Simulation Methods for Optimization and Learning - Heidergott Project

Group 7 Members:

Jiakang Huang, Hongkai Liu, Shengzhe Ji, Jiahe Jiang,

Chenghao Cao

Supervisor:

Dr. Bernd Heidergott

TAs: Franssen Christian & Zehao Li

Abstract

The Stochastic Approximation (SA) algorithm is a powerful method for solving optimization problems (Nemirovski et al., 2009) that involve noisy or difficult-to-evaluate functions. The SA algorithm iteratively adjusts parameters to converge towards the optimal solution by using stochastic perturbations and gradient estimates. This approach has been extensively studied and applied in various fields, including queueing theory and optimization. Notably, Kushner and Yin (2003) provide a comprehensive analysis of SA methods and their applications, while Spall (1998) offers an overview of the simultaneous perturbation method for efficient optimization.

This report addresses two distinct problems. The first part focuses on optimizing an investment strategy for an investor who plans to invest in three companies over a specific time period, with the objective of maximizing the investment's risk-adjusted performance, as measured by the Sharpe ratio. The second part concentrates on the optimization of service times in a GI/GI/1 queueing system with a first-come, first-served (FCFS) discipline. This involves adjusting the server's mean service time to ensure the average waiting time for customers closely approximates a desired target.

Acknowledgements

We would like to express our sincere gratitude to Professor Bernd Heidergot for his invaluable guidance and insightful lectures, which provided the foundational knowledge for our project. His expertise and dedication have greatly influenced our understanding and execution of the optimization methods utilized in this study.

We also extend our heartfelt thanks to our Teaching Assistants, Franssen Christian and Zehao Li, for their continuous support and patience. Their assistance during tutorials and willingness to address our queries outside of class hours have been instrumental in the successful completion of this project. Their constructive feedback and clarifications have significantly enhanced our learning experience.

We are deeply appreciative of the time and effort invested by our instructors in ensuring our understanding and application of the concepts presented in the coursework and tutorials.

Contents

1 Question 1: Investment Strategy Optimization

1.1 Introduction

In this project, we aim to find the optimal investment strategy for an investor who wants to invest in n companies (a total of 3 companies). The goal is to maximize the risk-adjusted performance of the investment, measured by the Sharpe ratio, by determining the optimal allocation vector (p_1, p_2, p_3). The investor's objective is to allocate the total capital of 1 across the three companies in a way that maximizes the expected return while minimizing risk.

Figure 1: Illustration of investment returns.

Source: `https://giphy.com/gifs/cash-finance-dollars-BfFFYPSVYr9UR6EtEL`

1.2 Parameter Definition

The parameters and variables used in this analysis are as follows:

1. n: Number of companies (totally 3).
2. t: Run time.
3. i: A specific company (1, 2, 3).
4. X_i: Market Value of company i.
    - Formula depends on ρ, V, η_i, and W.
    - $$X_i = \frac{\rho V + \sqrt{1-\rho^2}\eta_i}{\max(W, 1)}, \quad 1 \le i \le n$$

5. x_i: The minimum market value to get profit.
   - $x_1 = 2$, $x_2 = 3$, and $x_3 = 1$.
   - If $X_i \geq x_i$, then we can get the profit.
   - If $X_i < x_i$, then the profit is zero.
   - $p_i E[Y_i \mathbf{1}_{X_i \geq x_i}]$
6. Y_i: The profit return by the investment.
   - Independent uniformly distributed on $[0, X_i]$.
7. p_i: The fraction of the investment for each company.
   - $p_1 + p_2 + p_3 = 1$.
   - $\sum_{i=1}^{n} p_i = 1$ and $0 \leq p_i \leq 1$
8. η_i: Company's idiosyncratic risk.
   - Normally distributed with mean 0 and variance i.
9. V: Common factor that affects the economy.
   - Standard normally distributed.
10. W: Common market shocks.
    - Exponentially distributed with rate $\frac{1}{0.3}$.
11. ρ: Weight factor.
    - $\rho = 0.6$.

Notice that V, W, and η_i are all independent.
In summary, the investor aims to find the optimal allocation of their capital (p_1, p_2, p_3) to maximize the risk-adjusted performance of the investment, also known as the Sharpe ratio.

1.3 Well-Posedness of this Optimization Problem

Problem Statement

The investor has a total capital of 1 to invest in n companies over time t. The market value of company i at time t is given by X_i. The company i generates profit Y_i if $X_i \geq x_i$. The investor seeks to find the optimal investment strategy by maximizing the risk-adjusted performance, also known as the Sharpe ratio.

Optimization Formulation

$$\max_{p_1,p_2,p_3\geq 0,\sum p_i=1} E\left[\frac{\sum_{i=1}^{3} p_i Y_i 1_{X_i\geq x_i}}{\text{std}\left(\sum_{i=1}^{3} p_i Y_i 1_{X_i\geq x_i}\right)}\right]$$

where:

- p_i are the fractions of capital allocated to each company,
- Y_i are uniformly distributed profits,
- X_i are the market values of the companies modeled by:

$$X_i = \frac{\rho V + \sqrt{1-\rho^2}\eta_i}{\max(W,1)},$$

with η_i normally distributed with mean 0 and variance i, V standard normally distributed, and W exponentially distributed with rate $1/0.3$.

Criteria for Well-Posedness

A problem is well-posed if:

1. **Existence of Solutions**: There must be at least one solution.
2. **KKT Points**: All solutions must be Karush-Kuhn-Tucker (KKT) points.
3. **Boundedness**: The set of KKT points must be bounded.

Analysis

1. **Existence of Solutions**:
   - The feasible region defined by $0 \leq p_i \leq 1$ and $\sum_{i=1}^{3} p_i = 1$ ensures that there are possible solutions within the bounded simplex in R^3.
2. **KKT Points**:
   - The problem involves optimizing a ratio (Sharpe ratio), which can be complex. However, given the constraints $\sum_{i=1}^{3} p_i = 1$ and $0 \leq p_i \leq 1$, we can leverage Lagrangian multipliers to find stationary points.
   - For well-posedness, these stationary points should satisfy the KKT conditions, ensuring that they are either maxima or saddle points within the feasible region.
3. **Boundedness**:
   - The constraints on p_i (i.e., $0 \leq p_i \leq 1$ and $\sum p_i = 1$) inherently provide a bounded feasible region. This ensures that the solution space does not extend to infinity, which is crucial for well-posedness.

- The function to be maximized, the Sharpe ratio, is also bounded by the nature of Y_i and X_i being generated from uniform and normal distributions, respectively.

4. **No Direction of Unbounded Decrease**:

   - Since the objective is to maximize the Sharpe ratio, and given the bounded feasible region, there are no directions in which the objective function decreases indefinitely.

Conclusion

The given optimization problem is likely well-posed because:

- It has a non-empty feasible region.
- Solutions are expected to satisfy KKT conditions.
- The feasible region is bounded, preventing solutions from extending to infinity.
- The objective function does not have unbounded directions of decrease within the feasible region.

1.4 Description of the Program

1.4.1 Simultaneous Perturbation Stochastic Approximation (SPSA) Algorithm

The SPSA algorithm for optimizing the Sharpe Ratio works as follows:

1. **Initialization**: Start with an initial guess for the weight vector $\mathbf{p}$.
2. **Iteration**: For each iteration:
   (a) Generate correlated normal samples $\mathbf{X}$ and uniform samples $\mathbf{Y}$.
   (b) Compute the Sharpe ratio for the given weight vector.
   (c) Estimate the gradient of the objective function (Sharpe ratio) using stochastic perturbations.
   (d) Adjust the weight vector $\mathbf{p}$ based on the estimated gradient.
   (e) Project the updated weight vector back onto the simplex to ensure the weights sum to 1 and are non-negative.
   (f) Check for convergence based on the change in the objective function value or a predefined number of iterations.

