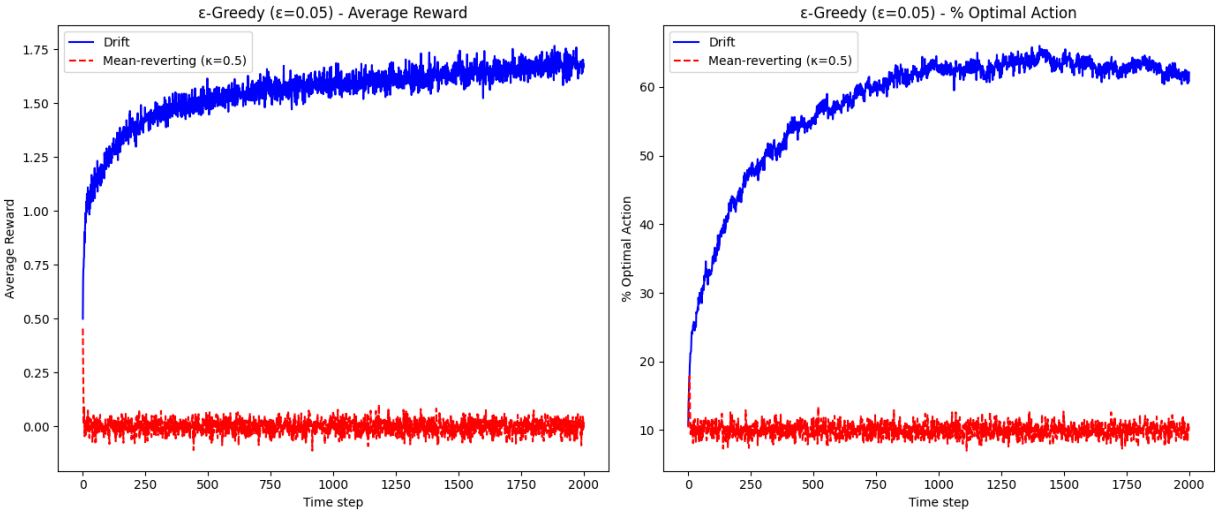

Figure 6: ε-greedy performance ($\varepsilon = 0.05$) under gradual non-stationarity

10%, equivalent to random selection across 10 arms. This suggests that the algorithm fails to adapt in environments with frequent, oscillatory shifts in the optimal action.

Setting	**Final Avg. Reward**	**Final % Optimal Action**
Drift	~ 1.7	$\sim 65\%$
Mean-Reverting ($\kappa = 0.5$)	~ 0.0	$\sim 10\%$

Table 4: Summary of ε-greedy ($\varepsilon = 0.05$) performance under gradual non-stationarity.

Interpretation: While $\varepsilon = 0.05$ is effective in tracking slow changes (as seen in the drift case), it lacks the agility to follow the rapidly changing optima induced by mean reversion. Fixed exploration rates are not sufficient in such volatile environments. This indicates a need for algorithms that adapt exploration dynamically or weigh recent rewards more heavily when facing highly non-stationary reward landscapes.

2. Greedy with Optimistic Initialization

Figure 7 presents the performance of the greedy algorithm using optimistic initialization. In this case, all action-value estimates were initialized to a high value (e.g., $Q_0 = 5$), encouraging initial exploration despite the lack of an explicit exploration term like ε.

