Using Genetic Algorithm to Optimize Agent Parameters in Stock Market

CMPUT 498

Wang, Tan, Narayanan, Pramanick

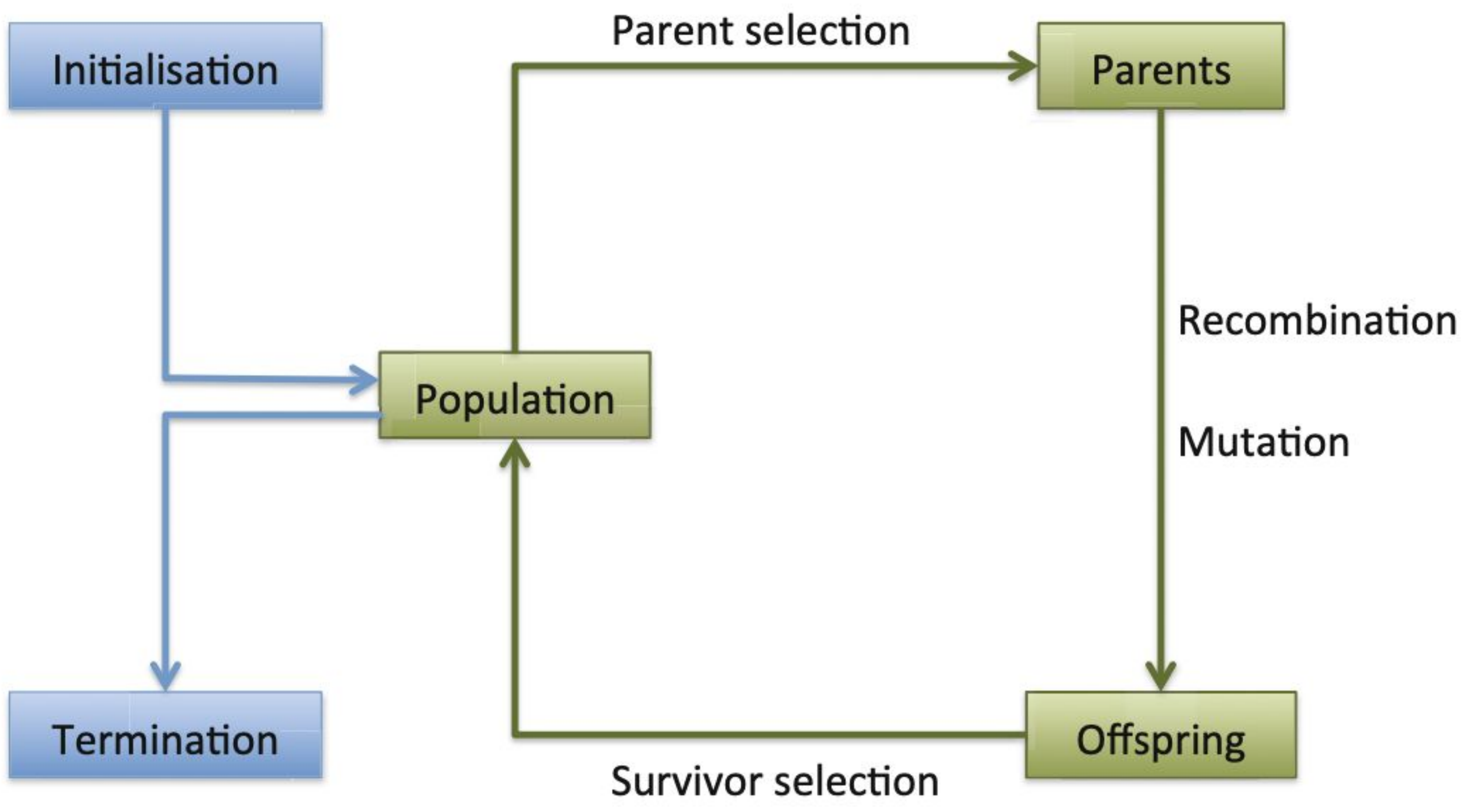
Abstract

- The purpose of this project is to study the effectiveness of genetic algorithms in learning and predicting trends in financial markets.
- We do this by learning which parameters will be most effective in producing a higher portfolio value in each generation.
- Due to the ever-changing nature of financial markets, a genetic algorithm is appropriate because of its ability to adapt and mutate the agent's parameters to fluctuations in the stock market.

