Start-up and Financial Technology

e-ISSN: 3090-661X
Vol 01 (2) 2025 p. 87-103

Corresponding author:
Arif Yelği
Email : arefyelghi@topkapi.edu.tr

Received 20 September 2025;
Accepted 2 October 2025;
Published 4 October 2025.

Conflict of interest statement: Author(s) reported no conflict of interest

DOI: http://doi.org/10.70764/gdpu-sft.2025.1(2)-7

A HYBRID NAKA-FA-PSO ALGORITHM WITH NAKAGAMI DISTRIBUTION FOR MULTI-OBJECTIVE PORTFOLIO OPTIMIZATION

Arif Yelği[1], Shirmohammad Tavangari[2]

[1]*Istanbul Topkapi University*
[2]*University of British Columbia*

ABSTRACT

Objective: This study aims to optimize portfolio allocation under cardinality constraints by maximizing expected return and minimizing risk, while addressing the NP-complete nature of the problem.

Research Design & Methods: A hybrid multi-objective optimization approach is proposed by combining Particle Swarm Optimization and Firefly Algorithm (PSO-FA) with Nakagami distribution to preserve solution diversity and achieve optimal results. The algorithms were applied to the OR-library dataset and executed 30 times for analysis and evaluation.

Findings: The experimental results demonstrate that the proposed algorithm outperforms existing methods in terms of accuracy, diversity, and stability. On the P5 test sample, the reported metrics were 2.76E-07 IGD, 7.43E-08 GD, and 2.94E-03 HV, with consistent improvements also observed in other test samples.

Implications & Recommendations: The findings suggest that the PSO-FA with Nakagami distribution can serve as an effective alternative for solving cardinality-constrained portfolio optimization problems, particularly in tackling NP-complete challenges in finance. Future research may extend its application to larger datasets and dynamic market conditions.

Contribution & Value Added: This study contributes by introducing a novel hybrid optimization framework (PSO-FA and Nakagami distribution) that enhances solution quality in portfolio optimization. The value added lies in its ability to balance return, risk, and solution diversity, offering new insights beyond existing approaches in the literature.

Keywords: Optimization, Portfolio Management, Swarm Intelligence, Metaheuristic, Distribution.

JEL codes: C61, C63, G11
Article type: research paper

INTRODUCTION

One of the biggest problems in finance and investment management is the cardinality portfolio management problem. Considering the cardinality, or the number of assets to be included, it is associated with the most appropriate selection for allocation of assets within a portfolio. Developing a portfolio of investments that optimizes returns while minimizing risks is the primary objective of portfolio management. Markowitz's mean-variance framework and other traditional approaches for portfolio optimization use an assumption that investment returns following a continuous distribution and focus on acquiring a profitable frontier based on expected returns and variances (Markowitz, 1952). The number of assets that need to be included in the portfolio or the cardinality, nevertheless, are not explicitly taken into consideration by these methodologies.

Investors with preferences or limits on the number of assets in their portfolio face the cardinality portfolio management challenge. This could be a result of factors such as transaction costs, liquidity restrictions, knowledge asymmetry, diversification demands, bounds on holdings, and cardinality constrained (CC). The problem is in identifying the best subset of assets for use in the portfolio that keeps to these restrictions and achieves the optimal risk-return trade-offs. The foundation of current financial issues has focused on the portfolio optimization problem by various scientists and researchers (Markovitz, 1959; Markowitz, 1952; Merton, 1969). The primary intention is that the expected (mean) return maximizes and the risk (variance) of the portfolio minimizes. Markowitz's portfolio optimization problem is called the Mean-Variance (MV) Model (Markowitz, 1952). In fact, the percentage of stocks that stockholders will buy, hold, and sell is predicted by (MV) portfolio optimization.