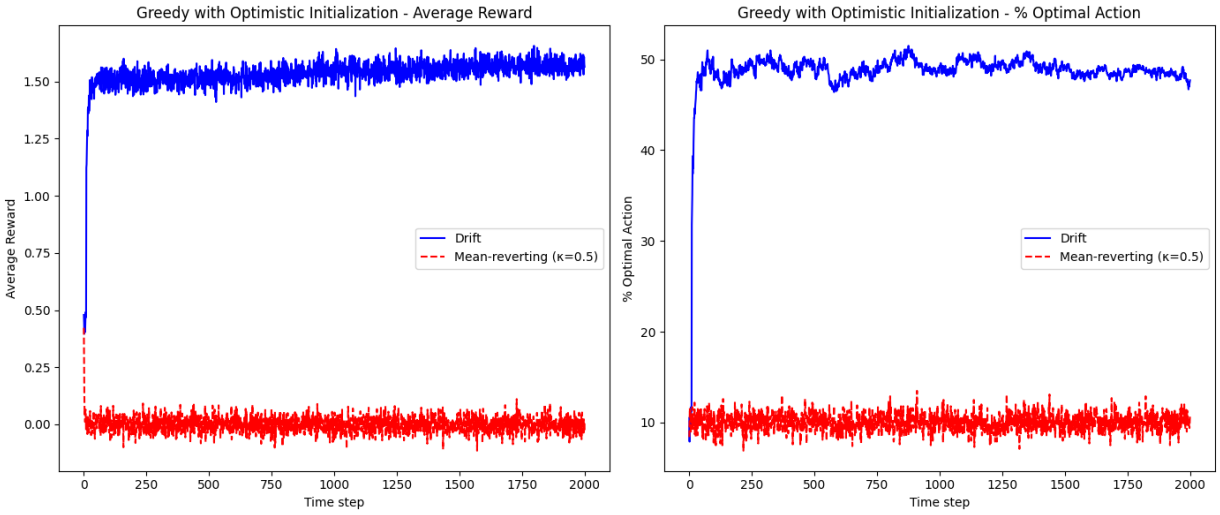

Figure 7: Greedy with optimistic initialization under gradual non-stationarity

Drift Environment (Blue Solid Line): In the presence of slowly drifting rewards, the optimistic initialization performs reasonably well. The average reward converges quickly, stabilizing around 1.55. However, compared to ε-greedy, the percentage of optimal actions selected is lower, plateauing around 48–50%. This indicates that while the initial optimism promotes exploration early on, the absence of continued exploration limits adaptability as the environment changes.

Mean-Reverting Environment (Red Dashed Line): As with the previous method, performance is poor. The average reward remains close to zero, and optimal actions are selected at random-like frequencies (10%), suggesting that the agent fails to keep up with the frequent switches in optimal action. Once the initial optimism is exhausted, the algorithm behaves like a purely greedy strategy, which is ill-suited to high-variance or oscillating conditions.

Setting	**Final Avg. Reward**	**Final % Optimal Action**
Drift	~ 1.55	$\sim 48\%$
Mean-Reverting ($\kappa = 0.5$)	~ 0.0	$\sim 10\%$

Table 5: Summary of greedy with optimistic initialization under gradual non-stationarity.

Interpretation: Optimistic initialization offers a one-time incentive to explore, which is effective when the environment is initially unknown but slowly changing. However, its static nature makes it unsuitable for environments where the optimal action can revert or shift frequently. Continuous or adaptive exploration mechanisms are more appropriate in such settings.

3. Gradient Bandit

Figure 8 shows the performance of the gradient bandit algorithm in environments with gradual changes. Unlike value-based methods, the gradient bandit method directly optimizes action preferences using a softmax policy and receives baseline-subtracted rewards for stability.