Genetic Algorithms

- A method of solving global numerical optimization problems
- An agent embodies a potential solution to the given problem
- It must evolve with every generation, utilizing natural selection and stochastic mutations.

Components of Genetic Programming



Introduction

- Two systems used to model the financial market
  - The first system uses a GA (with mutations of parameters and natural selection)
  - The second system uses stochastic sampling of parameter values
  - The results of both systems are compared with a baseline – Nasdaq 100

Parameters

- The agent has two parameters that define the behavior
  - Risk: the distribution of out portfolio based on market capitalization
  - Diversity : the amount of money invested in each stock
- The agent's policy is determined by the two parameters

Problem Formulation

- State (s) - Represents current configuration of the system. Tuple holding 2 values -
  - Agent's parameter values.
  - Price of each stock i bought by the agent (P_i)
  - Number of shares of each stock i (M_i)

- Action ($\vec{a}$) - Represents the current action an agent can take.
  - $\vec{a}$ defines the amount of money (D_i) invested in each stock i.
  - (D_1, D_2, …, D_n)

Problem Formulation

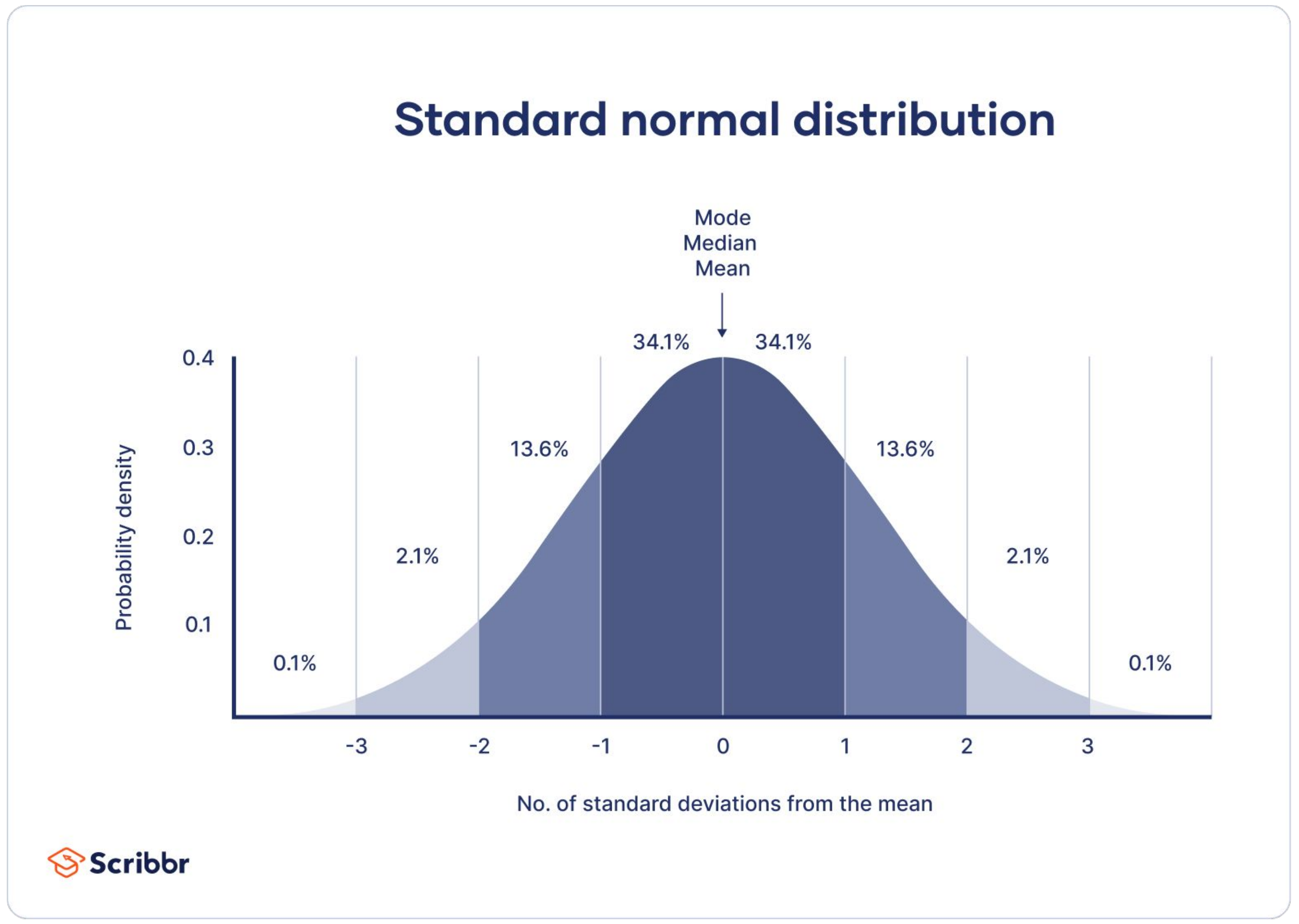
Return (Objective Function)

