JOURNAL OF NORTHEASTERN UNIVERSITY

Volume 25 Issue 04, 2022 ISSN: 1005-3026 https://dbdxxb.cn/ Original Research Paper

HYBRID REINFORCEMENT LEARNING BASED GA FOR PORTFOLIO OPTIMIZATION

Abdul Khayyum Farooqui

Research Scholar, Department of Computer Science Engineering, Sreenidhi Institute of Science and Technology, Mail ID - abdulkhayyum.519@gmail.com

Abstract

Investors must constantly balance the competing objectives of lowering risks and simultaneously increasing earnings in all markets. American financial economist Harry Markowitz developed the so-called optimal portfolio theory in 1952, taking into consideration the trade-offs between risk and return. In this work, a novel GA based on Q-learning was developed to maximise returns and minimize losses. When compared to a convective GA, the suggested approach is then used to assess the model's efficacy.

1. Introduction

Portfolio optimization is a technique for increasing net returns while decreasing risk in a portfolio. A portfolio is a selection of stocks selected by the investor. Risk is described as the risk of losing all or part of the original investment. Returns are the earnings made when the stock price increases above the original investment[1]. The goal of decreasing risks or gaining greater benefits while retaining the same level of risk may be achieved by applying probability statistics, linear algebra, optimization, and other approaches to rebalance the investment portfolio within the set target returns and risk restrictions. Portfolio optimisation is the process by which investors seek to maximise return while minimising risk. Because the payoff varies with the level of risk taken, investors must find a way to balance the two. The ideal portfolio is defined by the investor's risk and return choices.

The Risk greatly influences the selection of equities for a portfolio. Figure 1 depicts the efficient frontier graph that investors often apply to mitigate these risks[2].

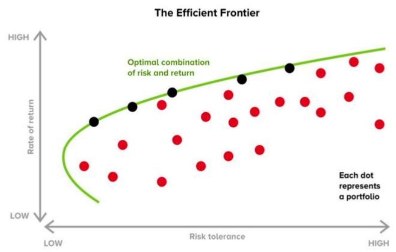

Figure 1: Portfolio optimization

On the x-axis is shown the data's standard deviation, which is sometimes called to as the risk tolerance in common usage. You may see the projected return rate as a percentage along the y-axis. The spots that are perfect are located along the solid line, which is sometimes referred to as the border line. We consider any places that fall outside of the curve to be less desirable than

Submitted: 03/12/2022 **Accepted**: 24/12/2022

those that fall inside. This is because better anticipated returns have been achieved with maintaining the same degree of risk[3].
The goal of portfolio optimization is to increase returns relative to risk taken within the portfolio, making it a helpful tool for investors looking to maximise the risk-return trade-off. Investors may reduce risks and maximize relevant profits while this system is in effect. They might do this with the support of a top-notch portfolio manager, who can assist in selecting the ideal ratio between high-risk and low-risk investment vehicles to achieve the trade-off goals. The projected rate of return and the degree of risk that the investors are willing to bear would always guide the decision[4]. Last but not least, it's crucial to remember that, even though every model and theory has benefits and drawbacks, portfolio managers may maximize the advantages of the portfolio maximizing method if they use the technique diligently.
The advantages of the portfolio optimization are given below:

- utilizing portfolio optimization is one way to increase the potential for a positive return on investment. This may be performed by the use of the efficient frontier graph, which identifies the point at which the risk-return trade-off of the portfolio reaches its highest point and produces the optimal portfolio[5].
- The optimization of the portfolio helps to contribute to the higher degree of variety that the portfolio has. The manager of the portfolio decides to go with a diversified investment strategy so that the underperformance like any one property will not have an influence on the performance of the portfolio as a whole[6].

In order to choose the optimal portfolio for their clients or investors, portfolio managers do a large amount of research into the relevant markets. Because of their research, they are able to see opportunities in the market before other persons do, which is ultimately advantageous for the customers or investors in question.

Contribution

- In this paper, a novel genetic algorithm (GA) based on Q-learning is proposed.
- The proposed model is compared with Convectional GA.

Organization

After this, the remaining parts of a paper are structured as follows:
Section 2 describes Background study. Section 3 explains the proposed methodology. We analyze and discuss the experimental results in Section 4. the paper is concluded in Section 5.

1. Background

Q-learning:

In 1989, Chris Watkins gave the first presentation on Q-learning. In the year 1992, Watkins and Peter Dayan presented a convergence proof. back to the circumstances of the consequence scenario that were experienced earlier in time. back propagates[7].
Q-Learning is a Reinforcement learning approach that, Given the present state of the process, will always choose the most suitable next step to take. This particular competition is selected at random, and the objective is to win the most valuable reward available. Q-learning is a sort

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

of off-policy relevance feedback that does not need a model in order to work. Instead, it simply takes into consideration the agent's present state in order to decide the most productive course of action and then propose it. The next action that has to be taken will be decided by the agent based on its current location within its surroundings and will rely on what it finds. The purpose of the model is to identify[8], given the existing conditions, the action that would be most beneficial to do. It is possible that in order to achieve this objective, it will be necessary for it to devise its very own framework of rules or deviate from the conventional practice[9]. The fact that this is the case demonstrates that there is not really a need for a policy, which is why we refer to it as off-policy. The use of predictions about the expected response of the environment by the agent in order to make judgments is what is meant when we talk about model-free decision making. People are more likely to learn from their mistakes and experiences than they are from receiving rewards[10].

