FUZZY OPTIMIZATION AND MODELLING 3(1) (2022) 41-63
E-ISNN: 2676-7007

Contents lists available at FOMJ

Fuzzy Optimization and Modelling

Journal homepage: http://fomj.qaemiau.ac.ir/

Paper Type: Research Paper

Fuzzy Portfolio Optimization Using Credibility Theory: Multi-Objective Evolutionary Optimization Algorithms

Marieh Al-Sadat Miraboalhassani[a], Farzad Movahedi Sobhani[a,*], Emran Mohammadi[b]

[a] *Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*
[b] *School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran*

ARTICLE INFO

Article history:
Received 12 January 2022
Revised 6 March 2022
Accepted 7 March 2022
Available online 7 March 2022

Keywords:
Value at Risk
Conditional Value at Risk
Fuzzy Uncertainty
Meta-Heuristic Algorithms

ABSTRACT

Investors are always interested to choose the portfolio with the highest return and lowest risk for optimal asset management. A multi-objective portfolio optimization problem with cardinality constraint that determines the number of assets in a portfolio is considered in this paper. Objectives are maximizing the expected value of wealth and minimizing value at risk and conditional value at risk. Due to the complexity of the problem, it is necessary to use meta-heuristic algorithms. We use multi-objective evolutionary algorithms (Multi-Objective Particle Swarm Optimization, Non-Dominated Sorting Genetic Algorithm-II) to overcome this problem. In this research, the liquidity constraint and the thresholds of investments are considered. We use experts' opinions in a fuzzy method to deal with the uncertainties in the parameters and provide better and more quality decisions. Finally, an Iranian stock market case study is presented to examine the proposed model in various situations. The results indicate that examining uncertainties and other real-world assumptions provides more efficient and practical solutions.

1. Introduction

In post-modern portfolio theory, researchers have focused on undesirable risk measures such as value at risk (VaR) and conditional-value at risk (CVaR). The VaR of a portfolio is the loss does not exceed 1-α percent confidence. The CVaR is the average conditional loss above VaR. The CVaR can be used to balance risk and return. Portfolio optimization problems are affected by political, social, environmental, and economic uncertainties. The fuzzy approach is a powerful instrument to manage this kind of uncertainty [26-30]. VaR and CVaR should be redefined in fuzzy circumstances. In this research fuzzy portfolio optimization problem is considered by using credibility theory. Markowitz [21] introduced the modern portfolio theory. He defined

* Correspondig author
E-mail address: farzadmovahedi@gmail.com (Farzad Movahedi Sobhani)

DOI: 10.30495/fomj.2022.1949727.1054

portfolio risk as the variance of the expected return on the assets, based on the weighted average expected return on each asset. It is based on the assumption that investors have a quadratic utility function or that asset returns follow a multivariate elliptical distribution. Variance is introduced as a suitable risk measure in modern portfolio theory but in post-modern portfolio theory, undesirable risk measures such as VaR and CVaR are considered [1].

It is important to note that this optimization problem is non-convex and its solution is complex in minimizing the risk. Therefore, common commercial software cannot be used to solve these models. To solve the problem of value optimization at risk, heuristic and metaheuristic methods are suggested, Gilli et al. [9]. In addition, daily changes may occur in the parameters of the stock portfolio selection optimization problem. It is related to the inherent changes within the parameters, which occur over time due to the dynamic nature of the parameters. This uncertainty refers to regions where the frequency is high but the intensity is low in terms of frequency and intensity. Considering crisp parameters for input parameters leads to the optimal or low-quality solutions [4, 10].

Probability and fuzziness approaches are two important approaches to dealing with uncertainties. However, the first approach has weaknesses because it requires parameter distribution information. In many real-world problems, there is not enough information about distribution functions. Probability approaches also impose a lot of complexity on the problem. On the other hand, the fuzzy approach does not require information from the parameter distribution and does not add much complexity to the problem.

A multi-objective model will be presented in the current research that minimizes VaR and CVaR simultaneously. Furthermore, the inherent uncertainties of the parameters are considered by credibility-based fuzzy theory. In addition, because the proposed model has a non-convex space and cannot be solved using commercial software, two met heuristic methods will be used to solve the problem. The article continues as follows: In Section 2, the theoretical foundations and research background are presented. In Section 3, the research method introduces a fuzzy probability-based research method for conditional risk value and risk value. Section 4 introduces the limitations applied to the model. Section 5 describes the research methodology and solution approach. The main results obtained are listed in Section 6 and suggest potential developments.

2. Theoretical Foundations and Research Background

In this approach, concerning the tendency of the decision-makers (DM) in taking optimistic, pessimistic, and compromise attitudes, three measures including the possibility, necessity, and credibility measures are used to form the Fuzzy DEA (FDEA) models, respectively. However, decision-makers may have different preferences and so it is necessary to customize fuzzy DEA models according to the properties of DMU. Mishra et al. [22] used the MOPSO algorithm to solve the stock portfolio optimization problem and compared it with SPFGA, NSGAII, and SPEA-II methods. They used standard deviation minimization, return maximization, and return optimization to solve the optimization problem, calling risk measurement, and compared it with NSGAII, SPFGA, SPEA-II methods. Armananzas & Lorenzo [2] processed three meta-heuristic algorithms SA, GLS, and ACO, and showed the best optimizers, which are ACO and SA. Markowitz [21] sought to minimize deviations from the standard returns on assets, and deviations from expected returns on risk were called risk. Even though the return is higher than the desired average of capitalists, to eliminate it, he later replaced the measure of semi-variance [24] for the calculation of negative deviations. Mendelburt [23] and Fama [7] showed for the first time that the distribution of return on assets has a wider sequence and a higher peak than the normal distribution.

Therefore, there was a need for a measure of risk that did not consider the assumption of normality. So, in the early 1990s, JP Morgan introduced value at risk (VaR). This measure indicates how much portfolio assets are exposed to risk over a given time horizon at a given confidence level. Still, it does not provide information about the portfolio's expected losses. To cope with this issue, a conditional risk-based risk measure was later introduced by Rockefeller & Oriaso [33] that identifies expected losses at a set confidence level. One of the advantages of using this metric is becoming a linear programming model. Fuzzy theories are used to create a model that is averaged CVaR. In their study, evolutionary optimization algorithms is described in NSGAII, PESA, and SPEA to optimize their portfolio and replaced the mean-variance model with VAR and C-VAR criteria are closer to the Pareto optimal level. According to a genetic algorithm, a model developed based on CVaR using fuzzy simulation.

Gao et al. [11] developed their model based on CVaR in a fuzzy environment and used hybrid intelligent algorithms to solve it. The present paper uses fuzzy validation theory to estimate VaR and CVaR. To make the liquidity constraint model more efficient, according to the daily trading volume and the minimum and maximum constraints, the investment ratio and cardinality constraint have been added to the model to determine the number of assets in the portfolio to be more efficient. In Iranian studies, there is little use of multi-objective evolutionary optimization algorithms to solve problems. Derakhshan et al. [5] used the Markowitz model and the combined ant community optimization (ACO) algorithm. Gaivoronski, A., & Pflug [8] used a two-objective continuous ant algorithm to optimize the risk value as a measure. They showed that the Pareto front obtained by the NSGAII method is less extensive and more convergent than that of the NSGAII method [16].

3. Modelling and Research Methodology

3.1. Credibility-Based Fuzzy Programming

Based on the epistemic uncertainty we encountered in the problems, it is possible to deal with the imprecise parameters. As a result, the epistemic programming method should deal with the epistemic data. The two main branches of fuzzy programming are imprecise programming and flexible programming. The first approach is used to deal with the lack of knowledge about the exact values of the model parameters (epistemic uncertainty), and the second method is used to deal with the control of flexible target values and soft constraints [15, 20]. In the proposed model, the credibility-based fuzzy programming approach, Liu and Liu [17] is used as one of the valid methods to cope with the ambiguity in the parameters. Credibility fuzzy programming is based on a concept that enables the decision-maker to control the degree of confidence in satisfying constraints. It also supports different fuzzy data types, including triangular and trapezoidal numbers. It should be noted that, contrary to the criteria of possibility and necessity, which do not have a self-dual property, the credibility criterion is a self-dual criterion [18]. In other words, if the value of a fuzzy event is one, the decision-maker believes that this fuzzy event will happen. However, when the criterion of possibility is one, that fuzzy event may not occur, and when the criterion of necessity is zero, a fuzzy event may occur. Pishvaee et al. [32] presented the possibilistic programming method to convert the credit, fuzzy planning models, into definitive models. Based on fuzzy axis constraint programming and the mean expected value method, epistemic programming combines two criteria.