1.4.2 Parameter Settings

- **Number of companies:** $n = 3$
- **Weight factor:** $\rho = 0.6$
- **Minimum market values to get profit:** $x = \{2, 3, 1\}$
- **Maximum number of iterations for the optimization algorithm:** 1000
- **Initial investment fractions for each company:** $p = \{0.3517, 0.1819, 0.4664\}$
- **Decay rate for the stochastic approximation algorithm:** $\alpha = 0.9$
- **Initial learning rate for the stochastic approximation algorithm:** $\epsilon = 0.1$

1.4.3 Distribution of Random Variables

The following code is used to generate the random variables V, W, and η_i:

- V follows a standard normal distribution.
- W follows an exponential distribution with a rate of $1/0.3$.
- η_i follows normal distributions with mean 0 and variances i for $i = 1, 2, 3$.

The distributions are visualized as follows:

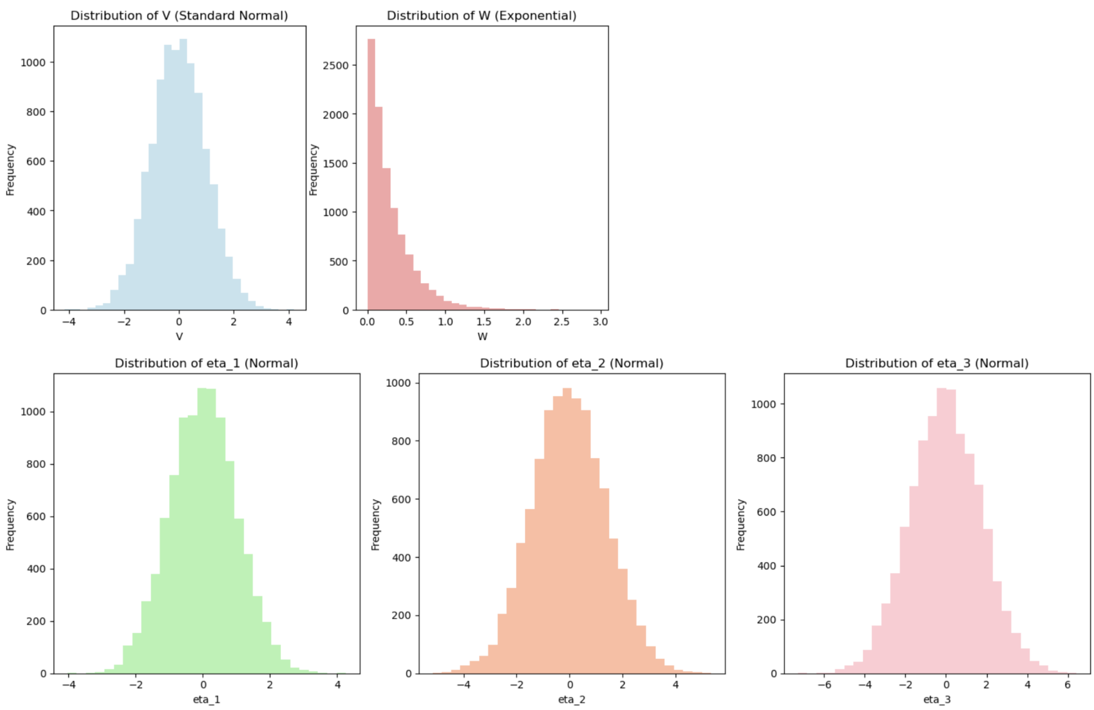


Figure 2: Illustration of investment returns.

Summary: From the distributions plotted above, we can observe that the random variables V, W, and η_i follow their respective required distributions:

- V follows a standard normal distribution.
- W follows an exponential distribution with a rate of $1/0.3$.
- η_i follows normal distributions with mean 0 and variances of i for $i = 1, 2, 3$.

These observations confirm that our method for generating these random variables is effective and accurate. The distributions align with the theoretical expectations, validating our approach.

1.5 Simulation and Distribution of Conditional Expectations

We performed 1000 simulations to collect data on the conditional expectations $E[Y_i \cdot 1_{X_i \geq x_i}]$ for three companies. The collected data was then converted into a DataFrame and visualized using histograms to show the distribution of these conditional expectations.

- **Data Collection**: We ran 1000 simulations and collected the conditional expectations for each company.
- **Data Conversion**: The collected data was converted into a DataFrame with columns representing each company's conditional expectation.
- **Visualization**: Histograms were plotted to visualize the distribution of the conditional expectations for each company.

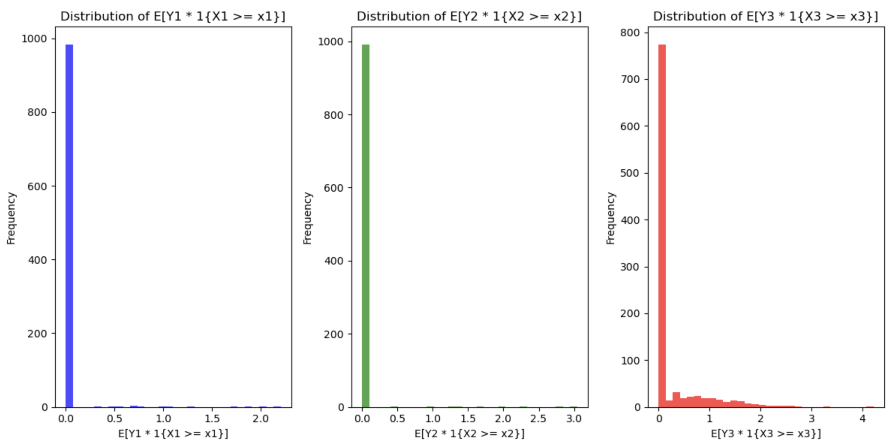


Figure 3: Distribution of Conditional Expectations $E[Y_i \cdot 1_{X_i \geq x_i}]$ for three companies

Summary: The histograms above illustrate the distribution of the conditional expectations for the three companies. Each subplot represents the distribution for one company, showing the frequency of different values of $E[Y_i \cdot 1_{X_i \geq x_i}]$. These distributions provide insight into the expected returns conditioned on the market values meeting the threshold x_i.

1.6 3D Plot of Sharpe Ratio Surface Based on Conditional Expectations

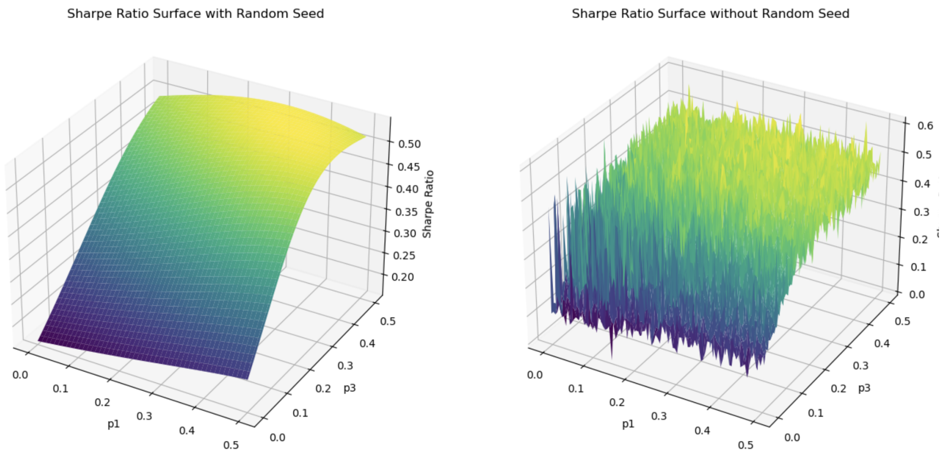


Figure 4: Sharpe Ratio Surface Based on Conditional Expectations

Summary: Based on the two plots, we can make several observations. Firstly, the plot with a random seed clearly shows that as (P3) increases, the overall Sharpe ratio also increases. This result aligns with our previous section's bar chart, suggesting that the company represented by (P3) is more likely to yield profits. Secondly, when the Sharpe ratio reaches a certain value, it gradually levels off and approaches a stable value. This trend is shown as the flattened part of the curve in the plot. Lastly, the plot without the random seed highlights that our model contains numerous local maxima and is particularly sensitive to small parameter changes. This insight is crucial for our understanding of the model parameters and their impact, guiding us in fine-tuning our future models.