Imagine a computational being that moves through a finite and discrete world by selecting single action at each time step in the process of doing so. A Markov process is under the agent's control. The agent may take into account the e_j $(\in E)$ state of the world and decide on an action $c_j(\in \alpha)^1$, at step *j*. The law guarantees that the agent will get a probabilistic compensation u_j whose mean value $X_{e_j}(c_j)$ is principally dependent on state as well as action, and that the world will change probabilistically to f_j.

$\text{Prob}[f_j = f[e_j, c_j] = S_{u_j}[c_j]$ (1)

The agent must choose a policy that maximizes discounted anticipated return. Discounted rewards are worth $\gamma^y (0<\gamma<1)$ less than rewards earned presently. With respect to a given policy π, the worth of a given state x is

$$Z^{\pi}(e) = X_e(\pi(e)) + \gamma \sum_f S_{ef}[\pi(e)] Z^{\pi}(f) \quad (2)$$

because the agency expects to get $X_e(\pi(e))$ quickly for carrying out the action prescribed by π, and so proceeds to a state 'worth' $Z^{\pi}(f)$ to it, given probability $S_{ef}[\pi(e)]$ According to π DP theory, there is at minimum one optimal stationary policy π^* that

$$Z^*(x) \equiv Z^{\pi^*}(e) = \max_c \left\{ X_e(c) + \gamma \sum_f S_{ef}[c] Z^{\pi^*}(f) \right\} \quad (3)$$

from state x. DP presents many ways for estimating Z^* and one π^*, assuming $X_e(c)$ and $S_{ef}[c]$. Q learners must determine a π^* without knowing these values. Traditional approaches (e.g., Sato, Abe & Takeda, 1988) for learning $X_e(c)$ as well as $S_{ef}[c]$ while executing DP assume certainty equivalence, i.e., expensive in the early phases of learning because decisions are made as if the current model is valid (Barto & Singh, 1990). Because it finds the best policy step-by-step, Watkins (1989) calls Q-learning incremental dynamic programming.

Policy π defines Q action-values as:

$$T^{\pi}(e, c) = X_e(c) + \gamma \sum_f S_{ef}[\pi(e)] Z^e(f) \quad (4)$$

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

Q-learning aims to anticipate the T value of the optimal policy values by discounting the expected payoff for an action at a state and a policy subsequently. $T^{*}(e, c) \equiv T^{\pi^{*}}(e, c), \forall e, c$ for convenience. It is easy to establish that $Z^{*}(e) = \max_{c} T^{*}(e, c)$ as well as that if c^{*} is an optimum policy action that optimizes and formed as $\pi^{*}(e) \equiv c^{*}$. If an agent learns the T- values, it can readily choose the best action. There may be several optimum policies or c^{*}, but T^{*} values are unique.
Q-learning involves a series of events, stages or episodes. In the j^{th} episode, the agent:
Observes Current State of e_j.
Selects and performs action c_j.
Observes the every next state f_j.
Gets an immediate payoff u_j and
adjusts its T_{j-1} values using a learning factor α_j, according to:

$$T_j(e,c) = \begin{cases} (1-\alpha_j)T_{j-1}(e, c) + \alpha_j[u_j + \gamma Z_{j-1}(f_j)] \; if \; e = e_j \; and \; c = c_j, \\ T_{j-1}(e, c) \end{cases} \qquad (5)$$

Otherwise, Where

$$Z_{j-1}(f) \equiv \max_{d} \{T_{j-1}(f, d)\} \qquad (6)$$

from state *f*. At first, *T* values may not be an accurate reflection of the policies they imply (the maximizing actions in equation 2). The initial *T* values, $T_0(e, c)$ is assumed for all states and actions.

Markowitz

The Markowitz Model Parameters, Determined Using Historical Stock Price Information
Rate of Return
The amount of an investment's gains to its losses, represented as a percentage, is referred to as the rate of interest[11]. When discussing investments, the phrase "cost basis" means the initial sum of money that is invested.
It is possible to calculate an investor's expected rate of return by making use of both the existing data and estimates for potential new investors. In most cases, percentages are used in order to represent the outcomes of the rate of return.
The formula below is being used to calculate the arithmetic returns vov_{te} on an asset investment made between time te and time te and $te-1$.

$$vov_{te} = \frac{Sv_{te} - Sv_{te-1}}{Sv_{te-1}} \qquad (7)$$

Where Sv_{te} is the stock price at time, as well as we assume for the time being that these stock pays dividend, and Sv_{te-1} is the price in time $te-1$.

Expected Return

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

Let's imagine we have "*x*" asset. The anticipated return on asset a, $a = 1, \ldots\ldots, x$ is then calculated using the equation below.

$$\mu_a = F\left(n^a\right) = \frac{\sum_{te=1}^{qr} n_{te}^a}{qr} \qquad (8)$$

Where qr is the number of times that we have calculated the return on asset *a*, and n_{te}^a is the return on asset *a* between periods te and *te*-1, *te* =1, qr, and is the return on asset *a* between those two periods.