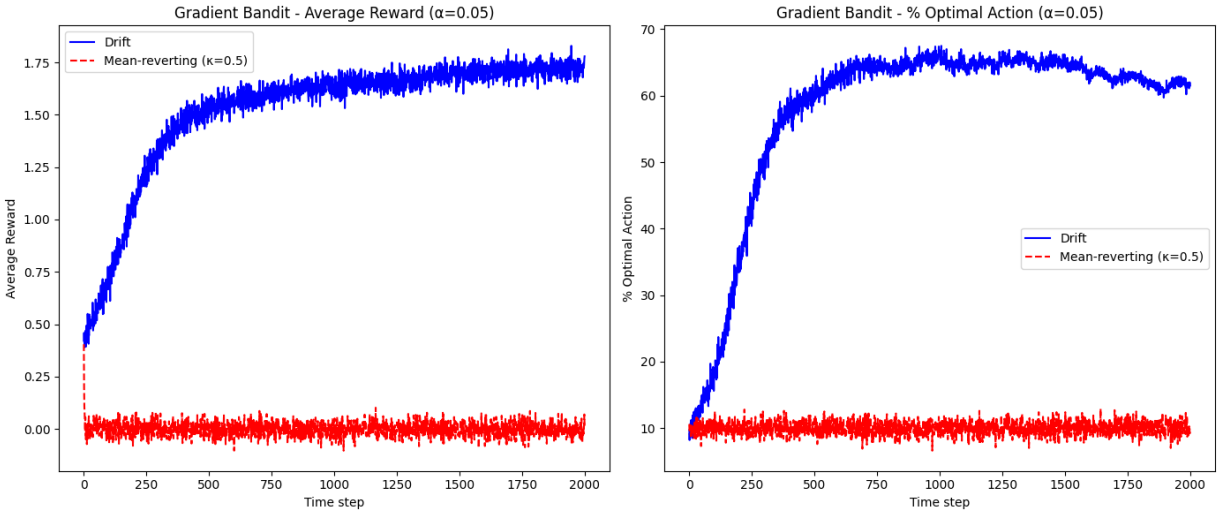

Figure 8: Gradient bandit algorithm with $\alpha = 0.05$ under gradual non-stationarity

Drift Environment (Blue Solid Line): In the slowly drifting reward setting, the gradient bandit algorithm performs competitively. The average reward improves rapidly and converges to a value around 1.75. The proportion of optimal actions also rises steadily, peaking at about 65% and remaining relatively stable. This demonstrates the gradient method's capacity to track gradual changes when a reasonable learning rate is used.

Mean-Reverting Environment (Red Dashed Line): The performance in the mean-reverting setting remains poor. The average reward fluctuates around zero, and the optimal action is chosen only 10–12% of the time. This suggests that although the gradient method can adapt to trends, it lacks the responsiveness or robustness required for environments that frequently oscillate in their optimal action.

Setting	Final Avg. Reward	Final % Optimal Action
Drift	~ 1.75	$\sim 63\%$
Mean-Reverting ($\kappa = 0.5$)	~ 0.0	$\sim 11\%$

Table 6: Summary of gradient bandit performance under gradual non-stationarity.

Interpretation: The gradient bandit algorithm adapts well to smoothly drifting environments due to its continuous update mechanism. However, it still struggles under mean-reverting dynamics, likely because the fixed learning rate cannot compensate for the environment's rapidly changing direction of optimality. More responsive mechanisms, such as time-varying learning rates or hybrid exploration strategies, may be needed to address these limitations.

4. Standard Greedy (Zero Initialization)

Figure 9 shows the performance of the standard greedy algorithm initialized with zero action values in a gradually changing environment.

Drift Environment (Blue Solid Line): In the slowly drifting environment, the greedy strategy achieves a moderate performance. The average reward stabilizes just above 1.0, and the percentage of optimal action selections remains around 23–25%. This is significantly lower than the performance of more adaptive methods like ε-greedy or gradient bandits.

The reason is that, with zero initialization and no explicit exploration, the greedy policy prematurely commits to suboptimal actions. Once an action's initial value happens to yield