1.7 SPSA Algorithm:

1.7.1 Summary

- **Gradient Estimation**: We implemented a function to estimate the gradient of the objective function using the SPSA method. This method involves perturbing the solution, evaluating the objective function at the perturbed points, and computing the gradient estimate based on the differences in these evaluations.
- **Projection Step**: We included a projection function to ensure that the investment fractions always sum to 1 and remain non-negative. This step is crucial to keep the solution within the feasible region.
- **SPSA Algorithm**: We implemented the SPSA algorithm to optimize the investment strategy. This algorithm iteratively updates the solution by moving in the direction of the estimated gradient. The step size for the updates decreases over time, allowing for convergence to an optimal solution. The algorithm prints the current Sharpe ratio and investment fractions at regular intervals, providing insight into the optimization process.

1.7.2 Results of SPSA Algorithm

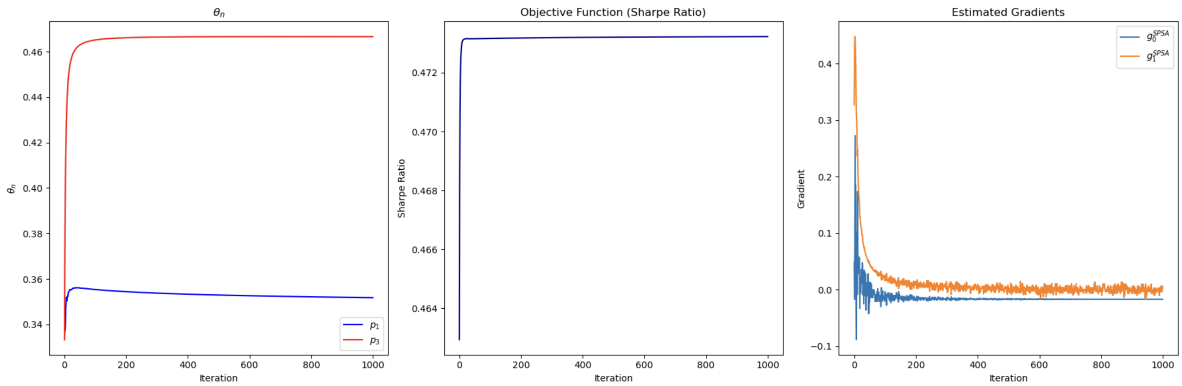


Figure 5: Objective Function (Sharpe Ratio) Over Iterations

The three plots above demonstrate the changes in investment proportions for companies (P1) and (P3) using the SPSA algorithm.

First Plot: Investment Proportions

- The first plot shows the investment proportions for companies (P1) and (P3) over the iterations.
- We can see that both proportions converge to stable values. Specifically, (P1) converges to approximately 0.35, and (P3) converges to around 0.6.
- This indicates that the algorithm is stabilizing, suggesting an optimal allocation strategy for these two companies.

Second Plot: Sharpe Ratio

- The second plot displays the Sharpe Ratio over the iterations.
- The highest Sharpe Ratio converges above 0.472.
- The Sharpe Ratio shows a rapid increase initially, followed by stabilization as the iterations progress.

Third Plot: Estimated Gradients

- The third plot shows the estimated gradients for (g1) and (g2), representing the gradients for (P1) and (P3) respectively.
- Both gradients converge to values close to zero, demonstrating a trend towards stabilization.
- This indicates that the model is progressively approaching the optimal point, reflecting the stability and reliability of our model.

In summary, these plots collectively demonstrate that the SPSA algorithm effectively stabilizes the investment proportions for companies (P1) and (P3). The Sharpe Ratio remains consistently high, and the gradients converge towards zero, indicating the robustness and effectiveness of the model.

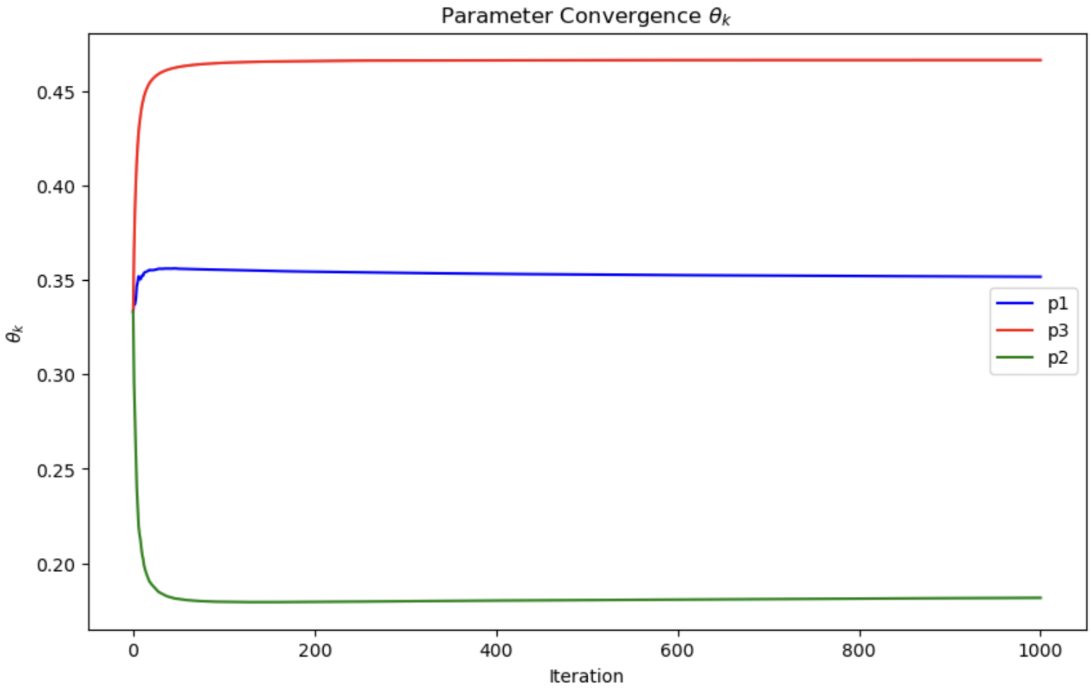


Figure 6: Parameter Convergence Over Iterations

The graph now includes P2, which represents the proportion $(1 - P1 - P3)$. We observe that P2 also approaches a **flat line below 0.2**. This confirms our previous hypothesis:

- **P1** is moderately profitable, with its proportion stabilizing around 0.35.
- **P2** is the least profitable company, leading to a decline in its proportion, stabilizing below 0.1.
- **P3** is the company most likely to yield profit, thus its proportion increases and stabilizes around 0.45.

These observations demonstrate that the SPSA algorithm effectively adjusts the investment proportions to maximize returns by allocating more capital to the better-performing companies.