- Return (for the agent) is the value of the portfolio in any given generation.

$$Return = \overrightarrow{P} \; . \; \overrightarrow{M}$$

$\overrightarrow{P}$ = Vector that holds the stock prices
$\overrightarrow{M}$ = Vector that holds the number of shares held by the agent in each stock

- Normal Gaussian distribution: N ~ (Risk, Diversity)



Related Work

- Black-Scholes Model
  - Pros:
    - Much faster than GA
  - Cons:
    - Models oversimplify the stock market
    - Assume that parameters such as risk remain the same over time

Proposed Approach

Intuition

- We wanted to compare the two systems and determine which system has effective parameters could maximize the return
- Both the parameters of GA and mutation determines the return
- The stochastic sampling system has a result that is similar to the real-world scenario

System 1 - Genetic Algorithm

Details

- Agent starts with random parameter values.
- Parameters are randomly mutated (+/- 0.1) in the second generation, and the agent is allowed to trade stocks for another epoch.
- At the end of this generation, the portfolio values are compared. The parameters and portfolio of the generation with the higher values is passed onto offspring.
- These inherited values are then mutated, and the process is repeated.

Approach 1

Algorithm 1: Approach 1

input : Pandas dataframe NAS_COMP containing the complete 25-year history of all stocks in the NASDAQ Composite

output: Parameters of the agent which performed the best

Create A, which is a 503 dimensional vector, containing the two randomized parameters (Risk aversion and diversification) of the starting agent and its starting portfolio values. The remaining 500 dimensions define the number of shares bought for each stock.

NAS_COMP is the data matrix containing the stock prices over the years.

Sort the elements of NAS_COMP according to the initial market capitalization.

$x \leftarrow A.Risk\_Aversion$

$y \leftarrow A.Diversification$

$Primary \leftarrow x \times 500$

$Diversification \leftarrow Gaussian(NAS\_COMP, mean = primary, variance = y \times 250)$

for *stock in NAS_COMP* **do**

    $agent.buy(quantity = Distribution.z\_score(stock), stock\_to\_buy = stock)$

for *t=0 to t = 20* **do**

    Create new agent using mutation of current agent. New agent will make its own investments given its new parameters. In following time step, compare change in two agents' wealth. Maintain superior agent. Discard inferior agent.

    if $new\_agent.gain > agent.gain$ **then**

        $agent \leftarrow new\_agent$

System 2 - Stochastic Sampling

Details

- Agent starts with random parameter values.
- Agent is allowed to trade stocks for 1 epoch of time.
- In the next iteration, parameter values are stochastically sampled/selected from a list of parameter values.
- This process is repeated for *n* iterations.