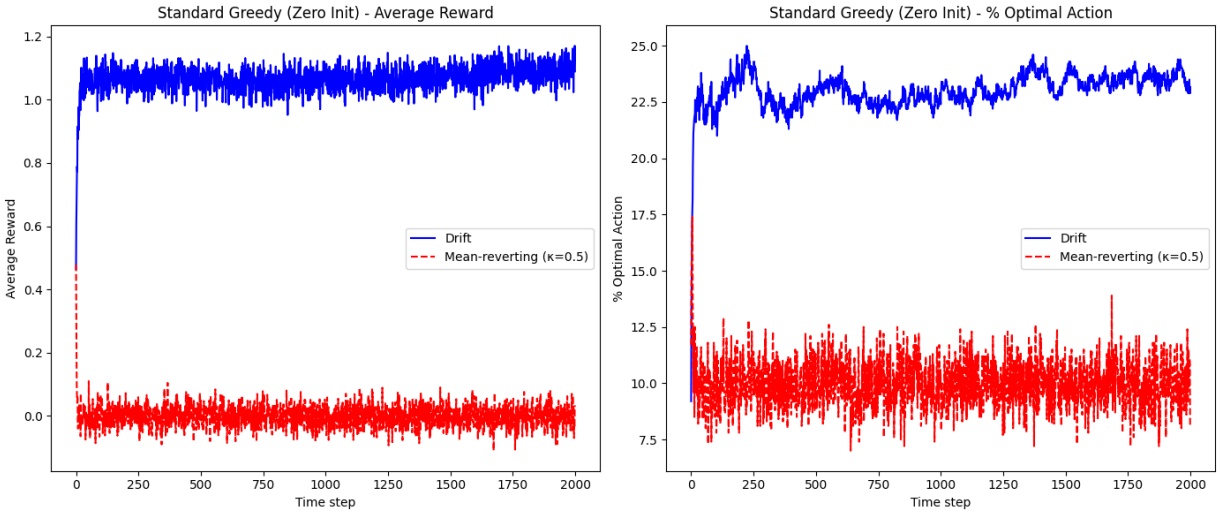

Figure 9: Standard greedy algorithm with zero initialization under gradual non-stationarity

the highest reward in early timesteps, it is consistently selected, even if better actions emerge over time.

Mean-Reverting Environment (Red Dashed Line): In the mean-reverting setting, performance is poor. The average reward hovers around zero, and optimal action selection remains near the chance level (10%), similar to previous methods. This result reinforces the need for active exploration in volatile environments.

Setting	**Final Avg. Reward**	**Final % Optimal Action**
Drift	~ 1.1	$\sim 23\%$
Mean-Reverting ($\kappa = 0.5$)	~ 0.0	$\sim 10\%$

Table 7: Performance of standard greedy under gradual non-stationarity.

Interpretation: The greedy strategy, while simple and computationally efficient, performs poorly in non-stationary settings. Its lack of exploration and reliance on initial estimates render it brittle to environmental changes, especially in dynamic or oscillating reward landscapes.

0.6.1 Abrupt Changes in Reward Distribution

To investigate the behavior of bandit algorithms under abrupt changes, a changepoint is introduced at time step $t = 501$, where the means μ_i of the ten-armed bandit are randomly permuted. This setup simulates an environment in which reward expectations change suddenly and unpredictably.

Two experimental conditions are considered:

1. **No Reset:** Algorithms continue learning as usual beyond the changepoint, with no modification to internal estimates or parameters.

2. **Hard Reset:** At $t = 501$, all internal action-value estimates and relevant learning parameters are reset to their initial values, simulating foreknowledge of the changepoint.

Each method is run over 2000 time steps and averaged across 1000 independent simulations. The same random permutation of means and identical random seeds for drift/noise processes are used across all simulations to maintain consistency and comparability.

Performance is evaluated in terms of two metrics: (1) the average per-step reward, and (2) the proportion of time the optimal action is selected.

The goal of this section is to assess and compare the adaptability of action-value methods and gradient-based methods in environments with abrupt, unforeseen changes in reward structure.