The model is an essential tool for maximizing returns and managing investment risks, including stocks and bonds. It enables investors to sell or reallocate underperforming assets and invest in more promising ones. It assumes that the risk action of investors is symmetric with respect to the two concepts of profit and loss, and the probability distributions are normal (Gaussian). The main problem is that it assumes that investors' risk behaviour in gain and loss is symmetrical and under the expected distribution probability. However, it is not practical in many cases, and many researchers, considering the weakness of the variance of the problem, have tried to present different general models that can have a more realistic and better performance against the demand of the investment market. Some of the models encompassed in this category are the semi-variance model (Markovitz, 1959), absolute deviation model (Konno and Yamazaki, 1991), Mean-CVaR model (Rockafellar and Uryasev, 2000), Mean-Variance model (Jorion, 1996), Nonlinear Futures Hedging model (Junhui et al., 2009), and others.

In particular, it can be summarized as follows. In POP, at least six issues have been discussed and overshadowed, transaction costs, liquidity restrictions, knowledge asymmetry, diversification demands, bounds on holdings, and cardinality constrained. In this study, cardinality constrained (CC) and portfolio optimization problem (POP) are called (CCPOP), which investors consider a limitation number of stocks instead of taking heed to all stock. In a case study, the CCPOP has been demonstrated to be an NP-Complete problem so far; many exact and heuristic methods have been proposed to solve the CCPOP. Several of these methods have an acceptable time, but the presented solutions cannot be said to be the best solutions.

Qu et al. (2017) proposed an algorithm based on MOEA/Din large-scale portfolio optimization, trying to reduce complexity by considering the two preselection procedures and removing the asset non-potential. Experiments showed that it improved in terms of time complexity and space. Silva and Silva (2023) proposed a hybrid multi-objective evolutionary algorithm for portfolio selection, integrating local search with non-dominated sorting. Their method improves convergence and the quality of Pareto-optimal solutions compared to existing approaches. The experimental results were demonstrated with different cardinality constraints and compared with the other algorithms in terms of proximity and diversity. Kalayci et al. (2020) proposed algorithm for solving cardinality constrained portfolio optimization (CCPO) problems inherited some crucial components from the genetic algorithm, artificial bee colony algorithm, and continuous ant colony optimization algorithm; these components are elitism mechanism, modification rate, and Gaussian formulation, respectively. Kalayci et al. (2017) proposed a well-ordered solution algorithm based on an artificial bee colony algorithm in the company of usefulness and indivisibility strategy for solving the cardinality constrained portfolio optimization problem (CCPOP).

Anagnostopoulos and Mamanis (2011) surveyed and compared with the successfulness of five state-of-the-art multi-objective evolutionary algorithms (MOEAs) on the mean-variance (CCPOP), which are applied and evaluated on large-scale datasets. Ruiz-Torrubiano and Suárez (2015) designed a memetic algorithm that integrates a genetic algorithm (GA) and quadratic programming, which was used to indicate and encode the problem of optimal portfolio selection with cardinality constraints (CC) and transaction costs. In their work, the combinatorial and the continuous optimization sides of the problem are steered independently and show several regularization mechanisms. Zhao et al. (2021) proposed a particle swarm optimization algorithm with multiple parallel evolutions (MPCoPSO) for multiple objectives in their strategy encoding, return risk ratio heuristic, and bi-directional local search defined, which leads to obtaining feasible solutions.

In some works, multi-objective optimization, implementation on the portfolio problem has shown proper results (Estrada-Padilla et al., 2023; Morteza et al., 2023; Wang et al., 2022). Several multi-objective optimization algorithms have demonstrated strong performance on intricate problems, including the Non-dominated Sorting Genetic Algorithm (NSGA) (Deb et al., 2002; Lv et al., 2024), Multi-objective Grey Wolf Optimizer (MOGWO) (Mirjalili et al., 2016), Multi-objective Multi-Verse Optimizer (MOMVO) (Qin et al., 2024; Yelghi, 2024), and Multi-objective Artificial Bee Colony Algorithm[24]. Designed to increase convergence, diversity, and robustness in challenging optimization problems, the article presents DTDP-EAMO, a dual-time dual-population multi-objective evolutionary algorithm (Song et al., 2024).