Variance and Standard Deviation

To get the variance of asset "*a*" the following formula is used to the data.

$$\sigma_a^2 = Var\left(n^a\right) = \frac{\sum_{te=1}^{qr}\left(n_{te}^a - \mu_a\right)^2}{qr-1} \qquad (9)$$

The confidence interval is frequently used by investors in order to determine the level of risk that is associated with their investment. The standard deviation is the statistical measure of risk that is used the most often since it demonstrates the degree to which actual values may deviate from their predicted values. The risk rises in proportion to the increase in the standard deviation, and vice versa. The formula for calculating the standard deviation looks like this:

$$\sigma_a = \sqrt{\sigma_a^2} = \sqrt{\frac{\sum_{te=1}^{qr}\left(n_{te}^a - \mu_a\right)^2}{te-1}} \qquad (10)$$

Covariance

When we engage with assets, the symbol $\Omega_{n\times n}$ that corresponds to the numerous risks that are mapped out in the return covariance matrix is represented by the symbol. This matrix takes into account covariances between all possible pairings of assets and other items, in addition to variances in both its major and minor diagonals.

$$\Omega_{n\times n} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{pmatrix}$$

Where

$$\sigma_{ab} = Cvc\left(n^a, n^b\right) = \frac{\sum_{te=1}^{qr}\left(n_{te}^a - \mu_a\right)\left(n_{te}^b - \mu_b\right)}{qr} \qquad (11)$$

Markowitz Model

Before Harry Markowitz released the current portfolio theory in his work "Portfolio Selection" in 1952, investors focused mostly on assessing the risk and return of individual assets when creating investment portfolios[11]. This continued until Markowitz published the theory. When deciding which assets to include in the investment portfolio, we prioritized those that provided the highest return with the fewest potential drawbacks. Diversification is another one of

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

Markowitz's investment strategies, and it entails constructing a portfolio after determining the total risk of the portfolio, or a selectable portfolio as a whole as opposed to a portfolio consisting of individual companies that have already been preselected. Therefore, rather than concentrating on the actual attributes, the emphasis is placed on the assets, which are the relationships that exist between the characteristics.

Harry Markowitz was recognized in 1990 with the Nobel Prize in Economics, together with Morton Miller and William Sharpe, for the contributions he made to the field of portfolio theory. It is based on the basic notion that the returns on investments over time are random variables. This allows for the computation of mathematical expectations as well as standard deviations, the latter of which is considered to be a proxy for investment risk. It is possible to demonstrate that the rate of return on portfolio F(*nr*) is just the total of the anticipated returns on the individual assets that make up the portfolio, taking into account a variety of factors, such as the proportional distribution of the assets within the portfolio.

The nonlinear standard deviations and covariances of return on individual assets are employed in the calculation of the standard deviation σ_s, which is used to assess investment risk. The idea of diversification as proposed by Markowitz is hindered by the observation that the proportional increase in the number of covariances that takes place whenever the number of assets in a portfolio expands is also present[12]. As a consequence of this, the risk associated with the portfolio will be determined more by the covariance between assets than it would be by the risk associated with the assets individually. The mathematical statement that follows corresponds to the apparent form of parametric optimization that is shown by the Markowitz model.

$$\left|\begin{array}{l} \max F(nrn) = \max \sum_{a=1}^{x} \omega_a \mu_a \\ \min \sigma_{sv} = \min \sqrt{\sum_{a=1}^{x} \sum_{b=1}^{x} \omega_a \omega_b \sigma_{ab}} \\ 0 \le \omega_a \le 1, \quad a = 1, \dots, x \\ \sum_{a=1}^{x} \omega_a = 1 \end{array}\right. \qquad (12)$$

Where ω_a represents the proportion of capital that will be invested in asset *a*, n^a stands for the asset's return, μ_a stands for the asset's anticipated return, μ_{ab} stands for the covariance between the returns of assets a and b, $F(nrn)$ stands for the stock's expected return, σ_S stands for the risk of the portfolio.

This model is simple enough for theoretical analysis as well as numerical solutions, and it can account for the vast majority of the real-world conditions that are encountered. This model is still referred to by the name of its creator, Markowitz, despite the fact that it can also be expressed as a Mean-Variance model. The Markowitz model is founded on a number of assumptions and presumptions regarding the behavior of investors and financial markets.

Investors are able to make an accurate prediction on the potential return distributions for a given holding time.

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

Single-period utility is the kind of utility that entrepreneurs focus on, with the goal being to maximize their utility in light of the declining marginal usefulness of wealth.
Investors consider the range of possible returns on an investment to be a useful proxy for the level of risk associated with the endeavor.
Investors are only concerned with two metrics: the portfolio's average value and its average deviation over a predetermined period of time.
As a measurement of both return and risk, investors look to the expected value of a probability returns distribution in addition to the variance of that distribution.
Gain is preferable, but one should try to prevent loss at all costs.