1. Epsilon-Greedy Algorithm ($\varepsilon = 0.05$)

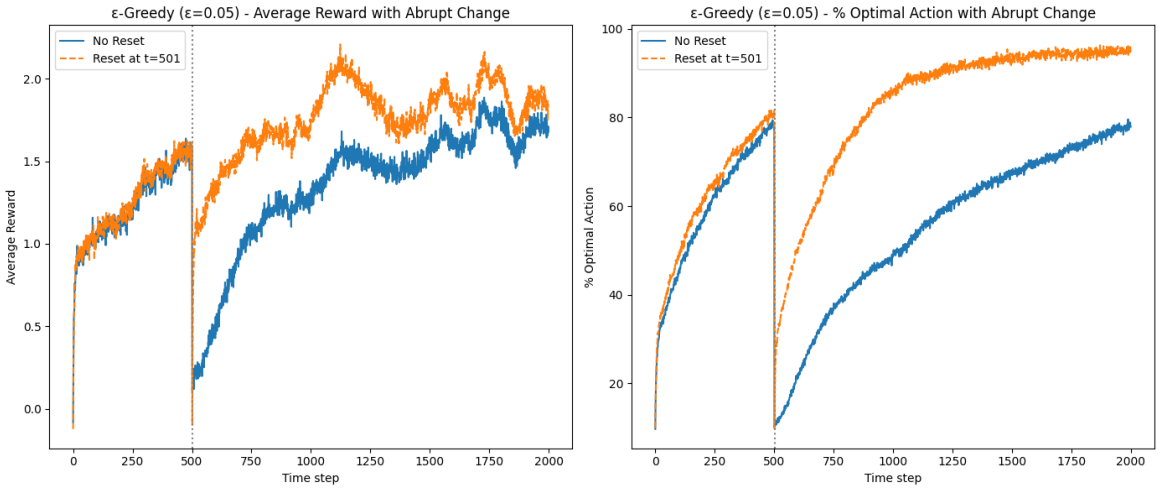

Figure 10: ε-Greedy ($\varepsilon = 0.05$) A comparison is shown between runs with no reset and those with a reset at $t = 501$.

After the abrupt permutation of reward means at $t = 501$, a clear divergence is observed between the reset and no-reset conditions. In the **no-reset** condition, performance deteriorates sharply following the changepoint. The average reward drops significantly and recovers only gradually, reflecting the algorithm's struggle to unlearn outdated action values and re-identify the new optimal arm. Similarly, the percentage of optimal actions selected decreases after $t = 501$ and then slowly increases, showing delayed adaptation.

In contrast, the **reset** condition demonstrates a substantial improvement in adaptability. By clearing previous action-value estimates at $t = 501$, the algorithm treats the environment as new, allowing faster convergence to the new optimal action. This is evident in both the average reward and the percentage of optimal actions, which begin improving almost immediately after the reset. The reset curve consistently outperforms the no-reset curve in both metrics after the changepoint. These results highlight the limitation of static action-value estimators in non-stationary environments and illustrate the potential of simple reset mechanisms to enhance responsiveness to abrupt environmental changes.

2. Greedy Optimistic

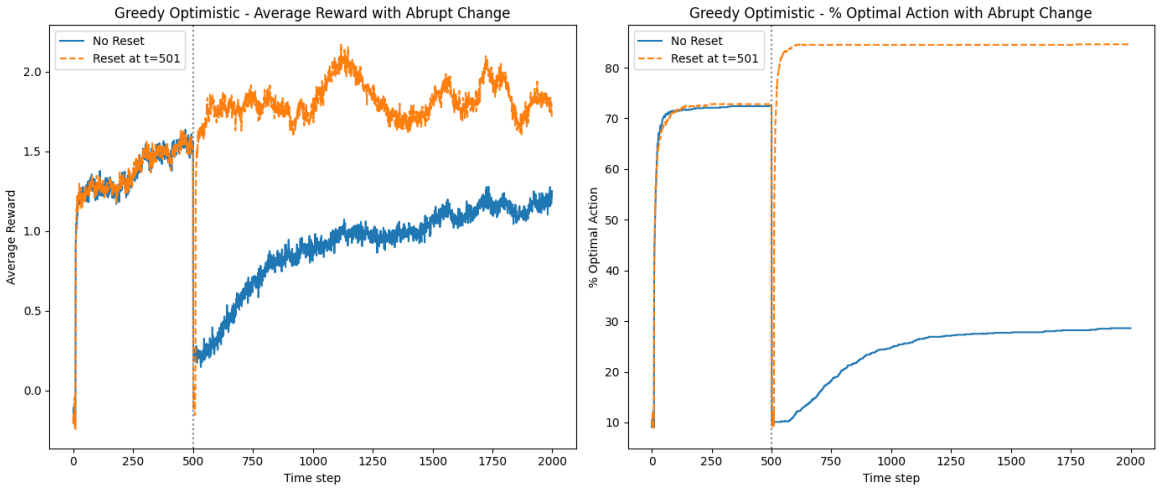

Figure 11: Greedy Optimistic. Comparison is made between no reset and reset at $t = 501$.