Approach 2

Algorithm 2: Approach 2

input : Pandas dataframe NAS_COMP containing the complete 25-year history of all stocks in the NASDAQ Composite

output: Parameters of the agent which performed the best

Create A, which is a 503 dimensional vector, containing the two randomized parameters (Risk aversion and diversification) of the starting agent and its starting portfolio values. The remaining 500 dimensions define the number of shares bought for each stock.

NAS_COMP is the data matrix containing the stock prices over the years.

Sort the elements of NAS_COMP according to the initial market capitalization.

$x \leftarrow A.Risk\_Aversion$

$y \leftarrow A.Diversification$

$Primary \leftarrow x \times 500$

$Diversification \leftarrow Gaussian(NAS\_COMP, mean = primary, variance = y \times 250)$

for *stock in NAS_COMP* **do**

  $agent.buy(quantity = Distribution.z\_score(stock), stock\_to\_buy = stock)$

for *epoch 0 to 50* **do**

  Create new agent. Identify profit made by agent by subtracting final portfolio value after 20 years from the initial, gained from the NAS_COMP matrix. Maintain superior agent. Discard inferior agents.

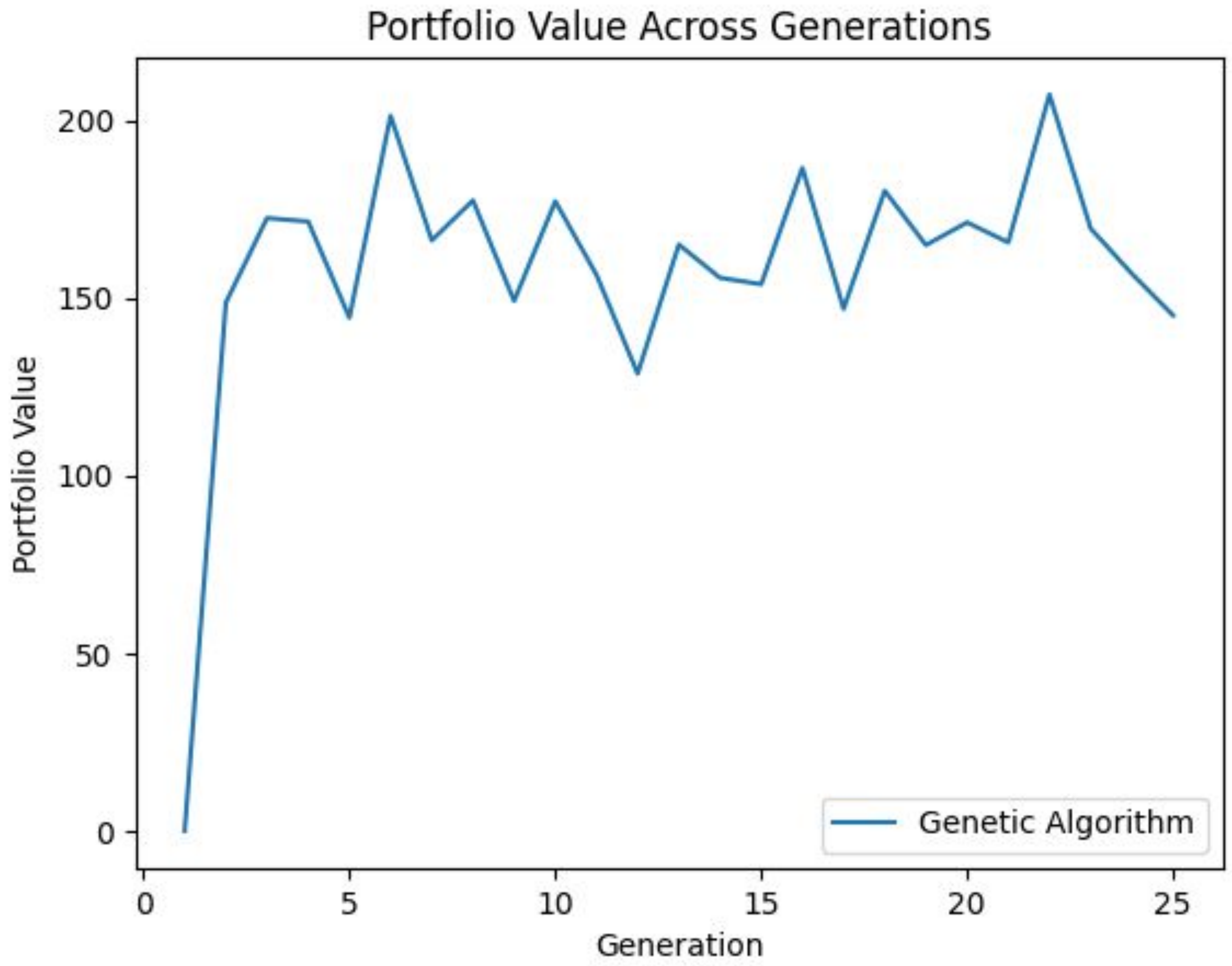
  if *new_agent.profit* $>$ *agent.profit* **then**