Genetic Algorithm

An optimization method inspired by the principles of heredity and natural selection is the Genetic Algorithm, or GA. It is sometimes abbreviated as GA. It is extensively used to find optimal or near-optimal solutions to difficult problems, the resolution of which would ordinarily take the work of a whole lifetime[13]. In addition to being used for the aim of resolving optimization-related challenges, it is also often employed in the research and machine learning domains.

Fitness Function

Each parameter's value is input into the fitness function, which then outputs a single number indicating how effective the solution is.
Based on the categorization method, a preliminary collection is created. According to this interpretation, the partition of the space is made up of relations between the characteristics that are indistinguishable from one another. Given a decision table $S=(U,R,V,f)$, U is a finite collection, or a domain, and R is an attribute set of elements both the condition attribute set C and the decision attribute set D.
Each attribute subset $A\subseteq R$ corresponds to a class in domain U, as well as the indiscernible binary connection $IND(A)$ between those classes is defined as follows:

$$IND(A)=\{(x,y)\in U^2,\forall a\in A(a(x)=a(y))\}$$

$$U|IND(A)=\{X|X\subseteq U\wedge\forall x\in X\forall y\in X\forall a\in A(a(x)=a(y))\}$$

According to the decision table S, the lower and upper approximation sets of X are computed as follows for every subset in and comparable class for attribute subset in U.

$$A_{-}(X)=U\{Y|Y\in U|IND(A)\wedge Y\subseteq X\}$$

$$A^{-}(X)=U\{Y|Y\in U|IND(A)\wedge Y\cap X\neq\varnothing\}$$

Finding the optimal breakpoints entails minimising the number the breakpoints while safeguarding the undecidable relations in the decision table, hence these two metrics should be used to create the fitness function:

$$Fitness(D)=\frac{(N_t-N_D)}{N_I}\times R_D$$

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

where N_D is the number of breakpoints, $\Delta N = N_I - N_D$ is the change in the number of breakpoints, and N_I is equal to the number of points that made up the first breakpoint set. The values of R_D are 0 and 1, respectively, indicating whether or not the discretization affects the decision table's indiscernibility.

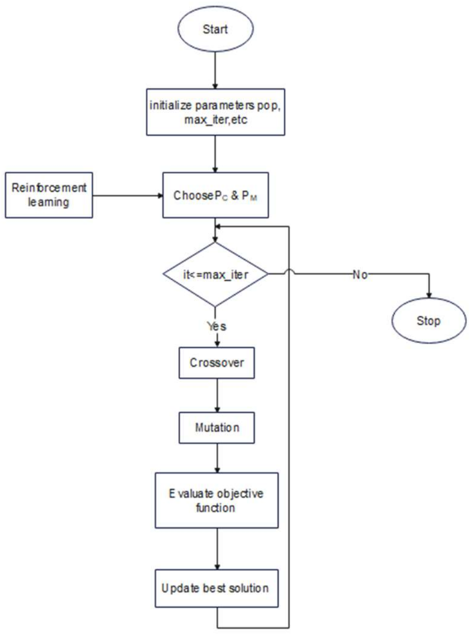

Figure 2 GA flowchart

The steps of a genetic algorithm look like this:

Step 1: Find the value for the generation, mutation, and crossover rates as well as the number of chromosomes.

Step 2: Make up a random number for the population's chromosome pairings and use it as the starting point for the genes' chromosome pairings.

Step 3: To get the appropriate number of generations, repeat steps 4–7.

Step 4: Analysis of chromosomal utility for fitness purposes

Step 5: Chromosomes selection

Step 6: Crossover

Step 7: Mutation

Step 8: Solution (Best Chromosomes)

The flowchart of algorithm can be seen

2. Methodology

Reinforcement Learning-Based Genetic Algorithm

A genetic algorithm is a method that can be applied to any problem in the field of artificial intelligence. To dynamically alter the population's diversity, we implemented a method of regulate the approximate sets-based fitness value[14]. We also used a reinforcement learning method to choose the cross segments as well as point mutations in a discretization scheme that

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

can be adjusted for various multidimensional data types. This significantly increased accuracy and accelerated convergence. The algorithm's flow was detailed below.

Discretization Scheme Evolution to Be Optimised

The best member of the population was kept around thanks to the global variable, while the best member of the discretization scheme was kept around due to the local variable. During each cycle, the population's fitness is calculated, the best member is identified, the global variable was updated, its discretization scheme is improved, cross- as well as The top performer in a population carries out the learning operations based on mutation, changing the local variable. The population continued to engage in regular evolutionary processes if the termination criterion was not met. Otherwise, the person with the highest The output is the fitness level calculated by comparing between global and local variables.