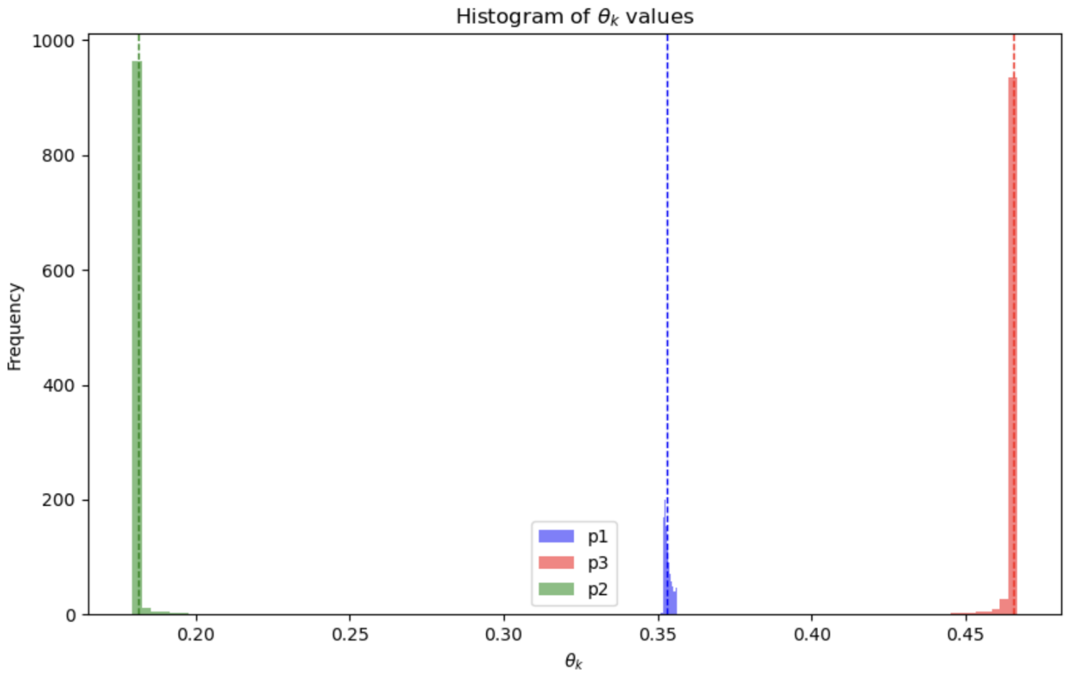


Figure 7: Histogram of Theta Values

This histogram further confirms our conclusions. The distribution of **P3** is **skewed to the right**, indicating that our model attempts to maximize the value of P3 as much as possible. The distribution of **P1** appears to follow a **standard distribution**, centered around 0.35, suggesting a stable proportion for P1. On the other hand, the distribution of **P2** is **skewed to the left**, demonstrating that our model is progressively decreasing the proportion allocated to P2, as it is the least likely to generate profit.

1.7.3 Parameter Convergence

These three plots provide a more detailed view of the final convergence values for each investment proportion.

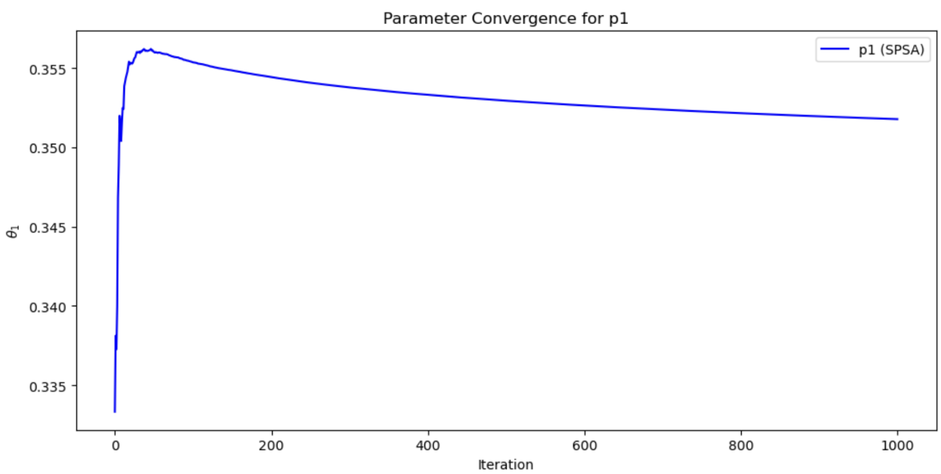


Figure 8: Parameter Convergence for P1

First Plot (P1 Proportion)

- The first plot shows that the proportion for $P1$ converges to approximately 0.355.
- This indicates that the optimal investment strategy stabilizes $P1$ around this value.

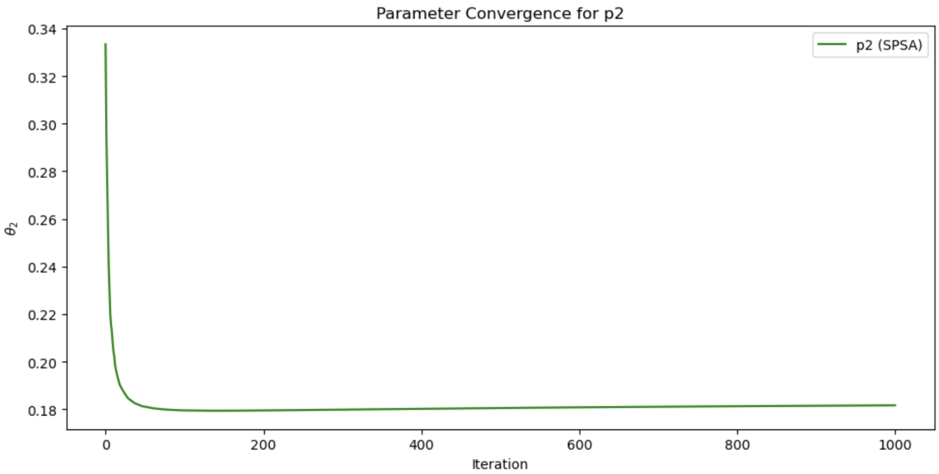


Figure 9: Parameter Convergence for P2

Second Plot (P2 Proportion)

- The second plot shows the proportion for $P2$ converging to around 0.18.
- This suggests that the algorithm considers $P2$ as a less favorable investment, reducing its proportion significantly.

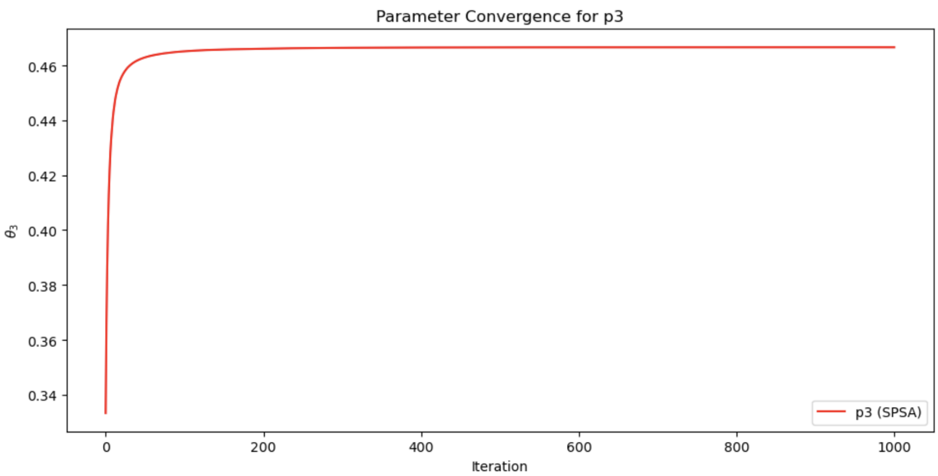


Figure 10: Parameter Convergence for P3

Third Plot (P3 Proportion)

- The third plot demonstrates that the proportion for $P3$ converges to approximately 0.46.
- This highlights $P3$ as the most favorable investment according to the optimization model, assigning it the highest proportion.

In summary, these plots confirm the final convergence values for each proportion, with $P1$ stabilizing around 0.355, $P2$ around 0.18, and $P3$ around 0.46. This detailed view further validates the robustness and effectiveness of our SPSA optimization model.

1.8 Statistical Analysis and Validation:

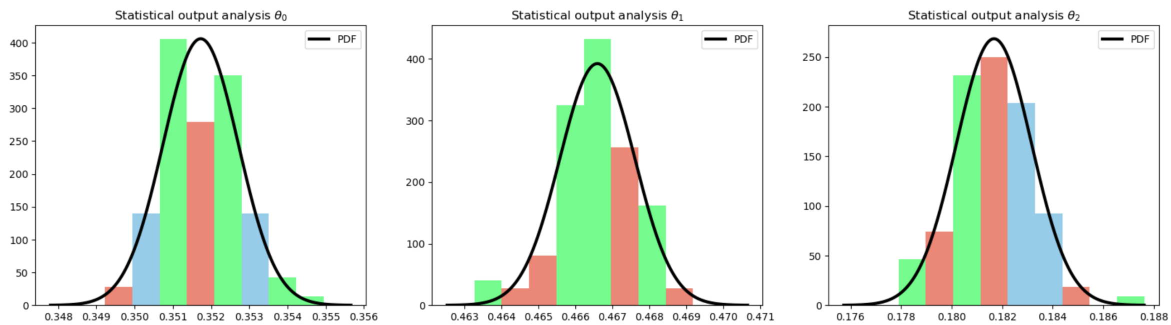


Figure 11: Statistical Analysis of Theta Values

Model Summary and Analysis

Overview

The given model uses the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm to optimize the allocation of investments among three companies, denoted as θ_0 (P1), θ_1 (P3), and θ_2 (P2), with the objective of maximizing the Sharpe ratio.