Suppose $\bar{\xi}$ it is a fuzzy variable with the degree of membership $\mu(x)$ and r is a real number. According to Liu and Liu [17], the credibility criterion is calculated as follows:

$$cr\{\bar{\xi} \le r\} = \tfrac{1}{2}\left(Pos\{\bar{\xi} \le r\} + Nec\{\bar{\xi} \le r\}\right) \tag{1}$$

It should be noted that since $Pos\{\bar{\xi} \le r\} = sup_{x \le r}\, \mu\,(x)$ and $Nec\{\bar{\xi} \le r\} = 1 - sup_{x > r}\, \mu\,(x)$, the credibility criterion can also be calculated as follows:

$$cr\{\bar{\xi} \le r\} = \frac{1}{2}(sup_{x \le r}\, \mu\,(x) + 1 - sup_{x > r}\, \mu\,(x)) \tag{2}$$

The average expected value based on the credibility criterion is calculated as follows:

$$E[\bar{\xi}] = \int_0^{\infty} Cr\{\bar{\xi} \ge r\}\, dr - \int_{-\infty}^{0} Cr\{\bar{\xi} \le r\}\, dr \tag{3}$$

Now suppose that $\bar{\xi}$ is a trapezoidal fuzzy number containing points $\bar{\xi} = \left(\xi_{(1)}, \xi_{(2)}, \xi_{(3)}, \xi_{(4)}\right)$ in Figure 1 below shown:

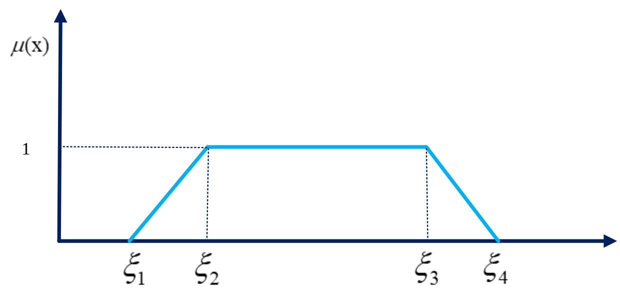

Figure 1- Possibility Distribution of a trapezoidal number

Based on the expected value equation, it can be calculated as follows:

$$\bar{\xi} = \frac{\xi_{(1)} + \xi_{(2)} + \xi_{(3)} + \xi_{(4)}}{4} \quad (4)$$

The corresponding credibility criterion can also be calculated as follows:

$$Cr\{\tilde{\xi} \le r\} = \begin{cases} 0, r \in -\infty, \xi_{(1)} \\ \dfrac{r - \xi_{(1)}}{2(\xi_{(2)} - \xi_{(1)})} r \in \xi_{(1)}, \xi_{(2)} \\ \dfrac{1}{2} r \in \xi_{(2)}, \xi_{(3)} \\ \dfrac{r - 2\xi_{(3)} + \xi_{(4)}}{2(\xi_{(4)} - \xi_{(3)})} r \in \xi_{(3)}, \xi_{(4)} \\ 1 r \in (\xi_{(4)}, +\infty) \end{cases} \quad (5)$$

$$Cr\{\tilde{\xi} \ge r\} = \begin{cases} 1 r \in -\infty, \xi_{(1)} \\ \dfrac{2\xi_{(2)} - \xi_{(1)} - r}{2(\xi_{(2)} - \xi_{(1)})} r \in \xi_{(1)}, \xi_{(2)} \\ \dfrac{1}{2} r \in \xi_{(2)}, \xi_{(3)} \\ \dfrac{\xi_{(4)} - r}{2(\xi_{(4)} - \xi_{(3)})} r \in \xi_{(3)}, \xi_{(4)} \\ 0 r \in (\xi_{(4)}, +\infty) \end{cases} \quad (6)$$

Thus, according to the definition, if $\alpha \succ 0.5$ can be shown [32, 49]:

$$Cr\{\tilde{\xi} \le r\} \ge \alpha \Leftrightarrow r \ge (2 - 2\alpha)\xi_{(3)} + (2\alpha - 1)\xi_{(4)} \quad (7)$$

$$Cr\{\tilde{\xi} \ge r\} \ge \alpha \Leftrightarrow r \le (2\alpha - 1)\xi_{(1)} + (2 - 2\alpha)\xi_{(2)} \quad (8)$$

The above equations can be used directly to convert potential probability inequalities into definite ones. The definition of expected value is also used to convert the objective function to its definite form.

3.2. Conditional Value at Risk (CVaR)

The VaR estimates the maximum possible loss at a given confidence level but does not determine how bad the loss is. Therefore, the VaR can be defined by the following relation:

$$p(L \succ VaR_\alpha) = 1 - \alpha \quad (9)$$

But the conditional value at risk (CVaR) estimates the expected loss at a set confidence level. The use of CVaR makes the stock portfolio selection model a linear programming model. Also, since CVaR≥VaR, the minimum VaR is obtained by solving this model. Accordingly, suppose f(x, ε) is a loss function of a stock portfolio. For a guaranteed level α, the CVaR is equal to the average of α -1% of losses, which can be calculated using the following function:

$$CVaR(x,\eta) = \eta + (1-\alpha)^{-1} \int_{\xi \in R^n} [f(X,\xi)]^+ p(\xi) d\xi \quad (10)$$

Where ξ is a random variable, η is the value at risk Var and $Z^+ = max\{Z, 0\}$

3.3. CVaR under Credibility Theory

Suppose ξ is a fuzzy variable and $\alpha \in 0,1$ is the risk confidence level. In this case, the value at risk under credit theory for ξ is expressed as a function $\xi_{VaR}: 0,1$ and is calculated as follows:

$$\xi_{VaR}(\alpha) = -sup\{x | Cr\{\xi \le x\} \le \alpha\} \quad (11)$$

Equally, we will also have $0 \le \alpha \le 1$ for a given level of risk confidence:

$$\xi_{VaR}(\alpha) = -\phi^{-1}(\alpha) \quad (12)$$

Where $-\phi^{-1}(\alpha)$ is the generalized inverse function of$\phi(x)$. The CVaR under the theory of credit for ξ is expressed as the function of $\xi_{CVaR}: 0,1 \to R$ and is calculated as:

$$\xi_{CVaR} = \frac{1}{1-\alpha} \int_\alpha^1 \xi_{VaR}(r)\, dr \quad (13)$$

Given the above relationships, the VaR and the CVaR can be obtained for a triangular fuzzy variable with parameters $\xi = (r_1, r_2, r_3)$at the confidence level$\alpha \in 0,1$:

$$\xi_{VaR}(\alpha) = \begin{cases} 2(r_1 - r_2)\alpha - r_1 \alpha \prec 0.5 \\ 2(r_2 - r_3)\alpha + r_3 - 2r_2 \alpha \ge 0.5 \end{cases} \quad (14)$$

$$\xi_{CVaR}(\alpha) = \begin{cases} \alpha r_1 - (1+\alpha) r_2 \alpha \le 0.5 \\ (\alpha - 1) r_2 - \alpha r_3 \alpha \succ 0.5 \end{cases} \quad (15)$$

A more than 50% confidence level is usually considered when estimating VaR and CVaR. Therefore, the case that $\alpha \prec 0.5$ is usually used.

3.4. Liquidity Constraint

Given the proximity to a real problem, daily trading volume is considered a trapezoidal fuzzy number, and liquidity constraints are obtained by fuzzy credit theory. Accordingly, the confidence level β determines the potential constraint on the credibility of a fuzzy event where the portfolio liquidity is higher than or equal to L, and is defined as follows:

$$Cr[L_1 x_1 + L_2 x_2 + \ldots + L_n x_n \ge L] \ge \beta \quad (16)$$

Note that any confidence level $\beta \leq 0.5$ may be meaningless in the real world because of its negligible value. Based on the topics we discussed earlier, if $\xi = (a, b, c, d)$ where $a \prec b \prec c \prec d$, we will have:

$$Cr\{\xi \geq r\} \geq \lambda \Leftrightarrow r \leq (2\lambda - 1)a + 2(1 - \lambda)b \tag{17}$$

If the trading volume of assets is in the form of a trapezoidal fuzzy number with $(L_{ai}, L_{bi}, L_{ci}, L_{di})$ parameters, then the liquidity limit will be the following:

$$\sum_{i=1}^{n} \left((2\beta - 1)L_{ai} + (2 - 2\beta)L_{bi}\right) x_i \geq Lq0 \tag{18}$$

where β is confidence level, and L is minimum portfolio liquidity. The investor determines this amount according to his/her expectations.

3.5. Cardinality Constraint

This limitation specifies the number of portfolio assets. For instance, it can be determined that there are exactly five assets in the investor's portfolio. This constraint is as follows:

$$\sum_{i=1}^{n} y_i = h \tag{19}$$

Where h is the number of assets in the portfolio and y_i is a zero-one variable.

3.6. Minimum and Maximum Investment Ratio Constraint

This constraint determines the maximum and minimum investment ratios for each asset and is defined as follows:

$$LO_i y_i \leq x_i \leq UO_i y_i \tag{20}$$

In the above relation, lo_i is the minimum investment ratio and uo_i is the maximum investment ratio for the ith asset.

4. Conceptual Model and Research Variables

According to the topics discussed, the mathematical model can be presented as follows. The objective function (21) minimizes the VaR. The objective function (22) minimizes the CVaR. The constraint (23) is the minimum return expected by the investor from the portfolio. Constraint (24) indicates the amount of liquidity based on the investor's wishes. Constraint (25) shows the cardinality limitation. Constraint (26) states the minimum and maximum investment ratio. Constraint (27) states that the total ratio of investment types equals one. Limits (28) and (29) show the problem decision variables.

$$Min\ VAR = \sum_{i=1}^{n} x_i[2(r_{i1} - r_{i2})\alpha - r_{i1}] \tag{21}$$

$$Min\ \ CVAR = \sum_{i=1}^{n} x_i[\alpha r_{i1} - (1+\alpha) r_{i2}] \tag{22}$$

s.t.

$$\sum_{i=1}^{n} x_i \left(\frac{r_{i1} + 2r_{i2} + r_{i3}}{4}\right) \geq R \tag{23}$$

$$\sum_{i=1}^{n} x_i[(2\beta - 1)L_{ai} + (2 - 2\beta)L_{ai}] \geq L \tag{24}$$

$$\sum_{i=1}^{n} y_i = h \tag{25}$$

$$LO_i y_i \leq x_i \leq UO_i y_i \tag{26}$$

$$\sum_{i=1}^{n} x_i = 1 \tag{27}$$

$$y_i = \{0,1\} \tag{28}$$

$$x_i = \{0,1\}, i = 1,2,\dots,n \tag{29}$$

The case study of the presented problem consists of 20 companies selected from all Tehran Stock Exchange-listed companies under the following conditions:

1. Select a company from each industry for which 30 more active stock exchange industries were considered.
2. Companies that have at least 50% of trading days each year
3. Elimination of companies from 50 more active stock exchange companies that have not entered in the three quarterly periods.