Operator Selection Based on Control Function

The examination of each person's fitness within the population serves as the foundation for the selection process. Higher fitness levels often increase the likelihood of selection. Roulette is a straightforward, effective, and selection method based on the laws of probability. The chance that person *i* is chosen is P(i) where *n* is the number of individuals and f_i, is the fitness value of *i*.

$$P_i = \frac{f_i}{\sum_{j=1}^{n} f_i}$$

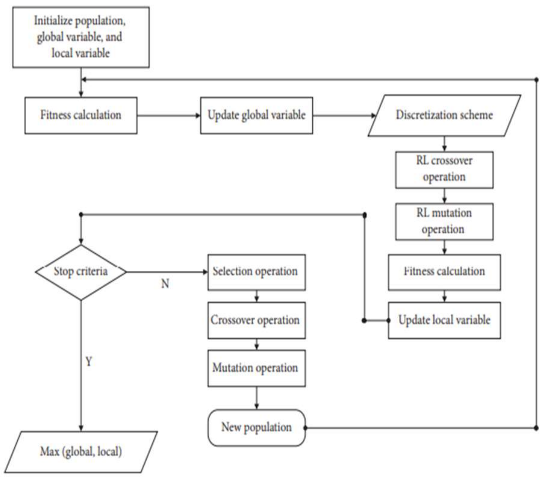

Figure 3 Evolution of the discretization scheme to be optimized.

Although fit people are even more likely to be selected, this reduces population diversity. Individuals having a fitness value of 0 are not selected during population evolution. R_D only accepts values of 0 or 1. Although initially not viable choices, these people could be quite near to the ideal possible answer and ultimately turn into it. However, feasible solutions with $R_D = 1$ and possible viable solutions with $R_D = 0$ tend to be quantitatively identical when taking into account the randomness and evenness of the original population. Due to the intricate interaction

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

between the characteristics of multidimensional data, even with a small population, $R_D = 0$ offers far more probable workable solutions than $R_D = 1$. Because the bulk of the population's plausible solutions were rejected when roulette was employed for individual selection, the population's variety was decimated.

$$\varphi(x) = \left[\frac{(N_I - N_D^x)}{N_I} \right]^z ,$$

According to the analysis that was mentioned earlier, We increased the fitness function by layering the controller on top of the real basis, as well as we determined the governing variables that in order to maintain the population's diversity, depend on the percentage of effective solutions present at any given time during the evolutionary process. during the early phase of development and speeding people's tendencies to converge on a single best answer as we near the end of development. When this happens, its X chromosome's fitness value increases to Fitness.

$$(x) = \frac{(N_I - N_D^x)}{N_I} \times R_D^x + \sigma(p) \times \varphi(x).$$

The control function has two parts: the control item, $\varphi(x)$, and the control factor, , $\sigma(p)$ where $\sigma(p)$ is the fraction of solutions in the population that are feasible and have an $R_D = 0$, As well as $\sigma(p)$ is a function of p . Both $\varphi(x)$ as well as $\sigma(p)$ may be expressed as

$$\sigma(p) = \begin{cases} 0, & \mu = 1 \\ \mu^{1-p} & 0 < \mu < 1 \\ \left[\dfrac{1}{2k \times \log_2 N_1} \right] & \mu = 0 \end{cases}$$

$$\mu = \frac{\sum_{i=1}^{k} (setcmp(C\_(d_i)) + setcmp(C^{-}(d_i)', C^{-}(d_i)))}{2k}$$

where N_I is the number of breakpoints with in initial breakpoint set, N_D^x is the number of breakpoints procured after chromosome x decoding, R_D^x seems to be the change of a decision table's indiscernible relationship after becoming discretized by chromosome x, k is the number of classes, d_i is the ith class, $C\_(d_i)$ as well as $C^{-}(d_i)$ are the lower as well as higher approximation sets of d_i before discretization, as well as If $setcmp(setA, setB)$ are equal, then perhaps the method returns 1, else it returns 0. Considering that each category has its own upper and lower approximation set, the possible values for M are all $0 \le \mu \le 1$. These observations pertain to the enlarged fitness function:

- The situation where fitness value of a feasible solution using $R_D = 1$ will be less than or equivalent to that of the potential viable solution with, $R_D = 0$ is one that we want to

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

avoid. meaning that there exists a circumstance where there are as many thresholds between the two as there are between the two variables. For this reason, we insist that z, the index of the variable under control, fulfil z>1. Here, z equals 2 for our purposes.

- As can be seen, the value range of μ is really $2K+1$ distinct points with an interval of $1/2k$, ranging from 0 to 1. The decision table's indiscernibility is lost after discretization when $\mu = 0$. The value of the control function is less than the minimum nonnegative value because $0 \le \mu \le 1, \mu = 1/2K$, For conceivably possible solutions with $R_D = 0$, the need of providing a control function is nullified if $\mu = 0$ .In order to reflect this, we altered the equation μ^{1-p} to $(1/(2k \times \log_2 N_1))^{1-p}$. Therefore, the control function has a non-zero value at $\mu = 0$, and this value is less than the value at $\mu = 1/2K$. Having "2" as that of the base number and N1 as the actual number makes possible to limit the difference between $\mu = 0$ and $\mu = 1/2K$ control function values to an acceptable level, thereby regulating the selection of persons extremely effectively even though the chromosome is binary coded.
- When $\mu = 1$, $R_D = 1$, and $\sigma(p) = 0$, this person is a possible solution, and the control item's value is 0. When $0 \le \mu \le 1$ and $0 \le \mu \le 1$, and when p is the independent variable and $\sigma(p)$ is an increasing function. It is important to remember that p is big as there are few or even no solutions with in population during the first phases of evolution. At this point, $\sigma(p)$ should be pretty big, so that in rouletteThere is a good possibility of picking a suitable answer, and also the search is progressively narrowed down to that space. p decreases because the population gradually becomes dominated by better and better solutions as evolution proceeds. Thus, as p decreases, $\sigma(p)$ should decrease, allowing us to move more quickly from practical to optimal solutions.