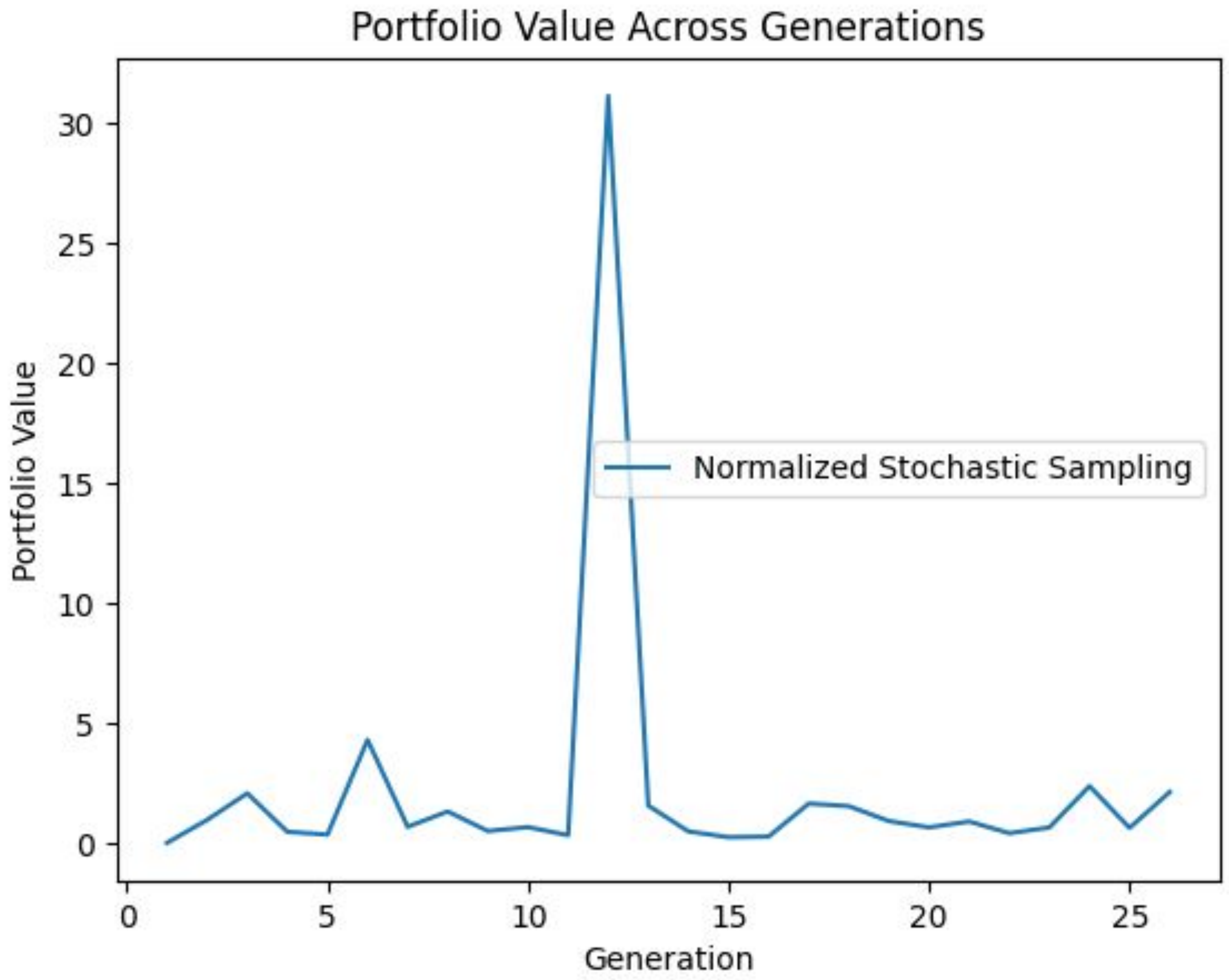
    $agent \leftarrow new\_agent$

Theoretical Analysis

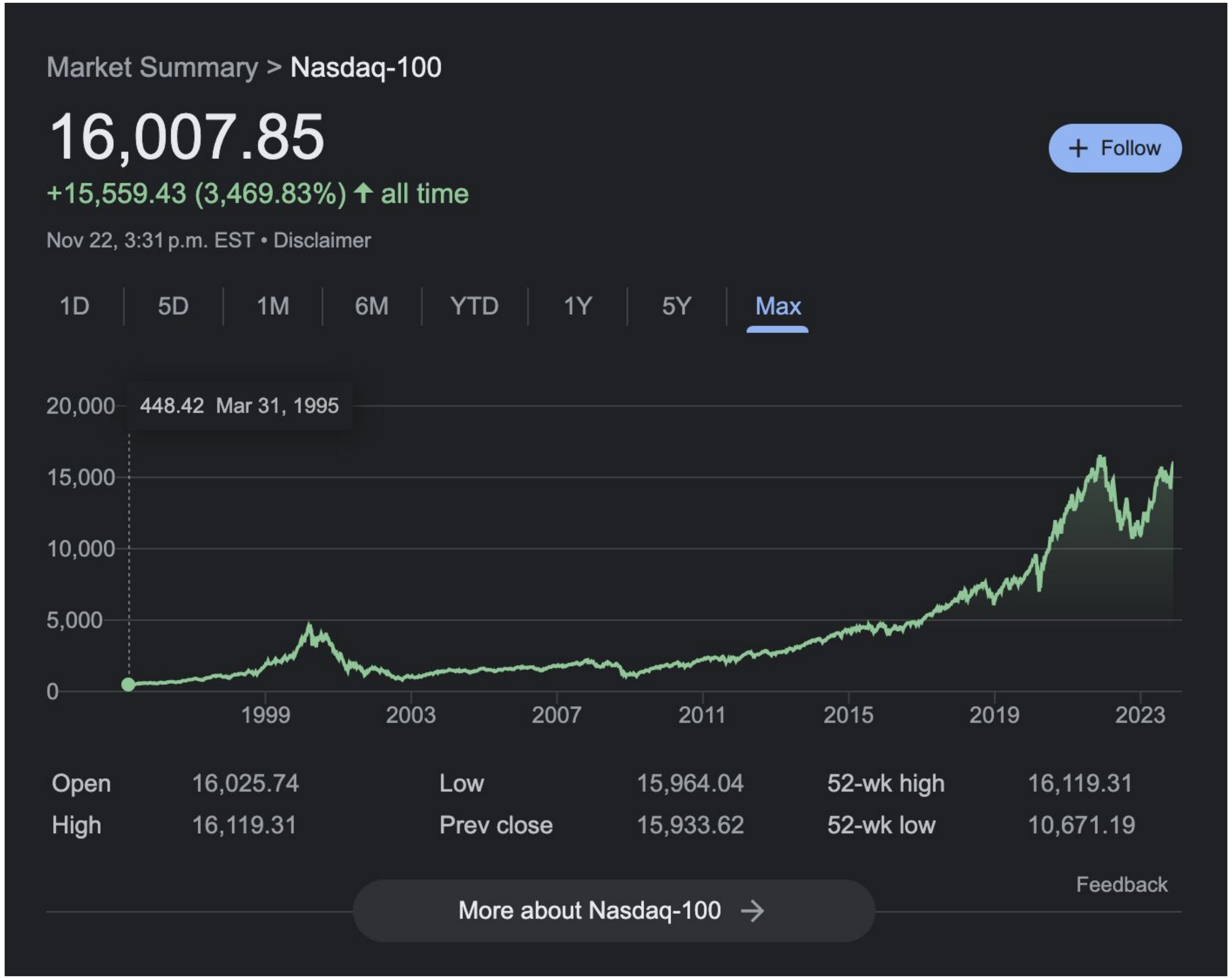
- The GA system will run with O(n) time complexity.
    - n = number of generations
- The stochastic sampling system will run with O(m) time complexity
    - m = number of iterations
- We predict that the GA system will increase the portfolio value over time