We explain and implement the multi-objective portfolio optimization problem based on a case study and analyze and evaluate the results. In other words, the designed algorithms are explained first, and then the results are presented based on them

5. Solution Approach

Some risk measures add to the problem's complexity. Computational complexity is, therefore, a major barrier to solving these problems. This section uses the MOPSO and NSGA2 algorithms described below to find quality answers promptly. Several authors have used these methods in the relevant literature, including Yang [46] and Lin and Liu [19].

5.1. Multi-Objective Non-Dominated Sorting Genetic Algorithm-II (NSGAII)

The NSGAII algorithm is based on the genetic algorithm used for situations where problems have multiple objective functions. A genetic algorithm was presented by Holland [14] and later used in optimization and machine learning problems [41]. This algorithm is the most famous evolutionary algorithm inspired by Darwin's theory Vidal & Goetschalckx [44]. Traditionally, this algorithm was used for binary representations, but today it is also used for other representations [41]. This algorithm starts with a set of candidate answers (population) and selects the best ones (parents) to generate new answers (children). New responses are made either by the combination (intersection operator 25) or by modification (mutation operator 26). Then, the children produced are replaced with weaker answers (placement).

This algorithm repeatedly generates better answers and uses them to generate new answers. The target population evolves in this way, similar to what happens in nature.

Figure 2 below shows: illustrates the evolution process of the proposed algorithm.

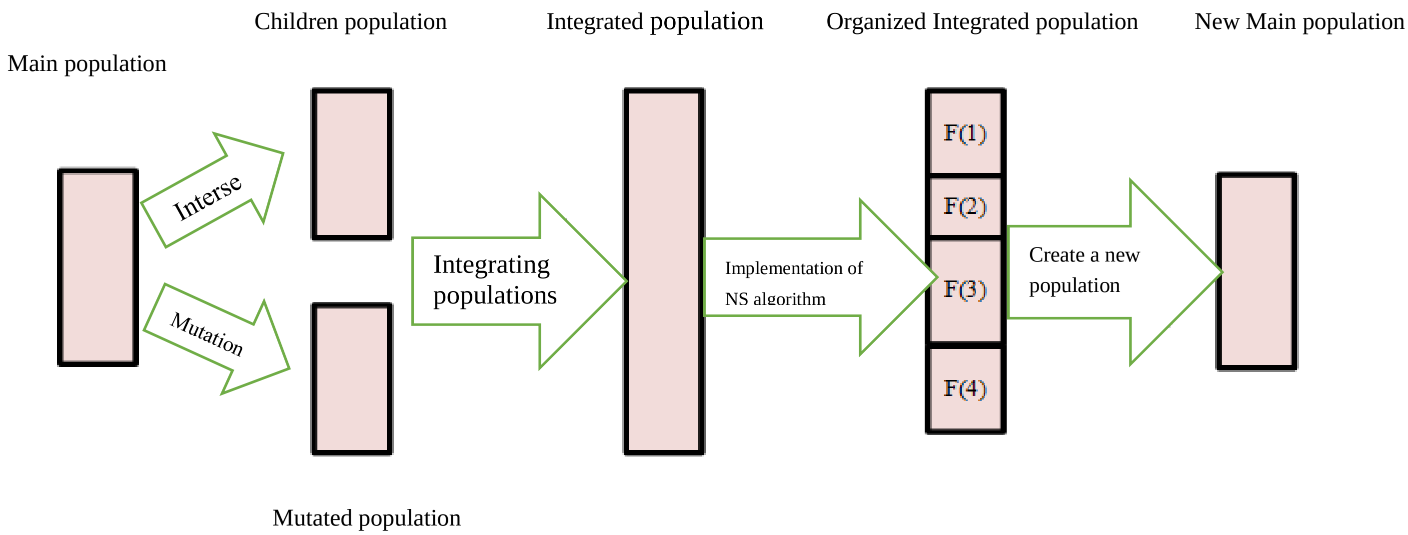

Figure 2. Graphic representation of the presented algorithm

According to the figure above, the offspring and mutated populations are first created using the intersection and mutation operators. In the next step, the three main populations, the offspring, and the mutant are merged. Then, the answers are evaluated and ranked. It should be noted when there are several objective functions, and as a result, the NSGAII algorithm is used. To evaluate and rank the answers, we use Non-dominated Sorting and Crowding distance, as described below. Finally, the number of other populations of the algorithmations I populations are this process will be repeated to stop the algorithm.

5.1.2. Structure of the NSGAII Algorithm

This section presents the structure of the NSGAII algorithm. In the proposed algorithm, three populations are considered: the main population, offspring, and mutants, whose number of members is N_{Or}, N_{Ch}, and N_{Mu}, respectively. The values of N_{Ch} and N_{Mu} are $2 \times \left[\frac{N_{Or} \times Pr_{Cr}}{2}\right]$ and $[N_{Or} \times Pr_{Mu}]$, respectively. It should be noted that [•] converts the value of a real number to its previous integer.

5.1.3. Evaluation of Answers

To evaluate and rank the answers in the NSGAII algorithm, two criteria are used:

1- Pareto Front Answer

2- Order on the Pareto front

Non-dominated Sorting algorithm is used to differentiate Pareto fronts. Different types of Pareto fronts can be detected using this algorithm. The second criterion, the order on the Pareto front, is done using Crowding distance. In the NSGAII algorithm, the answers are first ranked based on the Pareto front rank using Non-dominated Sorting. The Crowding distance algorithm ranks answers that have the same Pareto front. The following two methods are explained.

5.1.4. Non-dominated Sorting Algorithm

As previously explained, this algorithm is used to rank the Pareto front types. The symbols used are defined as follows:

S_p: The set of members of the population that are defeated by p.

N_p: The number of times p is defeated by others.

According to this algorithm, the answers that are not defeated by any answer are placed in the first Pareto Front. This answer is then deleted, and the answers that are not defeated by any other answer are placed in the second Pareto Front. This process is repeated for all answers, thus identifying all Pareto fronts.

5.1.5. Crowding Distance Algorithm

The Pareto fronts will be ordered using the Crowding distance method. Answers in which the Parthian front rank is the same are ranked by the Crowding distance method. This way, we try to have a representative all over the Pareto front. More precisely, it prevents the answers from concentrating on one part of the Pareto front. Figure 3 shows a Pareto mantle based on two objective functions.

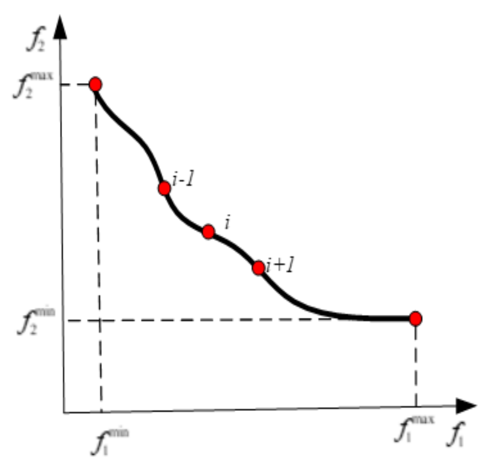

Figure 3. Pareto optimal solutions

In Figure 3, for an answer like i the Crowding distance is calculated as follows:

$$d_i^1 = \frac{\left|f_1^{i+1} - f_1^{i-1}\right|}{f_1^{max_1^{min}}} \tag{30}$$

$$d_i^2 = \frac{\left|f_2^{i+1} - f_2^{i-1}\right|}{f_2^{max_2^{min}}} \tag{31}$$

$$d_i = d_i^1 + d_i^2 \tag{33}$$

Based on the above relation, the number of objective functions is calculated at different distances, and then the final crowding distance is obtained from the sum of different dimensions. The greater the Crowding distance for the answer in example i, the lower the concentration of the answers in that part of the Pareto front. Hence, that answer is more desirable on the Pareto front. In general, if we have M dimensions, this criterion for each answer i is calculated as follows:

$$d_i^m = \frac{\left|f_1^{befor} - f_1^{next}\right|}{f_1^{max_1^{min}}} \tag{34}$$

$$d_i = d_i^1 + d_i^2 + \ldots + d_i^m = \sum_{j=1}^{m} d_i^j \tag{35}$$

5.1.6. Display and Evaluation of the Answers

The answer to the problem is represented by a vector containing n cells that n is the number of stocks considered in the problem. For each cell, a random number between zero and one is generated as Figure 4:

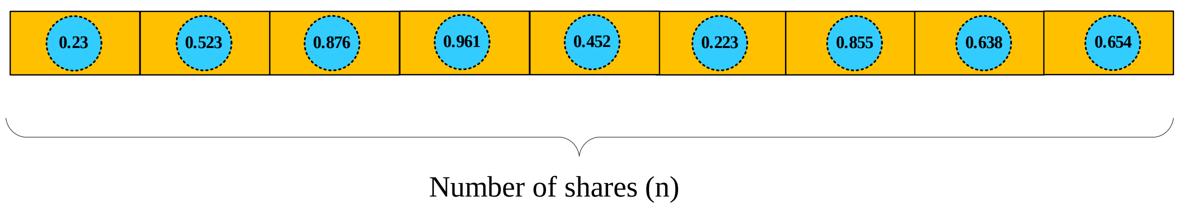

Figure 4. Schematic of the initial answer to the problem

Then, to make the sum of the weights equal to one, each weight is divided by the sum of the weights:

$$w_i^{new} = \frac{w_i^{old}}{\sum_{j=1}^{n} w_j^{old}} \forall i = 1,2,\ldots,n \tag{36}$$

So that w_i^{old} and w_i^{new} are the current weights and the new weights of the ith stock, respectively. See Figure 5.