Crossover Operator Based on Q-Learning

In a genetic algorithm, the crossover operation is carried out by moving certain genes from one set of matched chromosomes to another set of chromosomes in order to produce two children. Nevertheless, multidimensional data consist of numerous characteristics, and cross fragments have a propensity to concentrate on aspects that have huge value intervals. As a result, the breakpoints on the other features no longer have the opportunity to be crossed. In spite of this, there were complex interrelationships between the features. The high chance of destruction of certain high-quality pieces was due to the fact Since without any kind of background information, improving the crossing operations of a needed discretization scheme is like flying blind. To accomplish this, we used Q-deliberative learning's capabilities to choose characteristics of a settings with different to optimise during each cross operation cycle.

State: In accordance with the analysis that was presented before, the crossover procedure assigned a particular likelihood of variation to each characteristic. The state that is characterized by the shifting collection of characteristics that occurred throughout the crossover process. Due to the fact that there were N features in the multi-dimensional data,

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

each state in the search space is a mix of multiple of the attributes, and the space then partitioned in 2N 1 states. For the case when N = 3, there are seven distinct possibilities. Assuming that there are N features in the multidimensional data, the search space may be broken down into 2N 1 states, each of which is a unique permutation of the features. There are seven possible states when N = 3, for instance: $\{f_1\}$, $\{f_2\}$, $\{f_3\}$, $\{f_1, f_2\}$, $\{f_1, f_3\}$, $\{f_2, f_3\}$, as well as $\{f_1, f_2, f_3\}$, where f_i is the i^{th} feature of multidimensional data, $1 \le i \le N$, as well as the elements in $\{*\}$ relate to characteristics at the sites of most recent crossover and mutation.

Action: According to the concept of states that was presented earlier, the number of states expanded exponentially if the data set in question is multidimensional. In general, multidimensional data included a significant number of characteristics. One action was required whenever there was a change from one state to the next; hence, there were numerous actions that needed to be specified, which increased the complexity of the calculation. According to the findings of the prior investigation, In particular, we set out to develop a way to prevent the situation where cross segments prioritised features with a wider value range throughout each crossover operation. This made it so certain thresholds could no longer be exceeded. Additionally, numerous high-quality passages were removed without any prior knowledge that serve as guidance. As a consequence, three distinct factors must be considered in relation to the current state when deciding which state to transition to after completing an activity. To begin, the features of one state are included inside the features of the previous state. Second, Extracted features from the subsequent state fill a gap in the collection of features for the current state. Third, the features of a present state and the previous state do not overlap. Thus, we may categorise actions into three groups (G, H, and I) with corresponding letters. Once the algorithm has completed an operation on the current state, it will go to the next state and perform a cross operation on all of the attributes of that state.

Imagine that the current state is S_t and the future state is S_{t+1}. The following examples illustrate how $G(S_t)$ represents a random jump to any subset of the current state, $G(S_t)$ represents a random jump to any subset of the complementary set of a current state, as well as $H(S_t)$ is not blank and does not represent a subset of a current state, then it denotes a random hop to that set..

$$S_{t+1} = G(S_t),$$

$$S_{t+1} \in \{S \mid S_t \subseteq S_t \wedge S \neq \varnothing\},$$

$$S_{t+1} = H(S_t), \qquad S_{t+1} \in \{S \mid S_t \cap S_t = \varnothing \wedge S \neq \varnothing\},$$

$$S_{t+1} = I(S_t), \qquad S_{t+1} \in \{S \mid S \not\subset S_t \wedge S \neq S_t \wedge S \cap S_t \neq \varnothing\},$$

It is easy to see that the range of values for $G(S_t), H(S_t)$, As well as $I(S_t)$ covers all of the states

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

Reward: More can be accomplished, but only if you carefully choose the right feature set for each crossover operation. To get to the best solution as quickly as feasible, we assign a reward value to each conceivable state-action pairing. A algorithm's search for such optimal solution is evaluated in large part via the changes in individual performance, which is the basis for the reward value.