Empirical Evaluation





Benchmark



Market Summary > Nasdaq-100
16,007.85
+ Follow
+15,559.43 (3,469.83%) ↑ all time
Nov 22, 3:31 p.m. EST • Disclaimer
1D
5D
1M
6M
YTD
1Y
5Y
Max
448.42 Mar 31, 1995
20,000
15,000
10,000
5,000
0
1999
2003
2007
2011
2015
2019
2023
Open 16,025.74
High 16,119.31
Low 15,964.04
Prev close 15,933.62
52-wk high 16,119.31
52-wk low 10,671.19
Feedback
More about Nasdaq-100 →

Empirical Evaluation

Change in Portfolio Value

GA	Stochastic Sampling	Benchmark
~ 208 * S	~ 32 * S	~ 30 * S

- Start value = S units of currency
- The values here for GA and stochastic Sampling are obtained from the highest portfolio value over all generations.

Empirical Evaluation

Parameters

	Genetic Algorithm	Stochastic Sampling
Average Risk	0.508304	0.873072
Average Diversity	0.338368	0.7153883

Proposed Shortfalls

Approach 1

- We compare portfolio values for the max year between two agents determining which agent gets to continue holding the stocks.

- The problem with this approach is that since the stock market is erratic comparing portfolio value for a small period could give wrong trends and hence might not save the optimal agent.

Proposed Shortfalls

Approach 2

- We are using stochastic random sampling to generate agents, as a result there is no evolution/mutation within our agents that perform well.

- Hence we end up losing important information that could’ve given us a better result as we see in genetic algorithm.

Future Work

- Increase the number of parameters.
- Expand systems to account for multiple agents.
- Combine approach 1 and approach 2.

Conclusion

- The goal of our experiment is to determine the expected value of parameters that will lead to the maximum portfolio value at the end of experiment
- The ideal combination of parameters would be one where both risk and diversity are fairly low.

Citation

1. Black, Fischer, and Myron Scholes. The pricing of options and corporate liabilities.Journal of political economy 81.3 (1973): 637-654.
2. Shinde, A. S., and K. C. Takale. "Study of Black-Scholes model and its applications." *Procedia Engineering* 38 (2012): 270-279.
3. Nasdaq. (n.d.). Stock screener. Retrieved September 2023, from https://www.nasdaq.com/market- activity/stocks/screener