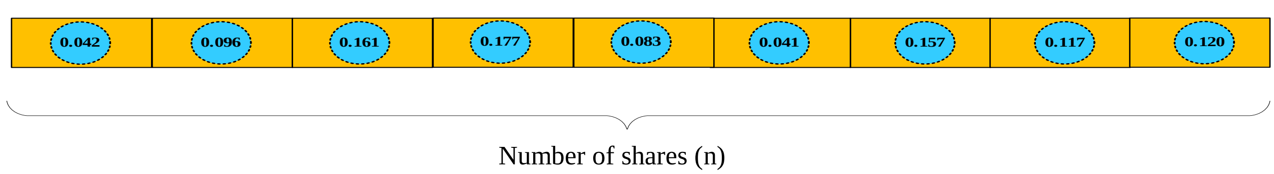

Figure 5. Schematic of the initial answer to the problem

In the above vector, the sum of the weights will equal one.

5.1.7. Intersection operator

The information from the two genotypes is combined with the intersection operator to make a new offspring. A special intersection operator is defined for each of the defined matrices in the wing.

Here, the sum of the weighted averages for the intersection operator is used. More precisely, a number between zero and one is generated, and two child vectors are generated based on it. The offspring cells are produced based on the following relation:

$$w_i^{o1} = \alpha w_i^{p1} + (1-\alpha) w_i^{p2} \quad (37)$$

$$w_i^{o2} = (1-\alpha) w_i^{p1} + \alpha w_i^{p2} \quad (38)$$

Where w_i^{o1} and w_i^{o2} are the i[th] yield vector weights for the first and second children. w_i^{p1} And w_i^{p2} are the i[th] yield vector weights for the first and second parents. α is the number produced between zero and one.

Figure 6 shows an example of an intersection assuming that the alpha is 0.35. Offspring vectors are obtained from the weighted sums of the parent vectors.

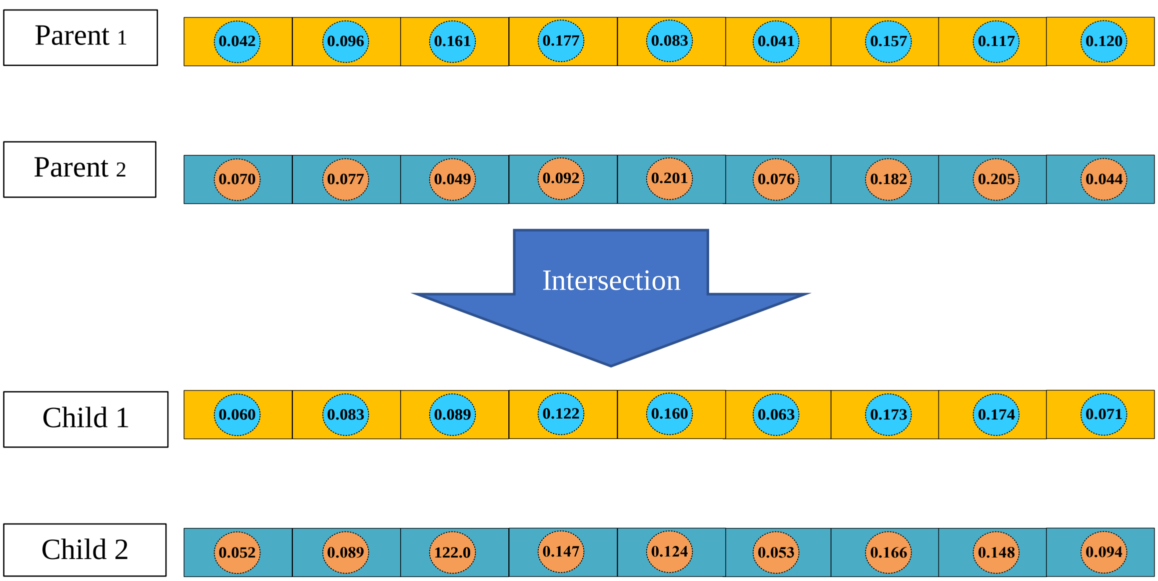

Figure 6. The intersection in the first vector

5.1.8. Mutation Operator

The purpose of the mutation operator is to generate variability in the algorithm that escapes the optimal local response. The mutation operator is used to modify one or more genes on one or more chromosomes for this purpose. In the present research, two cells are selected from a vector, changing their positions. Figure 7 shows an example of a mutation. As can be seen, the third and sixth positions are selected, and the positions are changed.

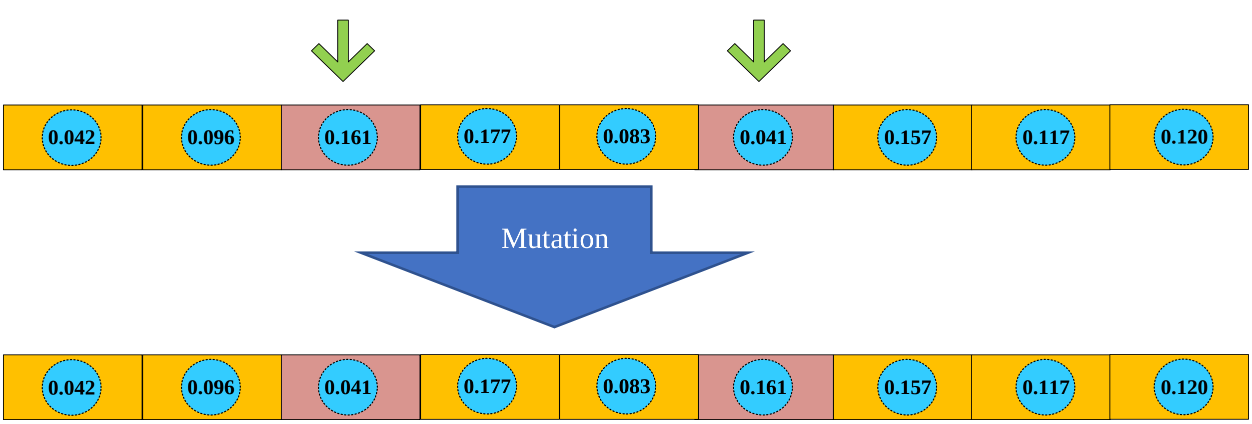

Figure 7. Mutations

5.1.9. New Population Selection

If a fixed population, one decision is which members will be allowed to go to the next stage. Normally, this is based on the value of the objective function. The present study involves first integrating the main population, offspring, and mutants and then selecting "Nor" numbers from the best population for the new population

5.2. Multi-Objective Particle Swarm Optimization (MOPSO)

The particle swarm optimization algorithm is a probabilistic optimization algorithm that employs a population-based approach. This algorithm was first proposed by Eberhard and Kennedy [6]. This algorithm simulates the social behavior of animals such as insects, birds, and fish. The algorithm is very similar to other evolutionary algorithms, such as genetic algorithms. These algorithms usually generate random answers and search for the optimal answer. However, unlike the genetic algorithm, the particle swarm optimization algorithm does not use intersection and mutation operators. According to a particle swarm optimization algorithm, potential solutions called particles are searched according to the position of the best particles at each level of the problem. The main advantage of this algorithm is its ease of implementation instead of genetic algorithms. Furthermore, this algorithm has a small number of input parameters that can be easily adjusted. This algorithm has been used well in various fields such as function optimization, training of artificial neural
Networks, fuzzy system control, etc.

In general, artificial life can be divided into two areas:

1- Artificial life focuses on how computational techniques can study biological phenomena.

2- Artificial life focuses on how biological techniques can solve computational problems.

Here the focus is on the second part. There are many computational techniques based on biological systems. For example, an artificial neural network is a simplified human brain model. We also discuss another biological system (i.e., social system) with the genetic algorithm based on human evolution developed here. This algorithm specifically emphasizes the swarm behavior of individuals, which is called swarm intelligence. All simulations use local processes, such as cellular automata models, and may overshadow social behavior. Some well-known examples in this field are Floy and Bid. Both simulations are designed to interpret the motion of organisms in a group of birds or fish. These simulations are also commonly used in computer animation or computer design.

Two popular inspirational methods have been developed based on the swarm behavior of individuals and in the fields of intelligent computing, including ant colony optimization and particle swarm optimization. It is based on the behavior of ants and has a very good performance in optimizing discrete optimization problems.

The particle swarm optimization algorithm is derived from the simulation of social systems. Initially, the main purpose of this algorithm was to simulate the movements of birds or fish graphically. However, it was later found that the particle swarm optimization algorithm could also be used to optimize various problems.