When we give each possible combination of states and actions a reward, we may fast approach the ideal solution. In order to evaluate how well the algorithm is doing its job of finding the best possible solution, the reward is calculated as the percentage increase or decrease in each individual's fitness. In the formula mentioned below, $P(S_t \to S_{t+1} \mid A_t)$ is the probability of jumping to the next state S_{t+1} after performing action A_t in the current state S_t. This is related to the number N_{state} of all of the possible states to jump to, which is $1/N_{state}$. fit(S_t) and fit (S_{t+1}) are, respectively, the present and future state's individual fitness, and 1best is the scheme's best fitness in the past that has to be enhanced.

$$\mathrm{Re}ward=\begin{cases}5\times P\left(S_t\to S_{t+1}\middle|A_t\right), & fit\left(S_{t+1}\right)>lbest,\\ 1\times P\left(S_t\to S_{t+1}\middle|A_t\right), & fit\left(S_t\right)<fit\left(S_{t+1}\right)\le lbest,\\ 0, & fit\left(S_{t+1}\right)=fit\left(S_t\right),\\ -1\times P\left(S_t\to S_{t+1}\middle|A_t\right), & fit\left(S_{t+1}\right)<fit\left(S_t\right).\end{cases} \tag{11}$$

If the optimal discretization scheme contains a cross operation, we may update Q in accordance with the planned reward value.

Mutation Operator Based on Q-Learning

Similar to how we used Q-decision-making learning's skills for such crossover operator, we used them for the mutation operation to choose attributes of a discretization scheme to optimise at each iteration. It was done this this in the hopes of improving the operation's success rate. The only change is that you'll be doing a mutation operation on the features instead of a cross one. Mutation and cross operations provide the same result (in terms of state, action, and reward) and are hence equivalent. Each person maintains their own, individual Q-table. The action of modifying a discretization scheme's two Q-tables in order to make use of the scheme's three features simultaneously.

The value 0 is used to start off both Q-tables. Assuming that the current state is $\{f_1, f_3\}$, action H is selected for the cross operation. Accordingly, the state jumps to $\{f_2\}$, as well as Q's value of 1 is updated to 6 in cross operation. Then, state $\{f_2\}$, selects action $\{f_2, f_3\}$, for the mutation operation. Accordingly, the state randomly jumps to $\{f_2, f_3\}$, as well as The mutation process increases Q′ from its initial value of 2 to 3. That manner, we can simultaneously update both Q-tables.

Flow of RLGA

We start by applying binary genetic coding toward a property of the dimensional data set in order to generate the state. Next, the Q-table is refreshed after the greedy algorithm is employed to choose an action, advance to the next state, and cross-operate with the global optimal person

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

with regard to a respective characteristics simultaneously. Similarly, a decision was made in the current state to carry out the mutation operation, resulting in a change to the Q-table. The population carried out the standard genetic operation and, at the end of each iteration, saved the person who was considered to be the most fit for the population as a whole. At the end of the procedure, the highest value of the either global or local variables is returned. While some members of the population have adapted to broaden the search field and enhance the probability of finding the best answer, the algorithm worked to optimise the discretization scheme offered. Whereas the algorithm was processing, this was done.

Results and discussion

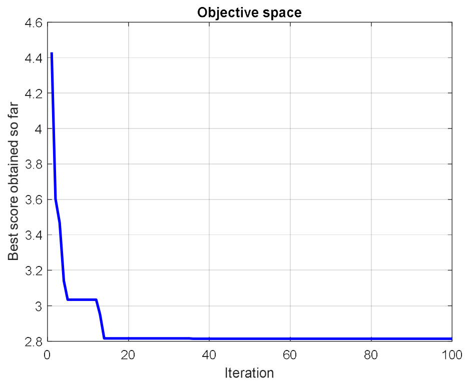

Figure 4 Convectional GA results

The above figure is the objective function of the Convectional GA, the value decreases by iteration process at iteration 20 the value got converged. The best value obtained from CGA is 2.81406.

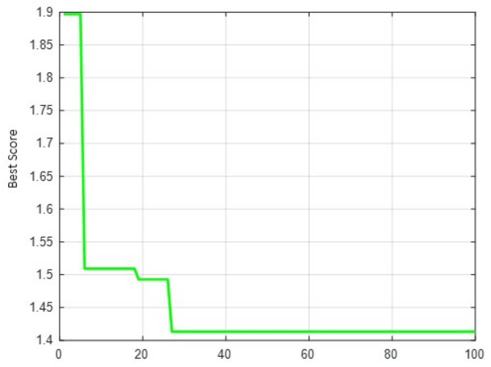

Figure 5 Q-learning based GA results

The above figure is the objective function of the Q-learning based GA, the value decreases by iteration process at iteration 27 the value got converged. The best value obtained from Q-learning based GA is 1.41336.

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

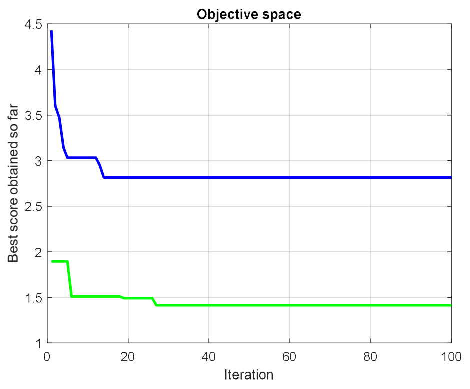

Figure 6 Comparison of CGA and RLGA

The above figure is the comparison plot of the CGA and RLGA, the proposed RLGA got the effective results than CGA. The obtained best score values are tabulated below.