Parameter Convergence

1. θ_0 (P1): The parameter θ_0 converges to approximately 0.3517, indicating that around 35.17% of the investment is allocated to the first company.
2. θ_1 (P3): The parameter θ_1 converges to approximately 0.4664, indicating that around 46.64% of the investment is allocated to the third company.

3. θ_2 (P2): The parameter θ_2 converges to approximately 0.1819, indicating that around 18.19% of the investment is allocated to the second company.

Statistical Analysis

The statistical analysis of the final parameter values shows the following:

Mean and Standard Deviation:

- θ_0: Mean = 0.3517, Std. Dev. = 0.00095
- θ_1: Mean = 0.4664, Std. Dev. = 0.0013
- θ_2: Mean = 0.1819, Std. Dev. = 0.00133

Histogram Analysis:

- The histograms indicate the distribution of the final parameter values across multiple repetitions of the SPSA algorithm. Each histogram is overlaid with a normal distribution curve (PDF) for comparison.
- The histogram bars are colored based on their comparison with the expected normal distribution:
  - **Red**: Bars below the expected PDF value.
  - **Green**: Bars above the expected PDF value.
  - **Blue**: Bars approximately equal to the expected PDF value.

1.9 Conclusion for Q1

- The parameters P_1, P_2, and P_3 have converged to stable values, suggesting the model has effectively optimized the investment allocation.
- The mean values and low standard deviations indicate that the model produces consistent results across multiple runs.

- The histograms show that the final parameter values fit well within a normal distribution, further validating the stability and reliability of the model.

In summary, the SPSA algorithm has successfully optimized the investment allocation, achieving stable and consistent parameter values that maximize the Sharpe ratio. The statistical analysis confirms the robustness of the model.

2 Question 2: Optimization of Service Times in GI/GI/1 Queueing Model

2.1 Introduction

2.1.1 Objective of the Study

The objective of this study is to optimize the mean service time (θ) in a GI/GI/1 queueing model to minimize the average waiting time for customers. By employing a Stochastic Approximation (SA) algorithm (Robbins et al., 1951), the study aims to ensure that the first N customers do not have to wait more than a specified threshold z time units, thereby improving service efficiency and customer satisfaction.

The problem is formulated as follows: customers arrive at the service station according to a renewal point process with inter-arrival times $\{A_n : n \in N\}$ that are i.i.d., and service times $\{S_n(\theta) : n \in N\}$ that are also i.i.d. and depend on the mean service time θ. The recursion for the waiting times $W_n(\theta)$ is given, and the average waiting time over the first N customers is denoted by $W_N(\theta)$. The objective is to minimize the expectation of the squared difference between the 90% quantile of $W_N(\theta)$ and the threshold z, leading to the optimization problem:

$$\min_{0<\theta} E[(q_\theta - z)^2]$$

where q_θ represents the 90% quantile of $W_N(\theta)$.

2.1.2 Parameter Definition

1. n: Number of customers.

2. A_n: Inter-arrival times for n customers.

   - $\{A_n : n \in N\}$

   - A_n follows an exponential distribution and is an i.i.d variable.

3. $S_n(\theta)$: Service times for n customers.

- $\{S_n(\theta) : n \in N\}$
- $S_n(\theta)$ follows an exponential distribution and is an i.i.d variable.

4. $W_n(\theta)$: Waiting times for n customers.

- $\{W_n(\theta)\}$
- Formula to calculate the waiting time:

$$W_{n+1}(\theta) = \max(0, W_n(\theta) + S_n(\theta) - A_{n+1})$$

2.1.3 Check for Well-Posedness

Analysis

1. **Existence of Solutions**:

   - The feasible region is defined by $0 < \theta$. Given the continuity of the waiting time distribution, there exist values of θ that minimize the expected squared difference $E[(q_\theta - z)^2]$.

2. **KKT Points**:

   - The optimization problem involves minimizing a convex function (expected squared difference). Given the constraints $0 < \theta$, we can use Lagrangian multipliers to find stationary points. These points should satisfy the KKT conditions, ensuring that they are either minima or saddle points within the feasible region.

3. **Boundedness**:

   - The constraints on θ (i.e., $0 < \theta$) inherently provide a bounded feasible region. This ensures that the solution space does not extend to infinity, which is crucial for well-posedness.

- The function to be minimized, $E[(q_\theta - z)^2]$, is also bounded by the nature of the service times being generated from the given distribution.

4. **No Direction of Unbounded Decrease**:

- Since the objective is to minimize $E[(q_\theta - z)^2]$ and given the bounded feasible region, there are no directions in which the objective function decreases indefinitely.

Conclusion

The given optimization problem is likely well-posed because:

- It has a non-empty feasible region.
- Solutions are expected to satisfy KKT conditions.
- The feasible region is bounded, preventing solutions from extending to infinity.
- The objective function does not have unbounded directions of decrease within the feasible region.

2.2 Description of Program Q2.1 - The Optimal Theta

This optimization approach leverages the SA algorithm to determine the optimal mean service time (θ) for a queueing model characterized as a GI/GI/1 system, where inter-arrival and service times are independent and identically distributed (iid). The algorithm iteratively fine-tunes θ to minimize the objective function, which measures the squared difference between the actual average waiting time and a predefined target waiting time.

The SA algorithm operates as follows:

1. **Initialization**: Begin with an initial estimate for θ.
2. **Iteration**: During each iteration:

- Generate inter-arrival and service times using the exponential distribution with mean = 5 and N = 10.
- Compute the waiting times for customers using a recursive formula.

- Calculate the average waiting time and the objective function value, defined as the squared difference between the 90th percentile of waiting times and the threshold $z = 8$.
- Adjust θ based on the gradient of the objective function.
- Project θ to the interval $[0.01, 100]$ to keep it within a realistic range.
- Check for convergence criteria.

2.3 Results of Q2.1

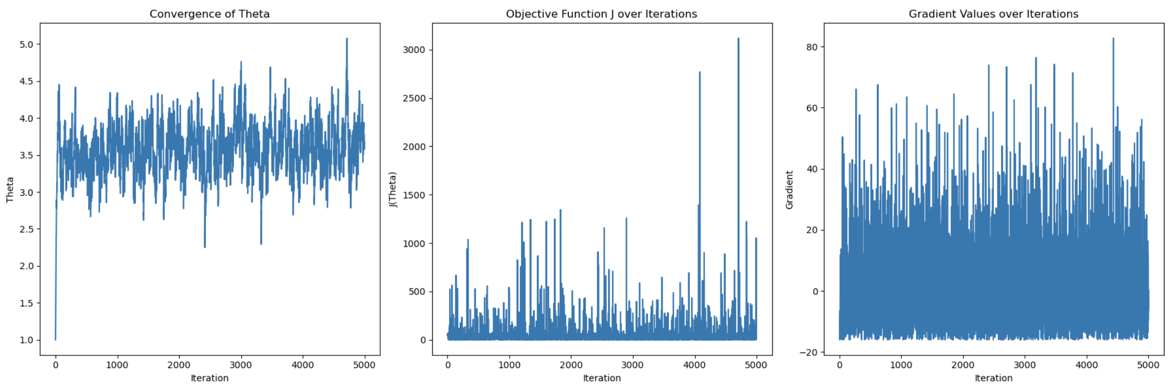


Figure 12: Illustration of investment returns.

2.3.1 Convergence of Theta θ

The convergence plot of theta indicates that the algorithm extensively explores the search space. Initially, the theta values exhibit significant variation, but as the iterations progress, the fluctuations diminish, and the values converge within a specific range. However, the convergence is not completely smooth, indicating the presence of noise and the challenging nature of the optimization environment. The mean value of theta seems to stabilize between 3 and 4, suggesting that this range may be close to the optimal mean service time.