As mentioned earlier, the particle swarm optimization algorithm simulates bird behavior. Let's assume a scenario as follows: A group of birds randomly search an area. There is only one food unit in the area. None of the birds know where the food is. They only know how far away they are from food in each iteration. Therefore, the best strategy is to follow the birds that are closest to the food.

The particle swarm optimization algorithm is based on the memory of past movements and uses it to solve optimization problems. In this algorithm, each bird is considered as an answer. Here, each answer is referred to as a "particle". All particles have a fit of life that is evaluated and must be optimized. Moreover, each particle has a specific velocity that determines its path. It should be noted that the particles in each movement also pay attention to the position of the best particle.

A particle swarm optimization algorithm starts from a group of answers and searches for the optimal answer using a special update formula. As shown in Figure 8, in each iteration of the algorithm, the update formula based on the three vectors updates the position of each particle. The first component of the velocity vector in the previous period of each particle is denoted by the symbol Vit for its particle in the t^{th} stage. This means that each particle believes in the path it has taken and moves in its path. The second part is based on the movement to the best position ever achieved for its particle in the t^{th} phase, which is represented by the Pit symbol. This movement means that the particle tends to move in the direction of its best memory. Up to the t th stage, the third component moves in the direction of the best particle obtained across the whole algorithm, which is marked with the symbol Gt Effect of current velocity.

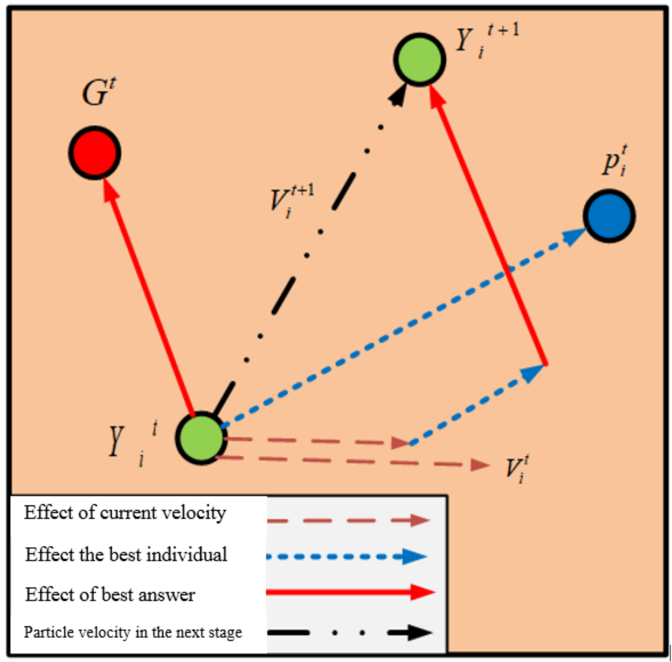

Figure 8. Movement of each particle in the particle swarm optimization algorithm

According to the figure, each particle will update its speed according to the following formula:

$$V_i^{t+1} \leftarrow \omega\, V_i^t + \varphi p\ rp\ (p_i^t - Y_{i,d}^t) + \varphi grg\ (G^t - Y_i^t) \quad (39)$$

Where Y_i^t is the position of the i particle in the t^{th} stage. Furthermore, ω, φ_p and φ_g are the best personal memory and the most accurate measures of the importance of moving to the previous speed. These parameters are calculated as the input of the algorithm. It should be noted that r_p and r_g are random numbers between zero and one. The position of each particle is now updated according to the velocity formula described above:

$$Y_i^{t+1}\ \ Y_i^t + V_i^{t+1} \quad (40)$$

Most steps of the multi-objective particle swarm optimization algorithm are clear. Nevertheless, in the next section of the tabulation, each repository's best personal memory and leader will be described.

Figure 9 illustrates how the response space is first divided into different sections to create order in the Pareto Front and distribute the answer evenly throughout the Pareto Front. In other words, the sections with less focus in terms of answers are given more weight than those with more answers.

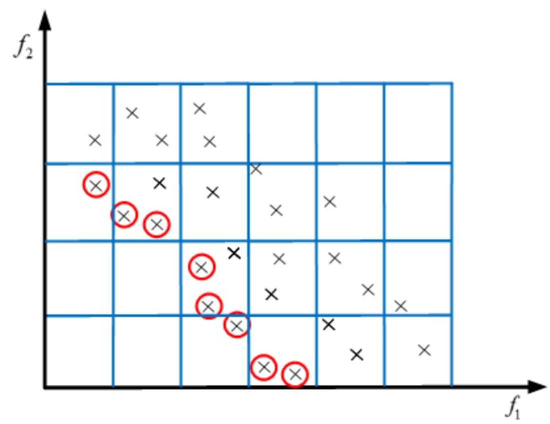

Figure 9. Tabulation in the MOPSO algorithm

The importance of each section means choosing them to lead each repository. According to Boltzmann's relation, any part where there is less focus on the answers is more likely to be chosen to lead each repository. The Boltzmann relation is as follows.

$$p_i \alpha \; exp(-\beta n_i) \qquad (41)$$

$$p_i = \frac{e^{-\beta n_i}}{\sum_j e^{-\beta n_i}} \beta \geq 0 \qquad (42)$$

Based on the above relation, the sections with fewer answers (n) are more likely to be selected

According to the particle swarm optimization algorithm, if one of the new answers defeats the best individual memory, the new answer is considered the best individual memory. If two answers do not defeat each other, one is selected randomly.

The answer is represented as a vector containing n cells, where n equals the number of stocks considered in the problem. Each cell generates a random number between zero and one at the beginning (Figure 10):

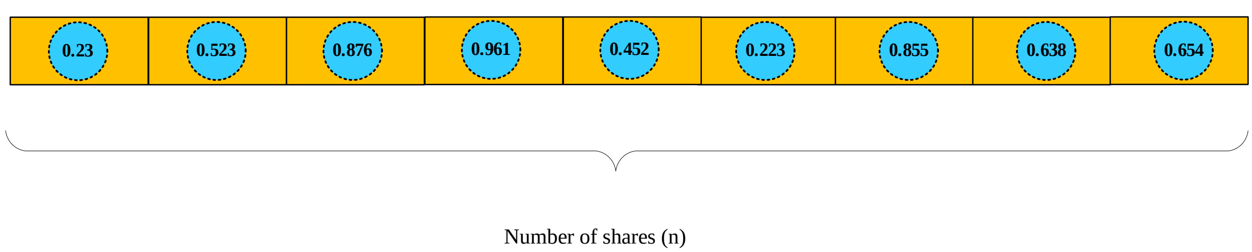

Figure 10. The initial answer to the problem

Each weight is divided by the sum of the weights to make the sum equal to one:

$$w_i^{new} = \frac{w_i^{old}}{\sum_{j=1}^{n} w_j^{old}} \forall i = 1,2,\ldots,n \qquad (43)$$

So that w_i^{old} and w_i^{new} are the current weights and the new weights of the stock i, respectively. Then, the new vector will be as Figure 11.

6.2.1. Computational Results

This section presents various experiments to evaluate the model and algorithm. MATLAB R2014a software is used to encode the genetic algorithm, and a personal laptop with a 2.93 GHz CPU and 8 GB of RAM is used to solve the examples.

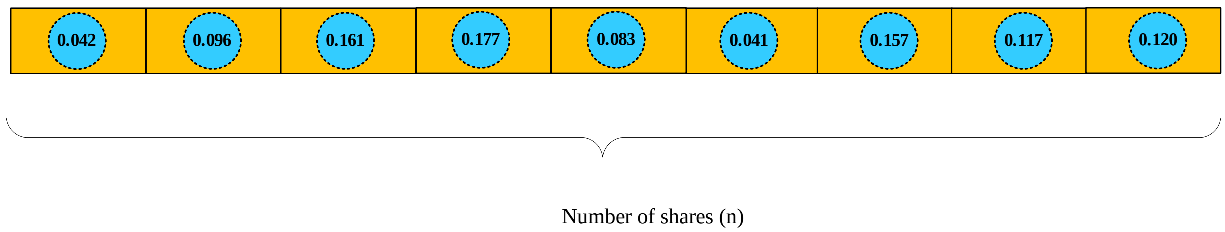

Figure 11. The final answer to the problem

An example is solved five times, and its average is reported. It should be noted that time is reported here in seconds.

6.2.2. Parameter Setting

The performance of meta-heuristic algorithms is significantly dependent on the values of their parameters. For this purpose, we used the Taguchi experiment design method to adjust the algorithm's parameters [3]. This method is based on two types of factors, including controllable and disorder (uncontrollable) factors. This method establishes the process by identifying the control factor and minimizing the effects of process disorders [13]. With a small number of experiments, the Gucci method studies the full range of parameters using a special orthogonal arrays design. In addition, the amount of deviation in the response is measured by the signal-to-noise (S/N) ratio, and the goal is to maximize it.

$$S/N \text{ ratio} = -10\left(\frac{1}{n}\sum_{j=1}^{n}\log_{10} {RE_j}^2\right) \tag{44}$$

Where n is the number of iterations and REj is the number of repetitions in iteration j. In the present study, we have two objective functions. This objective function is considered in the Taguchi model when we have one function. But when we have two functions (OF1 and OF2), we have to convert these functions into objective functions. To determine the final criterion for applying the Taguchi method, we should consider the following criteria.