Table 1 Objective function best values

S.No	**Algorithm**	**Objective function value (Best value)**
1	Convectional GA	2.81406
2	RLGA	1.41336

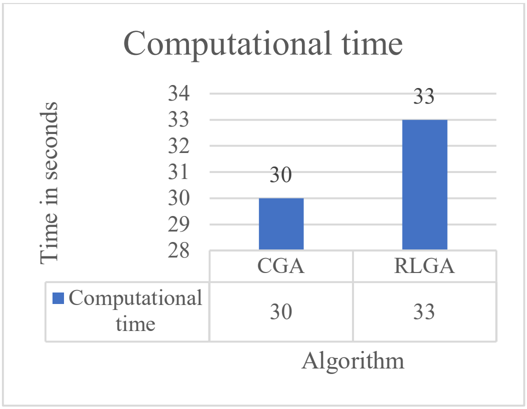

Figure 7 Computational time

The aforementioned time is the total processing time for the two suggested methods plus CGA. It takes more time to run the suggested method than CGA does, but it outperforms CGA in every way.

3. Conclusion

The study proposed an effective optimization technique of Q-learning based GA algorithm. The proposed method minimized the best score value. It was figured from the outcomes of the proposed algorithm that the performance of the proposed algorithm was very effective. The computational time of the proposed model is high but it resulted in best. The computing time for the CGA is 30 seconds, but the best score value is 2.81406, but the computational time for the proposed model is 33 seconds, but the best score is 1.41336, which is less than CGA. It is clear from the comparison graph that Q-learning GA performs better than CGA.

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/

References

[1] W. Chen, H. Zhang, M. K. Mehlawat, and L. Jia, "Mean–variance portfolio optimization using machine learning-based stock price prediction," *Appl. Soft Comput.*, vol. 100, p. 106943, 2021, doi: 10.1016/j.asoc.2020.106943.

[2] E. K. W. Leow, B. P. Nguyen, and M. C. H. Chua, "Robo-advisor using genetic algorithm and BERT sentiments from tweets for hybrid portfolio optimisation," *Expert Syst. Appl.*, vol. 179, no. April, p. 115060, 2021, doi: 10.1016/j.eswa.2021.115060.

[3] R. R. Bies, M. F. Muldoon, B. G. Pollock, S. Manuck, G. Smith, and M. E. Sale, "A genetic algorithm-based, hybrid machine learning approach to model selection," *J. Pharmacokinet. Pharmacodyn.*, vol. 33, no. 2, pp. 195–221, 2006, doi: 10.1007/s10928-006-9004-6.

[4] C. B. Kalayci, O. Polat, and M. A. Akbay, "An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization," *Swarm Evol. Comput.*, vol. 54, no. February, p. 100662, 2020, doi: 10.1016/j.swevo.2020.100662.

[5] S. Gupta, G. Bandyopadhyay, S. Biswas, and A. Upadhyay, *A Hybrid Machine Learning and Dynamic Nonlinear Framework for Determination of Optimum Portfolio Structure*, vol. 74. Springer Singapore, 2019.

[6] Y. Zhou, S. Zheng, and G. Zhang, "Machine-learning based study on the on-site renewable electrical performance of an optimal hybrid PCMs integrated renewable system with high-level parameters' uncertainties," *Renew. Energy*, vol. 151, no. xxxx, pp. 403–418, 2020, doi: 10.1016/j.renene.2019.11.037.

[7] J. Clifton and E. Laber, "Q-Learning : Theory and Applications," pp. 279–303, 2020.

[8] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A Theoretical Analysis of Deep Q-Learning," vol. 120, no. 1995, pp. 1–4, 2019, [Online]. Available: http://arxiv.org/abs/1901.00137.

[9] C. Salisbury *et al.*, "Effectiveness of an integrated telehealth service for patients with depression: A pragmatic randomised controlled trial of a complex intervention," *The Lancet Psychiatry*, vol. 3, no. 6, pp. 515–525, 2016, doi: 10.1016/S2215-0366(16)00083-3.

[10] Y. Xiao, J. Hoffman, T. Xia, and C. Amato, "Learning Multi-Robot Decentralized Macro-Action-Based Policies via a Centralized Q-Net," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 10695–10701, 2020, doi: 10.1109/ICRA40945.2020.9196684.

[11] M. E. Mangram, "A Simplified Perspective of The Markowitz Portfolio Theory," *Glob. J. Bus. Res.*, vol. 7, no. 1, pp. 59–70, 2013, [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=82211365&site=ehost-live.

[12] R. Ortiz, M. Contreras, and C. Mellado, "Improving the volatility of the optimal weights of the Markowitz model," *Econ. Res. Istraz.* , vol. 35, no. 1, pp. 2836–2858, 2022, doi: 10.1080/1331677X.2021.1981963.

[13] L. Wang, S. T. Branch, L. B. Johnson, and N. Aeronautics, "N91- 4o50."

[14] Q. Chen, M. Huang, Q. Xu, H. Wang, and J. Wang, "Reinforcement Learning-Based Genetic Algorithm in Optimizing Multidimensional Data Discretization Scheme," *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/1698323.

Copyright © 2022. Journal of Northeastern University. Licensed under the Creative Commons Attribution Non-commercial No Derivatives (by-nc-nd). Available at https://dbdxxb.cn/