2.3.2 Objective Function J(θ) Iterations

The objective function plot shows the values of J(theta) (i.e., $(q_\theta - z)^2$ over the iterations.) We observe that:

- The values of J(theta) decrease with the iterations, indicating that the algorithm effectively minimizes the objective function.
- The presence of spikes, particularly in later iterations, suggests that the algorithm occasionally explores distant regions of the search space, avoiding local minima.
- Despite the noise, the overall trend shows a reduction in the objective function value, indicating successful minimization.

2.3.3 Gradient Values over Iterations

The gradient values plot illustrates how the gradients of the objective function change over the iterations:

- The gradients display considerable fluctuation. As the iterations progress, the magnitude of the gradients remains substantial, suggesting that the algorithm has not yet converged to a stationary point, which is an issue that requires attention.

2.3.4 Change Gain size ϵ from fixed to decreasing

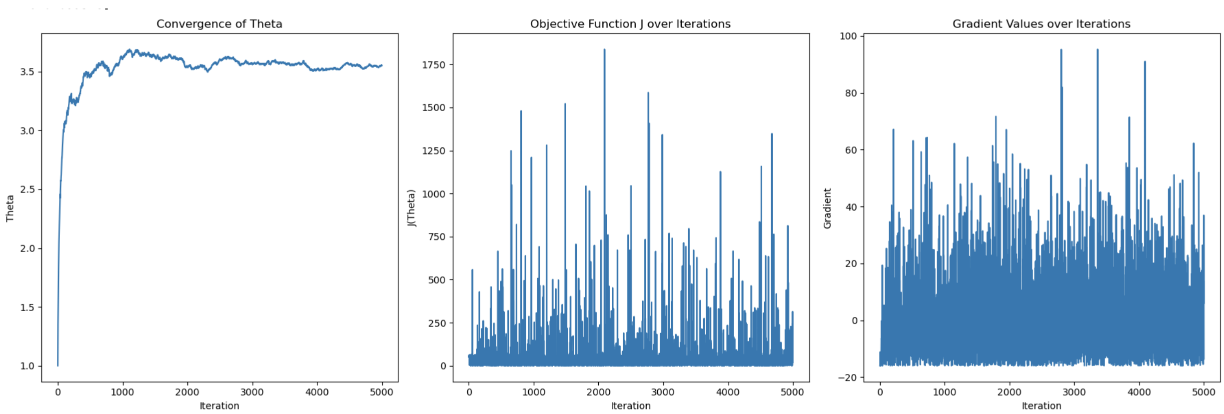


Figure 13: Illustration of investment returns.

- **Changes:** The convergence plot of Theta shows large fluctuations initially, but as the number of iterations increases, the fluctuations gradually decrease and eventually stabilize around 3.5.

- **Explanation:** The reduced epsilon helps the algorithm converge more smoothly, indicating the effectiveness of the exploration in the early stages and the stability in the later stages.

Inspiration: However, the gradient values over iterations do not change. We believe this may be due to the presence of noise, which could be related to $N = 10$. A smaller sample size may introduce more noise, so we will take steps to address the noise in the next phase.

2.3.5 Removing the Noise

We will try to remove the noise and smooth the data using a moving average filter by applying `np.convolve` with a sliding window to reduce random fluctuations and make the overall trend of the data more apparent. Moving average filtering is a widely recognized and effective method for noise reduction in signal processing. This method involves applying a sliding window that computes the average of data points within the window, thereby smoothing out short-term fluctuations and reducing random noise, while preserving the overall trend of the signal. This technique has been proven effective in numerous applications, such as financial data analysis, environmental monitoring, and signal processing, by significantly enhancing the clarity and interpretability of the data.

For instance, Shan et al. (2022) demonstrated the utility of adaptive moving average methods in denoising signals with strong noise backgrounds, highlighting their effectiveness in various applications. Pandey and Giri (2016) confirmed that larger averaging windows result in smoother filtered signals, effectively reducing high-frequency noise. Moreover, practical applications in signal processing have shown that moving average filters perform well for signals with less high-frequency noise, as demonstrated by Guiñón et al. (2007).

Convergence of Θ:

- **Original Plot:** Initially, the values of Theta exhibit large fluctuations and gradually converge.

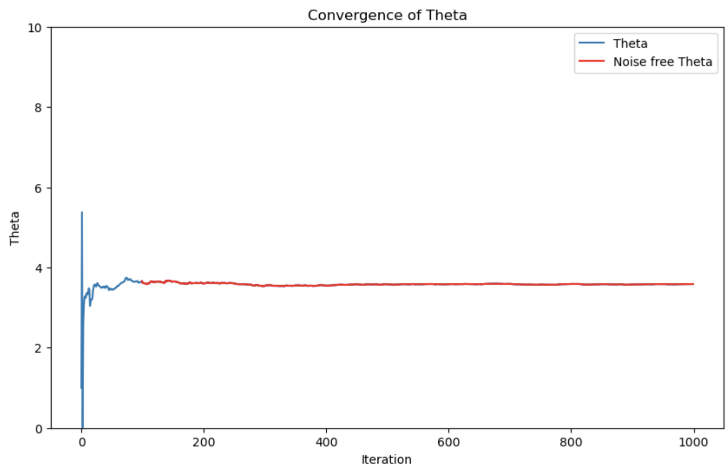


Figure 14: Illustration of investment returns.

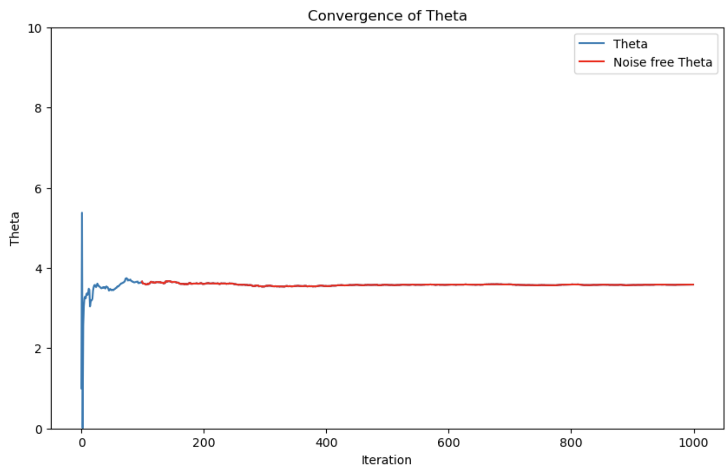


Figure 15: Illustration of investment returns.

- **Smoothed Plot:** After applying the moving average technique, the fluctuations in Theta values are significantly reduced, and the convergence is more stable, ultimately stabilizing around 3.56.
- **Comparison:** The smoothing technique effectively reduces random noise, making the convergence path of the algorithm clearer and more stable.

Iteration of Objective Function $J(\Theta)$:

- **Original Plot:** The values of $J(\Theta)$ decrease gradually with the increase of iterations, but there are multiple spikes, indicating that the algorithm intermittently explores the search space.
- **Smoothed Plot:** The smoothed curve of $J(\Theta)$ is more smooth, with significantly fewer spikes, and the overall trend is more apparent and coherent.

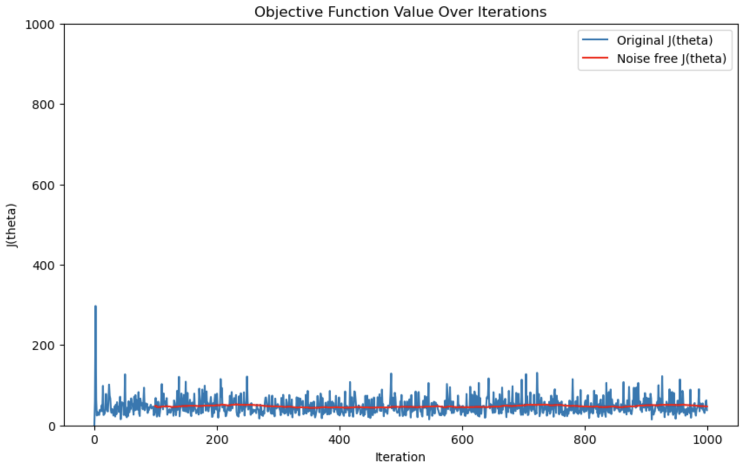


Figure 16: Illustration of investment returns.