6.2.3. Taguchi Methods

Taguchi Methods Statistical methods, also called robust design methods, developed by Genichi Taguchi to improve the quality of manufactured goods and now used in engineering in biotechnology, marketing, and advertising, have been widely welcomed.
Taguchi's work includes three principal contributions to statistics:

- A specific loss function
- Off-line quality control
- And Innovations in the design of experiments

Design of experiments

Taguchi independently developed her theories and in 1954, following Fisher, proposed

Taguchi's designs aimed to allow a greater understanding of variation than did many of the traditional designs from the analysis of variance (following Fisher). Taguchi contended that conventional sampling is inadequate here as there is no way of obtaining a random sample of future conditions. [9, 31] In Fisher's design of experiments and analysis of variance, experiments aim to reduce the influence of nuisance factors to allow comparisons of the mean treatment effects. Variation becomes even more central in Taguchi's thinking.

Taguchi proposed extending each experiment with an "outer array" (possibly an orthogonal array); the "outer array" should simulate the random environment in which the product would function. This is an example of judgmental sampling. Many quality specialists have been using "outer arrays".

Later innovations in outer arrays resulted in "compounded noise." This involves combining a few noise factors to create two levels in the outer array: First, noise factors that drive output lower, and second, noise factors that drive output higher. "Compounded noise" simulates the extremes of noise variation but uses fewer experimental runs than would previous Taguchi designs.

6.2.4. Mean Ideal Distance (MID)

$$MID = \frac{\sum_{i=1}^{n} \sqrt{\left(\frac{f_{1i}-f_1^{best}}{f_{1,total}^{Max}-f_{1,total}^{Min}}\right)^2+\left(\frac{f_{2i}-f_2^{best}}{f_{2,total}^{Max}-f_{2,total}^{Min}}\right)^2}}{n} \quad (45)$$

Where n represents the number of non-defeated solutions. f_{total}^{Min} And f_{total}^{Max} represent the minimum and maximum OFi in all iterations. It is clear that the lower the MID value, the better the algorithm performance.

6.2.5. Diversification Metric (DM)

This index shows the diversification of Pareto solutions and is defined as the following equation:

$$DM = \sum_{i=1}^{n} \sqrt{\left(\frac{Max(f_{1i})-Min(f_{1i})}{f_{1,total}^{Max}-f_{1,total}^{Min}}\right)^2 + \left(\frac{Max(f_{2i})-Min(f_{2i})}{f_{2,total}^{Max}-f_{2,total}^{Min}}\right)^2} \quad (46)$$

One of the advantages of the proposed algorithm is the variability of the solutions produced. Therefore, the larger DM is preferred

6.2.6. Spacing Metric (SM)

This index shows the uniformity of the set of non-dominant solutions and is defined as below:

$$SM = \frac{\sum_{i=1}^{n-1}|d_i-\bar{d}|}{(n-1)\bar{d}} \quad (47)$$

Where, n represents the number of non-defeated solutions, d_i represents the Euclidean distance of uccessive non-defeated solutions, and $\bar{d}$ is the mean ofd_i. The lower the value of this index, the higher its desirability.

This section seeks to adjust the parameters of two algorithms, NSGA-II and MOPSO, to obtain the best performance. In NSGA-II, the maximum number of iteration (Max_{it}), the combined probability of crossover ($P_{crossover}$), the probability of mutation ($P_{mutation}$), and the rate of mutation (M_u) are considered as basic parameters, and in MOPO, the maximum number of iteration (Max_{it}) is considered. Population number (nPop), leader selection pressure (Beta), and leader selection pressure (Gamma) are considered as basic parameters. According to the literature and expert opinions for the introduced parameters, three suggested levels are considered, which are presented in Table 1.

Table 1. Algorithm parameters level

NSGA-II				MOPSO			
Level	1	2	3	Level	1	2	3
Maxit	100	150	200	Maxit	100	200	300
Pcrossover	0.4	0.5	0.6	nPop	100	150	200
Pmutation	0.6	0.65	0.7	Beta	1	1.5	2
Mu	0.2	0.25	0.3	Gamma	1	1.5	2

According to the level set for the basic parameters, Designs L9 was performed by minitab19 software and the results are reported in Figures 12 and 13:

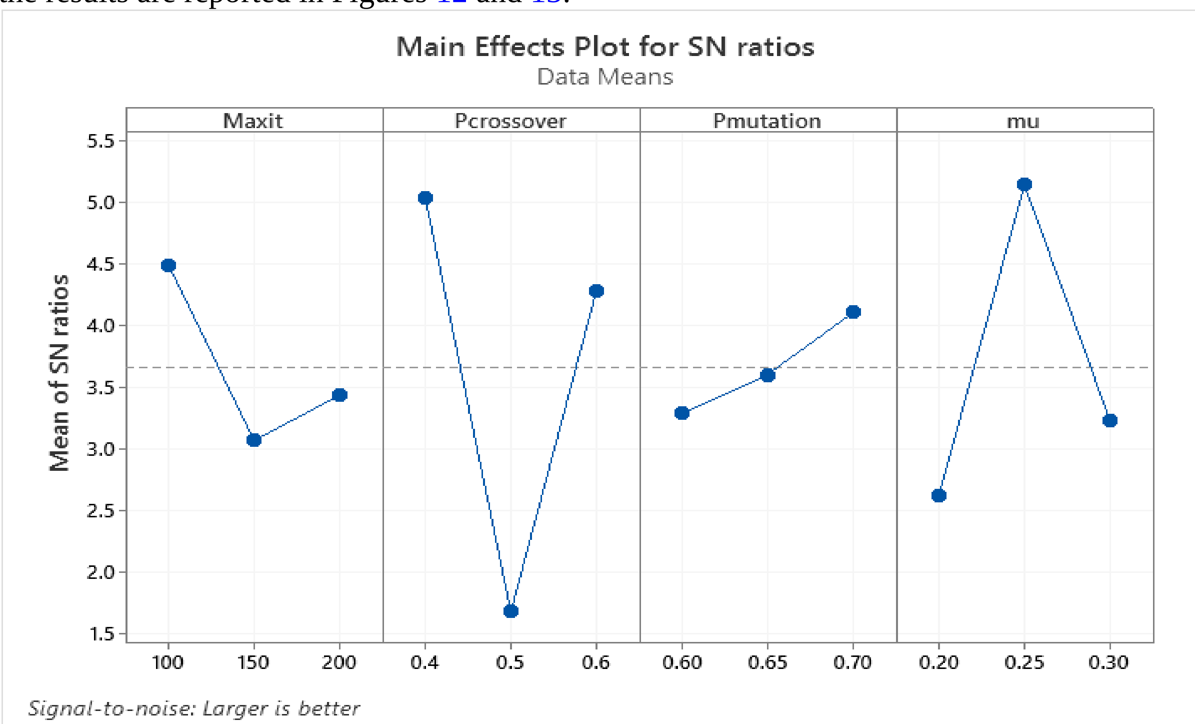

Figure 12- The initial answer to the problem

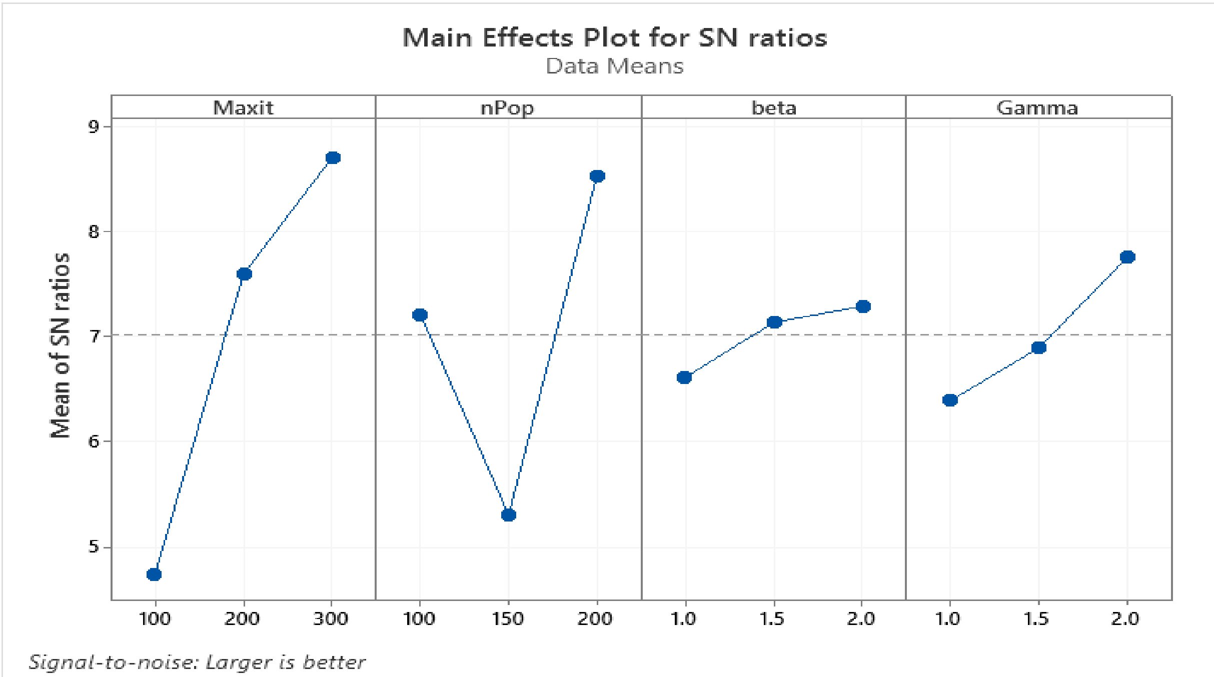

Figure 13- The initial answer to the problem

According to Figures 12 and 13, since the response variable is considered to be the maximum, the "S / N: Bigger is Better" mode in TM needs to be considered. Finally, the optimal levels for each basic parameter are defined as Table 2.