The Greedy Optimistic method demonstrates high initial performance due to optimistic initialization, which encourages exploration. Prior to the abrupt change at $t = 501$, the algorithm performs well in both the average reward and percentage of optimal actions selected, quickly converging to a high proportion of optimal actions.

After the changepoint, a clear divergence is observed. In the **no-reset** condition, performance deteriorates sharply. The algorithm remains biased by outdated estimates, causing a prolonged decline in average reward and a stagnation in the percentage of optimal actions selected. The system fails to recover efficiently, demonstrating the vulnerability of purely greedy methods in non-stationary contexts.

In contrast, the **reset** condition shows a sharp drop at $t = 501$ followed by rapid recovery. Resetting action-value estimates allows the method to re-explore and re-identify the new optimal action quickly. The percentage of optimal action selection returns to a high level almost immediately after the reset, and average reward recovers to a value notably higher than in the no-reset case.

This result confirms that while Greedy Optimistic benefits from initial aggressive exploration,

its performance under abrupt changes can be significantly improved with the addition of a hard reset strategy at known changepoints.

3. Standard Greedy(Q=0)

Figure 12: Standard Greedy. Results are shown with and without reset at $t = 501$.

The Standard Greedy algorithm, which always selects the action with the highest estimated value and performs no exploration, struggles significantly in dynamic environments. Before the abrupt change at $t = 501$, the algorithm achieves moderate performance, with a relatively low percentage of optimal actions selected and a slowly increasing average reward.

After the changepoint, performance deteriorates notably in the **no-reset** condition. Without any exploratory behavior or reset mechanism, the algorithm continues to exploit outdated estimates, resulting in a stagnant or even declining performance. The percentage of optimal actions remains low, and the average reward does not fully recover within the observed time horizon.

In the **reset** condition, there is a sharp drop at the changepoint due to reinitialization of action values, followed by a partial recovery. Although the average reward increases post-reset, it remains limited by the lack of exploration. The percentage of optimal actions plateaus quickly and fails to reach high levels, indicating the algorithm locks into suboptimal actions early.

These results underscore the limitations of purely greedy strategies in non-stationary settings.

Even with a hard reset, without exploration, the algorithm cannot reliably identify and adapt to new optimal actions after abrupt changes.