- **Comparison:** The smoothing technique reduces high-frequency noise, making the trend of changes in the objective function smoother and easier to observe.

Iteration of Gradient Values:

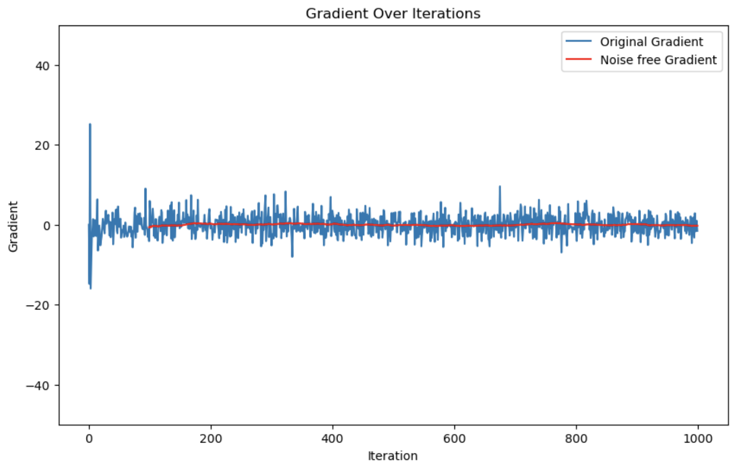


Figure 17: Illustration of investment returns.

- **Original Plot:** Initially, the gradient values show significant fluctuations, and although they gradually decrease with more iterations, there are still substantial variations.
- **Smoothed Plot:** The smoothed gradient value curve exhibits reduced fluctuations and more consistent changes.
- **Comparison:** The smoothing technique effectively diminishes random fluctuations in the gradient values, resulting in a more stable and smooth gradient evolution during the optimization process.

2.4 Validation and Verification for Q2.1

The rationale and theoretical basis for using the 95% confidence interval plot as a verification step are as follows:

Confidence intervals are a method used to estimate the range within which a parameter lies. A 95% confidence interval indicates that in repeated experiments, 95% of the samples will fall within this range. Calculating the confidence interval typically involves computing the sample mean and standard error, then using the formula CI $=$ mean $\pm$ $1.96 \times$ SE)to determine the confidence interval range.

In our example, we first apply a moving average filter to the objective function $J(\theta)$ to smooth the data and reduce random fluctuations. Then, we calculate the mean and standard error for each iteration point and plot the objective function with the 95% confidence interval. By observing that the red mean curve should be centered within the confidence interval, the width of the confidence interval should decrease with more iterations, and most data points should fall within the confidence interval, we can validate the stability and reliability of the results. This verification step ensures that the assessment of the convergence of the objective function $J(\theta)$ is statistically significant and quantifies the uncertainty of the results.

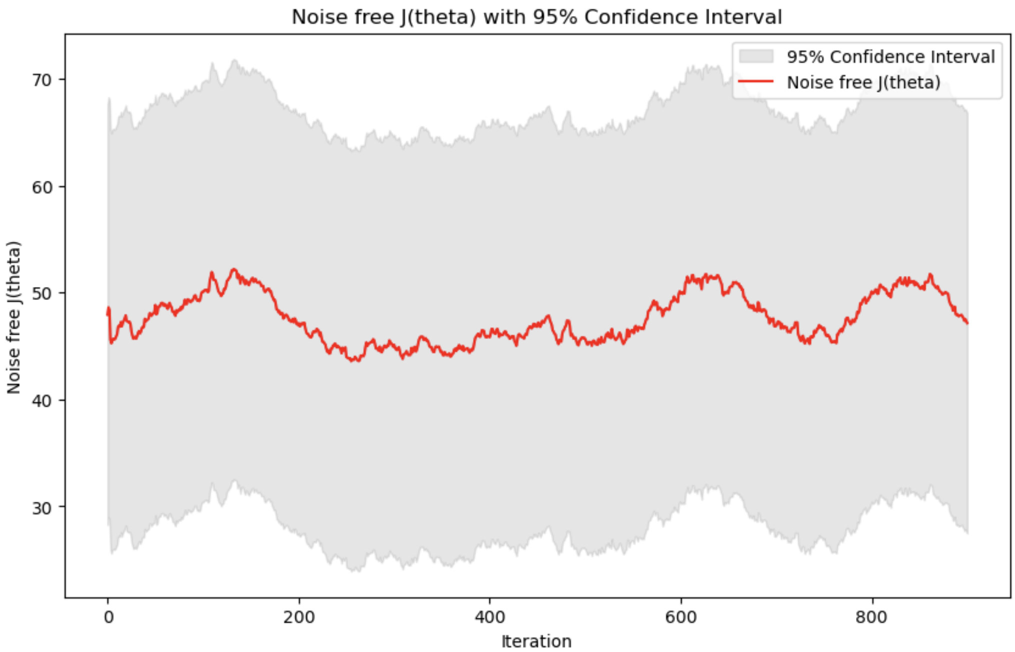


Figure 18: Illustration of investment returns.

2.5 Description of Program Q2.2 - The optimal choice of m and k

The purpose of this experiment is to evaluate the performance of the SA algorithm under different allocations of updates m for generating θ values and k for tracking the quantile $W^N(\theta)$, given a fixed total computational budget.

- m: Number of updates for generating θ values in each iteration.
- k: Number of updates for the sub-problem, tracking the quantile $W^N(\theta)$ at θ_n.

In this problem, we have fixed $m \times k = 10000$, which is a relatively large number. According to the law of large numbers, such a setting provides stable and reliable results because the law of large numbers indicates that as the sample size grows, the sample mean will approach the population mean, reducing the impact of random fluctuations and providing more accurate estimates.

Given $m \times k = 10000$ and m and k being integers, the combinations of m and k are limited to 15 options. Our algorithm explores these 15 combinations to find the optimal one.

Methodology:

1. **Parallel Execution**: The algorithm utilizes parallel processing to efficiently execute multiple instances of the SA algorithm with varying parameters, improving computational efficiency.
2. **Budget Allocation**: It ensures that the total computational budget $m \times k$ equals 10000, providing a fixed upper limit on the number of updates to balance exploration and precision.
3. **Statistical Robustness**: By running the experiments multiple times and averaging the results, the algorithm achieves statistical robustness, minimizing the influence of outliers and random variations.
4. **Confidence Intervals**: The use of 95% confidence intervals around the average objective function values $J(\theta)$ provides a measure of reliability and precision of the results, ensuring that the findings are statistically significant.

2.6 Results for Q2.2

The Plot of Confidence Interval below is our result:

```
m = 10, k = 1000 -> Average J(theta): 47.38351576557673, 95% Confidence Interval: ±17.826162351745428
m = 20, k = 500 -> Average J(theta): 42.12957030660331, 95% Confidence Interval: ±5.549873968060602
m = 50, k = 200 -> Average J(theta): 63.01886432680108, 95% Confidence Interval: ±20.806960334027725
m = 100, k = 100 -> Average J(theta): 46.45425061560718, 95% Confidence Interval: ±8.927182828418763
m = 150, k = 66 -> Average J(theta): 55.62877889296793, 95% Confidence Interval: ±20.853107701067625
m = 200, k = 50 -> Average J(theta): 48.571035438907295, 95% Confidence Interval: ±12.416181110080318
m = 250, k = 40 -> Average J(theta): 50.82119162558236, 95% Confidence Interval: ±9.597714626956423
m = 300, k = 33 -> Average J(theta): 55.6256591874707, 95% Confidence Interval: ±8.373773455986893
m = 350, k = 28 -> Average J(theta): 40.6967736484895, 95% Confidence Interval: ±12.859680807830166
m = 450, k = 22 -> Average J(theta): 42.00837494040701, 95% Confidence Interval: ±5.313622844331255
m = 500, k = 20 -> Average J(theta): 55.19518863837616, 95% Confidence Interval: ±16.25013183200746
m = 1000, k = 10 -> Average J(theta): 62.16055569320681, 95% Confidence Interval: ±16.57916578516589
```