Table 2. The optimal value of the parameters

NSGA-II		MOPSO	
Optimum Level		Optimum Level	
Maxit	100	Maxit	300
Pcrossover	0.4	nPop	200
Pmutation	0.7	Beta	2
Mu	0.25	Gamma	2

As discussed in the previous section, the optimal level of NSGA-II and MOPSO basic parameters was determined. In this section, the aim is to compare the performance of the two proposed algorithms. As in the previous section, SM, DM, and MID markers were used to evaluate the performance of the two proposed algorithms. In addition to the introduced indicators, we can also use the variable quality metric (QM). This index means that all the best Pareto solutions from each algorithm must be put together and compared. Dominant solutions should be eliminated, and the ratio of the number of responses remaining to the number of initial responses indicates quality. An algorithm performs better with more QM. Lastly, two algorithms are implemented with optimal levels of basic parameters, and the results are reported in Table 3.

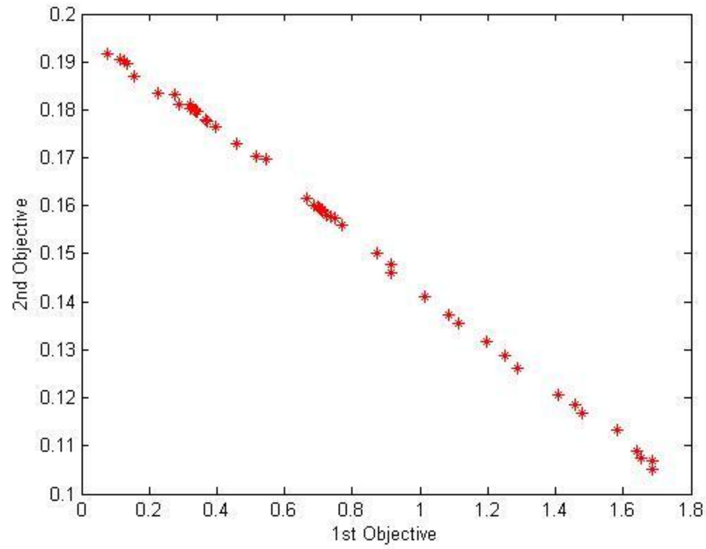

Figure 14. Pareto optimal solutions for MOSO algorithm

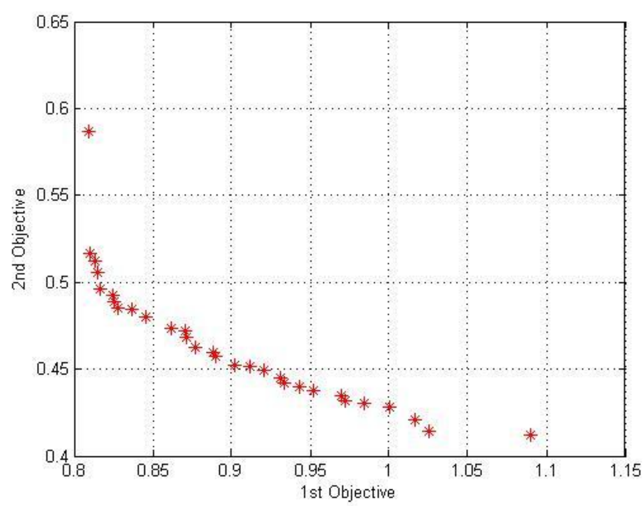

Figure 15. Pareto optimal solutions in NSGA2 algorithm

Table 3. NSGA2 versus MOPSO Algorithm

Algorithm					Index
	QM	MID	DM	SM	Final Score
NSGA-II	0.725	0.0627	1.4142	0.64	0.5799
MOPSO	1	0.0599	0.5343	0.9713	0.36

MOPSO outperforms NSGA-II in terms of QM because it has not mastered any of Pareto's optimal solutions, as shown in Table 9. Also, MOPSO has performed better in MID than NSGA-II. That is, MOPSO shows Pareto optimal solutions to be close to more ideal ones. But in DM, NSGA-II performs better than MOPSO, and it can be seen that the diversification of Pareto optimal solutions in NSGA-II is performed. Also, the SM index shows the superiority of NSGA-II over MOPSO. In light of the issues raised, it can be concluded that MOPSO has better performance than NSGA-II in terms of quality, while on the other hand, NSGA-II has better performance diversity than MOPSO.

6. Results Analysis

Different points can be chosen in the optimal Pareto diagram based on the decision maker's preferences. The chart below shows the percentage of assets invested in 20 different companies. Table 10 also shows the share of each company. It can be seen that the answers obtained for the two methods are almost identical. The results show that Ghadir Leasing with 0.4 shares and Fars Cement with 0% shares has the lowest investment share.

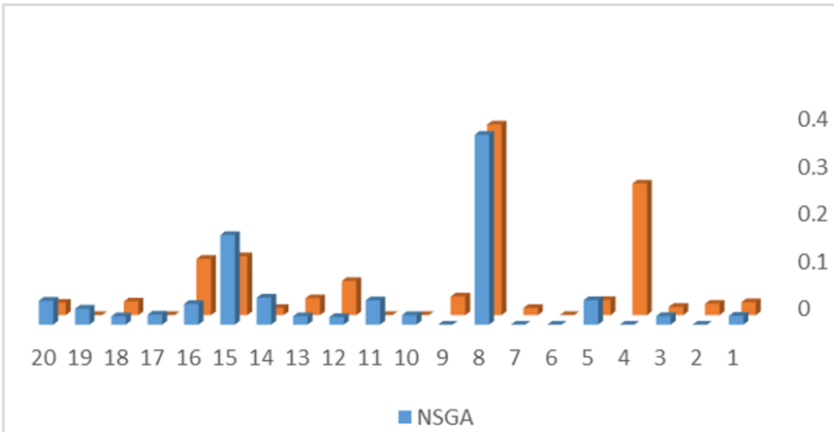

Figure 16. NSGA2 and MOPSO algorithms

Table 4. Percentage of investment on assets with NSGA2 and MOPSO algorithms

No.	Company Name	Percentage of Investment in Assets by MOPSO Method	Percentage of Investments in Assets by NSGA2 Method
1	Bafgh Mines	0.0272	0.0184
2	Building investment	0.0240	0
3	Eghtesad Novin Bank	0.0173	0.0178
4	Shipping	0.0276	0
5	Mapna	0.0318	0.0512
6	Saipa	0	0.0198
7	Informatics services	0.0149	0
8	Ghadir Leasing	0.4000	0.3984
9	Melat Investment	0.0390	0
10	Fars Cement	0	0
11	Oil industry investment	0	0.0509
12	Isfahan Mobarakeh Steel	0.0716	0.0158
13	Iran porcelain	0.0355	0.0176
14	Iran Telecommunication	0.0149	0.0566
15	Gas pipe	0.1236	0.1881
16	Tractor manufacturing	0.1180	0.0434
17	Iran Transfo	0	0.0210
18	Abadan Petrochemical	0.0286	0.0176
19	Behshahr Industries Development	0	0.0334
20	Razak	0.0260	0.0501
	Total weight percentage	1	1

In the present paper, a model is presented that minimizes the value at risk (VaR) and the conditional value at risk (CVaR) simultaneously and as a multi-objective model. The inherent uncertainties of the parameters were also taken into account, and the feasibility planning approach was used to deal with them. In addition, because the proposed model has a non-convex space and cannot be solved using commercial software, two meta-heuristic methods will be used to solve it. Accordingly, the research methodology presented in the present study consisted of the following four steps. To optimize the stock portfolio selection problem, a multi-objective optimization model was first proposed. The method of dealing with uncertainty was explained and based on that, the uncertain model became definite. Due to the complexity of these problems, two meta-heuristic methods including MOPSO and NSGAII were used to solve the proposed model. Finally, based on the proposed algorithms, the optimal Pareto answer was obtained. The results show that MOPSO outperforms NSGA-II in terms of quality, whereas NSGA-II outperforms MOPSO in terms of performance diversity. For the future research, robust optimization, interval programming, and uncertainty theory can be used in order to deal with data uncertainty [26-30]. Additionally, data envelopment analysis approach can be employed for stock evaluation [12, 33-49].