4. Gradient Bandit

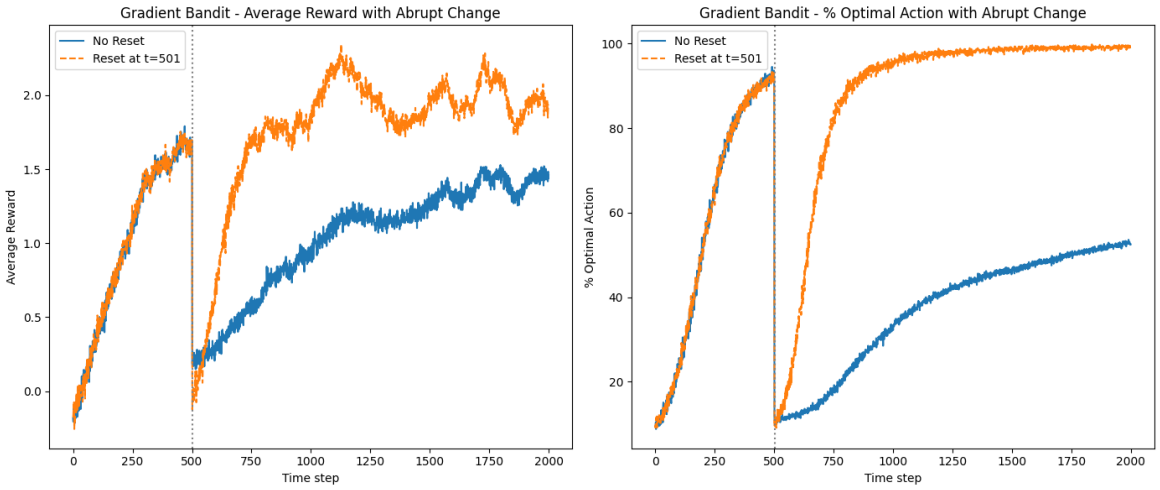

Figure 13: Gradient Bandit. Results are shown with and without reset at $t = 501$.

The Gradient Bandit algorithm demonstrates strong adaptability in both the **no-reset** and **reset** conditions, significantly outperforming standard greedy approaches.

Prior to the abrupt change at $t = 501$, the algorithm steadily increases both the average reward and the percentage of optimal actions. In the **no-reset** case, performance drops sharply right after the changepoint, as expected, due to outdated action preferences. However, thanks to its continuous update of preferences based on rewards, the algorithm is able to gradually recover and improve performance over time, even without any reset.

The **reset** version, which reinitializes preferences and baselines at $t = 501$, adapts more quickly. Following an initial drop, the average reward and optimal action percentage rapidly recover and eventually surpass the no-reset condition. This highlights the value of combining policy-based adaptation with strategic resets in non-stationary environments.

Overall, the Gradient Bandit method, especially when paired with a reset strategy, proves to be highly effective at tracking changing reward distributions. Its ability to recover from abrupt shifts

and maintain high performance emphasizes its robustness and suitability for dynamic problems.

0.7 Conclusion

This experiment compared four bandit algorithms—**Standard Greedy**, **ε-Greedy**, **Greedy with Optimistic Initialization**, and **Gradient Bandit**—on a non-stationary environment featuring an abrupt change in the reward distribution at $t = 501$. Each algorithm was evaluated with and without a reset at the changepoint, using two performance metrics:

1. **Average per-step reward**
2. **Proportion of time the optimal action is selected**

Average Per-Step Reward:

- **Gradient Bandit (with reset)** consistently achieved the highest average rewards, demonstrating effective adaptation through its preference-based update mechanism.
- **Optimistic Greedy (with reset)** performed comparably well post-change, aided by re-initialization, though with more variability.
- **ε-Greedy** maintained moderate performance, supported by continued exploration. Resetting improved responsiveness to the reward shift.
- **Standard Greedy** produced the lowest rewards across both conditions, hindered by its lack of exploration and fixed estimates.

Proportion of Optimal Actions Taken:

- **Gradient Bandit (with reset)** approached near-optimal action selection rapidly after the changepoint, maintaining dominance across all time steps.

- **Optimistic Greedy** initially selected the optimal action frequently but showed performance decay without reset.
- **ε-Greedy** demonstrated steady recovery, with exploration enabling rediscovery of the new optimal arm.
- **Standard Greedy** selected the optimal action least often and adapted poorly even with a reset.

General Observations:

- Resetting action-value estimates or preferences at changepoints significantly enhances adaptability for all methods.
- Gradient Bandit is the most robust and adaptive approach under abrupt non-stationarity.
- ε-Greedy balances exploration and exploitation effectively, performing reliably in dynamic settings.
- Greedy with optimistic initialization benefits from early exploration but is sensitive to changes unless explicitly reset.
- Standard Greedy is unsuitable for non-stationary environments due to its deterministic nature and inflexibility.

0.7.1 Summary

In non-stationary environments, algorithms that facilitate ongoing exploration or incorporate adaptive reset mechanisms—particularly Gradient Bandit and ε-Greedy—outperform rigid, deterministic strategies such as Standard Greedy.

References

Shestopaloff, A. (2025). Reinforcement learning lecture notes. Lecture notes, Memorial University of Newfoundland. Accessed during DSCI 6605-001 course, Spring 2025.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts and London, England, 2 edition. In progress draft (2014, 2015 versions available online).