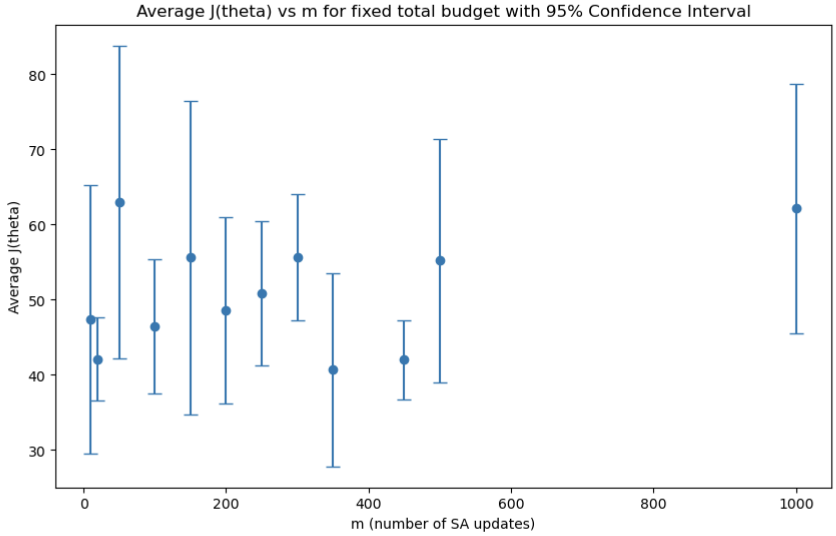


Figure 19: Illustration of investment returns.

Analysis: we observe that the combination $m = 450$ and $k = 22$ provides the most optimal results. Both $m = 20, k = 500$ and $m = 450, k = 22$ exhibit very low average $J(\theta)$ values. However, the combination $m = 450, k = 22$ has a smaller confidence interval, indicating that this $J(\theta)$ value is more reliable.

Additionally, the overall shape of the $J(\theta)$ values shows a trend where values are higher at both ends of the m spectrum and lower in the middle. This pattern suggests that the middle range values for m offer a better trade-off between updates for θ generation and quantile tracking.

Therefore, considering both the low average $J(\theta)$ value and the narrow confidence interval, $m = 450, k = 22$ is the final choice for the most optimal allocation.

Conclusion:

- **Best Combination**: $m = 450, k = 22$
  - **Average** $J(\theta)$: 42.0083

- **95% Confidence Interval**: ±5.3136

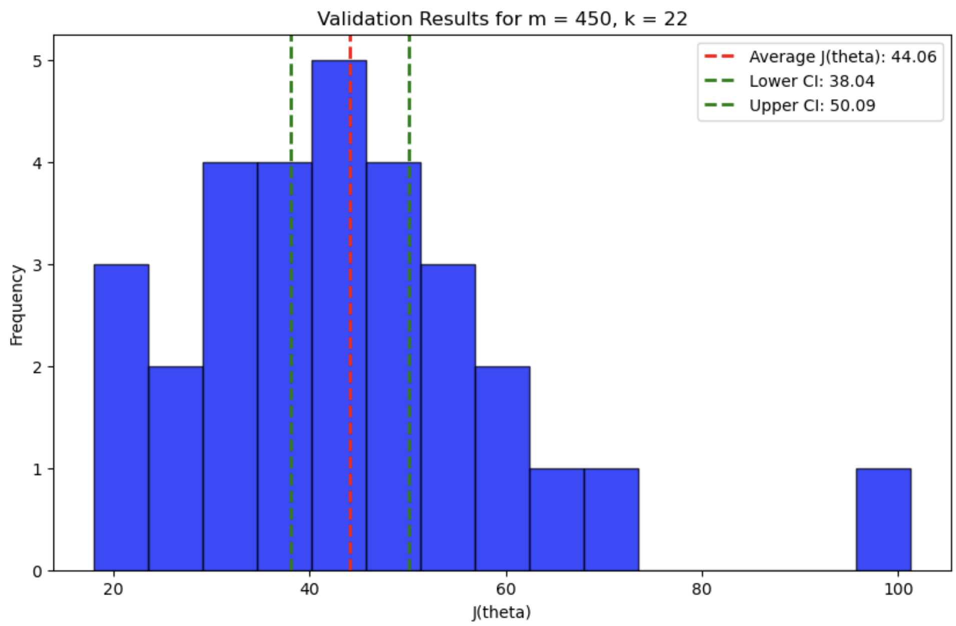
- $m = 20, k = 500$ also performed well with a very low average $J(\theta)$ but had a slightly larger confidence interval compared to $m = 450, k = 22$.
- **Overall Trend**: The graph shows a U-shaped pattern, where the extremes of m have higher $J(\theta)$ values, and the middle range provides more reliable and lower $J(\theta)$ values.

This experiment leads us to conclude that $m = 450, k = 22$ offers the best balance between performance and reliability for the given fixed budget.

2.7 Validation and Verification for Q2.2

We will run the SA algorithm with $m = 450$ and $k = 22$ multiple times, recording the results to plot histograms and 95% confidence intervals. This evaluation aims to assess the stability and reliability of the results, with the histogram showing the distribution of the objective function $J(\theta)$ values from 30 repeated runs of the SA algorithm.

```
Validation results for m = 450, k = 22:
Average J(theta): 44.06472769425999
95% Confidence Interval: ±6.020807926382759
```




```
(44.06472769425999, 6.020807926382759)
```

Figure 20: Illustration of investment returns.

Key Observations

- Average $J(\theta)$: The average value of $J(\theta)$ is approximately 44.06, as indicated by the red dashed line in the histogram. This value represents the central tendency of the objective function over multiple runs, suggesting that the algorithm typically converges to this performance level under the given parameters.

- 95% Confidence Interval: The 95% confidence interval for the average $J(\theta)$ is ± 6.02, providing a range from approximately 38.04 to 50.09. The confidence interval is indicated by the green dashed lines in the histogram. - The relatively narrow confidence interval indicates that the results are quite consistent across different runs, demonstrating high reliability in the algorithm's performance.

- Distribution Shape: The histogram shows an approximately symmetrical distribution around the mean $J(\theta)$, with most values clustering near the mean. This suggests that the algorithm's performance is stable and not significantly affected by outliers. - There are a few outliers at higher $J(\theta)$ values, such as a single instance near 100, indicating occasional performance variations.

2.8 Conclusion for Q2

- We have found the optimal value of $\theta = 3.56$. This value ensures that the 90% quantile of $W_n(\theta)$ is as close as possible to the preset Z value.

- We successfully identified a good combination of m and k values, which allows our SA algorithm to perform optimally within the constraints of a limited overall budget.

3 Appendix: All the code, steps, and output in the Jupyter Notebook

Click here to open the GitHub page to see all code, steps, and output in the Jupyter Notebook

4 Reference

1. Guiñón, J. L., Ortega, E., García-Antón, J., & Pérez-Herranz, V. (2007). Moving average and Savitzki-Golay smoothing filters using Mathcad. *Papers ICEE, 2007*, 1-4.

2. Kushner, H. J., & Yin, G. G. (2003). *Stochastic approximation and recursive algorithms and applications* (Vol. 35). Springer Science & Business Media.

3. Memar Ardestani, M., & Yan, H. (2022). *Noise reduction in human motion-captured signals for computer animation based on B-spline filtering*. *Sensors, 22*(12), 4629.

4. Nemirovski, A., Juditsky, A., Lan, G., & Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization, 19*(4), 1574-1609.

5. Pandey, V., & Giri, V. K. (2016, March). High frequency noise removal from ECG using moving average filters. In *2016 International conference on emerging trends in electrical electronics & sustainable energy systems (ICETEESES)* (pp. 191-195). IEEE.

6. Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics, 22*(3), 400-407.

7. Shan, Z., Yang, J., Sanjuán, M. A., Wu, C., & Liu, H. (2022). A novel adaptive moving average method for signal denoising in strong noise background. *The European Physical Journal Plus, 137*(1), 50.

8. Spall, J. C. (1998). An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest, 19*(4), 482-492.

9. Spall, J. C. (1998). Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems, 34*(3), 817-823.