Conflict of interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Alexander, G.J., Baptista, A.M., & Yan, S., (2007). Mean–variance portfolio selection with 'at-risk'constraints and discrete distributions. *Journal of Banking & Finance*, 31(12), 3761-3781.
2. Armananzas, R. & Lozano, J. A. (2005). A multiobjective approach to the portfolio optimization problem. *IEEE Congress on Evolutionary Computation*, 2: 1388- 1395.
3. Anagnostopoulos, K. & Mamanis, G. (2009). Multiobjective evolutionary algorithms for complex portfolio optimization problems. *Computational Management Science*, 8, 259-279.
4. Ben-Tal, A. & Nemirovski, A., (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3), 411-424.
5. Derakhshan, M., Golmakani, H., & Hanafizadeh, P., (2012). Multiobjective portfolio selection of Tehran stock exchange with the metaheuristic optimization approach. *International Journal of Industrial Engineering and Production Management*, 23(3), 318-331.
6. Eberhart, R. & Kennedy. J., (1995). Particle swarm optimization. *in Proceedings of the IEEE international conference on neural networks. Citeseer*.
7. Fama, E. F., (1965). The behavior of stockmarket prices. *The Journal of Business,* 38(1), 34-105.
8. Gaivoronski, A., & Pflug, G., (2005), Value-at-risk in portfolio optimization: properties and computational approach. *Journal of Risk,* **7**(2), 1-31.
9. Gilli, M., Këllezi, E., & Hysi, H., (2006). A data-driven optimization heuristic for downside risk minimization. *Swiss Finance Institute Research Paper*, 2-6.
10. Guzman, Y.A., Matthews, L.R., & Floudas, C.A., (2017). New a priori and a posteriori probabilistic bounds for robust counterpart optimization: II. A priori bounds for known symmetric and asymmetric probability distributions. *Computers & Chemical Engineering*, 101, 279-311.
11. Gao, J., Zhang, X., & Wang, Q. (2011). Fuzzy portfolio selection based on MeanCVaR models. *In 2011 International Conference on Business Computing and Global Informatization*. 98-100.
12. Gregoriou, G. N., & Zhu, J. (2007). Data envelopment analysis. *The Journal of Portfolio Management,* 33(2), 120-132.
13. Hamzadayi, A., & Yildiz, G., (2013). A simulated annealing algorithm based approach for balancing and sequencing of mixed-model U-lines. *Computers & Industrial Engineering,* 66(4), 1070-1084.
14. Holland, J.H., (1975). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. *Ann Arbor, Michigan: The University of Michigan Press.*

15. Inuiguchi, M., & Ramık, J., (2000). Possibilistic linear programming: a brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem. *Fuzzy Sets and Systems,* 111(1), 28-3.
16. Khaleiji, M., Zeiaee, M., Tabei, A., Jahed-Motlagh, M.R. & Khaloozadeh, H. (2009). Dynamically weighted continuous ant colony optimization for bi objective portfolio selection using value-at-risk. *Third Asian International Conference on Digital Object Identifier, 1*(2), 230-235.
17. Liu, B., &. Liu, Y. K., (2002). Expected value of fuzzy variable and fuzzy expected value models. *IEEE Transactions on Fuzzy Systems,* 10(4), 445-450.
18. Li, X. & Liu, B., (2006). A sufficient and necessary condition for credibility measures. International Journal of Uncertainty. *Fuzziness and Knowledge-Based Systems,* 14(05), 527-535.
19. Lin, C., & Liu, Y.T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research*. 185(1): p. 393-404.
20. Mula, J., Poler, R., & Garcia, J., (2006). MRP with flexible constraints: A fuzzy mathematical programming approach. *Fuzzy sets and systems,* 157(1): p. 74-97.
21. Markowitz, H., (1952). Portfolio selection, *The Journal of Finance, 7*(1), 77–91.
22. Mishra, S.K., Panda, G. & Meher, S. (2009). Multi-objective particle swarm optimization approach to portfolio optimization. *World Congress on Nature & Biologically Inspired Computing*. DOI:10.1109/NABIC. 5393659.
23. Mandelbrot, B. B., (1997). The variation of certain speculative prices. *InFractals and Scaling in Finance. Springer New York*. 371-418.
24. Markowitz, H. M., (1991). Foundations of portfolio theory. *The Journal of Finance,* 46(2), 469-477.
25. Peykani, P., & Seyed Esmaeili, F.S. (2021). Malmquist productivity index under fuzzy environment. *Fuzzy Optimization and Modeling Journal, 2*(4), 10-19.
26. Peykani, P., Hosseinzadeh Lotfi, F., Sadjadi, S.J., Ebrahimnejad, A., & Mohammadi, E. (2022). Fuzzy chance-constrained data envelopment analysis: a structured literature review, current trends, and future directions. *Fuzzy Optimization and Decision Making*, 21, 197–261.
27. Peykani, P., Memar-Masjed, E., Arabjazi, N., & Mirmozaffari, M. (2022). Dynamic performance assessment of hospitals by applying credibility-based fuzzy window data envelopment analysis. *Healthcare, 10*(5), 876.
28. Peykani, P., Namakshenas, M., Nouri, M., Kavand, N., & Rostamy-Malkhalifeh, M. (2022). A possibilistic programming approach to portfolio optimization problem under fuzzy data. *Advances in Econometrics, Operational Research, Data Science and Actuarial Studies*, 377-387. Springer, Cham.
29. Peykani, P., Namazi, M., & Mohammadi, E. (2022). Bridging the knowledge gap between technology and business: an innovation strategy perspective. *Plos One,* 17(4), e0266843.
30. Phadke, M.S., (1995)., Quality engineering using robust design. *Englewood Cliffs, New Jersey: Prentice Hall*.
31. Pishvaee, M. S., Torabi, S. A., & Razmi, J. (2012). Credibility-based fuzzy mathematical programming model for green logistics design under uncertainty. *Computers & Industrial Engineering, 62*(2), 624-632.
32. Rockafellar, T.R. & Uryasev, S. (2000). Optimization of Conditional value-at-risk. *Journal of Risk* 21- 47.
33. Rotela Junior, P., Rocha, L. C. S., Aquila, G., Balestrassi, P. P., Peruchi, R. S., & Lacerda, L. S. (2017). Entropic data envelopment analysis: a diversification approach for portfolio optimization. Entropy, 19(9), 352.
34. Santos Arteaga, F. J., Ebrahimnejad, A., & Zabihi, A. (2021). A new approach for solving intuitionistic fuzzy data envelopment analysis problems. *Fuzzy Optimization and Modeling Journal,* 2(2), 46-56.
35. Seyed Esmaeili, F.S., Rostamy-Malkhalifeh, M., & Hosseinzadeh Lotfi, F. (2020). Two-stage network DEA model under interval data. *Mathematical Analysis and Convex Optimization,* 1(2), 103-108.
36. Seyed Esmaeili, F.S., Rostamy-Malkhalifeh, M., & Hosseinzadeh Lotfi, F. (2021). A hybrid approach using data envelopment analysis, interval programming and robust optimisation for performance assessment of hotels under uncertainty. *International Journal of Management and Decision Making*, 20(3), 308-322.
37. Seyed Esmaeili, F.S., Rostamy-Malkhalifeh, M., & Hosseinzadeh Lotfi, F. (2022). Interval network Malmquist productivity index for examining productivity changes of insurance companies under data uncertainty: a case study. *Journal of Mathematical Extension, 16*(8), 9.

38. Shahkooeei, M., & Rezai Balf, F. (2022). Attractiveness and Progress in Integer-Valued Data Envelopment Analysis. *Fuzzy Optimization and Modeling Journal*, 2(4), 65-74.
39. Singh, A. K., Sahu, R., & Bharadwaj, S. (2010). Portfolio evaluation using OWA- heuristic algorithm and data envelopment analysis. *The Journal of Risk Finance*, 11(1), 75-88.
40. Talbi, E. G., (2009). Metaheuristics: from design to implementation. *New York: John Wiley & Sons.*
41. Tarnaud, A. C., & Leleu, H. (2018). Portfolio analysis with DEA: Prior to choosing a model. *Omega*, 75, 57-76.
42. Tavana, M., Keramatpour, M., Santos-Arteaga, F. J., & Ghorbaniane, E. (2015). A fuzzy hybrid project portfolio selection method using data envelopment analysis, TOPSIS and integer programming. *Expert Systems with Applications,* 42(22), 8432-8444.
43. Vidal, C. J., & Goetschalckx, M. (1997). Strategic production-distribution models: A critical review with emphasis on global supply chain models. *European Journal of Operational Research*, 98(1), 1-18.
44. Wong, P. Y., Leung, S. C., & Gilleard, J. D. (2013). Portfolio performance benchmarking with data envelopment analysis. *Asia-Pacific Journal of Operational Research,* 30(5), 1350011.
45. Xiao, H., Ren, T., Zhou, Z., & Liu, W. (2021). Parameter uncertainty in estimation of portfolio efficiency: Evidence from an interval diversification-consistent DEA approach. *Omega,* 103, 102357.
46. Yang, X., (2006). Improving portfolio efficiency: A genetic algorithm approach. *Computational Economics*, 28(1): p.1.
47. Zamani, L., Beegam, R., & Borzoian, S. (2014). Portfolio selection using Data Envelopment Analysis (DEA): A case of select Indian investment companies. *International Journal of Current Research and Academic Review*, 2(4), 50-55.
48. Zhou, Z., Gao, M., Xiao, H., Wang, R., & Liu, W. (2021). Big data and portfolio optimization: a novel approach integrating DEA with multiple data sources. *Omega*, 104, 102479.
49. Zhu, H., & Zhang, J. (2009). A credibility-based fuzzy programming model for APP. *International Conference on Artificial Intelligence and Computational Intelligence.* 1, 455-459. IEEE

Miraboalhassani, M., Movahedi Sobhani, F., & Mohammadi, E. (2022). Fuzzy Portfolio Optimization Using Credibility Theory: Multi-Objective Evolutionary Optimization Algorithms. *Fuzzy Optimization and Modelling Journal*, 3 (1), 41-63.

https://doi.org/10.30495/fomj.2022.1949727.1054

Received: 12 January 2022 Revised: 6 March 2022 Accepted: 7 March 2022