The effectiveness of all of these algorithms in tackling the multi-objective problem has been demonstrated. Then an inquiry may come up. Still, a new algorithm (or algorithms) is (are) needed. According to the No Free Lunch (NFL) algorithm (Wolpert and Macready, 1997; Yang, 2020; Yelği, 2025), there is no single algorithm that can solve all problems. Thus, NFL theory offers the creation of new algorithms or the improvement of already existing ones.

The authors don't take convergence guarantees into consideration or provide theoretical convergence guarantees for the real Pareto front. The nature of the problem, algorithm design, and decision parameters all have an important effect on the quality of the answers that are provided. Their research demonstrates that there is no guarantee that the algorithm will discover all or almost all optimal options. Additionally, they tried numerous methods to estimate the Pareto front and have made an effort to present a number of optimal exchange strategies in order to get the true Pareto front. However, the solutions found may be suboptimal or come short of covering the full Pareto front due to the complexity and ambiguity of real-world problems.

The objective of this study is threefold. First, it aims to develop and evaluate the NAKA-FA-PSO algorithm, which integrates the Nakagami distribution with the Firefly Algorithm and Particle Swarm Optimization. Second, the study seeks to compare the performance of the proposed NAKA-FA-PSO algorithm with other existing algorithms using real datasets. Finally, it intends to analyze and illustrate the findings to highlight the impact and effectiveness of the proposed approach.

LITERATURE REVIEW

Particle Swarm Optimization (PSO)

Single objective Particle swarm optimization (PSO) (Kennedy and Eberhart, 1995; Yelghi et al., 2024; Yelghi & Köse, 2018) is a swarm intelligence-based algorithm that imitates birds flocking to solve NP problems. Particles in PSO search try to find a global optimum in search space. Each particle proceeds based on the Previous footprint, the global best leader, and a little randomly. The particles move on until the termination condition is satisfied. The updated velocity vector for each particle i is

$$V_i^{(k+1)} = wv_i^{(k)} + c_1r_1(p_{best,i}^{(k)} - S_i^{(k)}) + c_2r_2(g_{best}^{(k)} - S_i^{(k)}) \quad (1)$$

$x_i^{t+1}=x_i^t+V_i^{(k+1)}$ (2)

k is number of iteration.

$v_i^{(k)}$ is current velocity.

w is inertia weight.

c_1, c_2 are the individual and social acceleration constants respectively.

r_1, r_2 are the random values in the range [0, 1].

$b_{best,i}$ is the personal best position.

g_{best} is the global best position.

Firefly Algorithm (FA)

The firefly algorithm (FA) (Chang et al., 2000; Qin et al., 2024; Veldhuizen and Lamont, 1998) is a swarm intelligence-based algorithm and has been demonstrated to be applicable in solving NP problems, particularly multimodal problems where the objective landscape can have many local maxima or minima. This algorithm was developed by Xin-She Yang in 2008. It is based on the light scattering behavior of fireflies. In this study, we know that two algorithms are common in finding the optimal value, and we can say that Firefly is a class of PSO. Firefly's heavy-tailed nature of Levy flights allows for occasional long jumps, which can help the algorithm explore a wider search space and potentially discover better solutions that are farther away. To get the best results, we use the Nakagami distribution instead of Levy flights. In FA, a firefly or a solution to an optimization problem is proposed as the position in the search space. The movement formula is the position vector x_i of i'thfirefly at generation t.

$x_i^{t+1}=x_i^t + B_0 e^{-Gamma*r_{i,j}^2}(x_j^t - x_i^t)+\alpha \in_i^t$ (3)

t is number of interation.

i is number of firefly in population.

x_i is Position of firefly i.

B_0 is the attractiveness amplification factor.

Gamma is Light absorption coefficient.

a is the randomization factor.

$\in$ is Random vector used for movement.

The proper firefly is picked out from the population of (n) firefly at each generation. The firefly moves on until the termination condition is satisfied. Another feature of this algorithm is nonlinearity, which can be divided into several independent swarms in the search space. Recent studies (Chang et al., 2000; Coello and Cortes, 2005; Veldhuizen and Lamont, 1998; Zitzler and Thiele, 1999) have introduced novel optimization methods, combining heuristic algorithm development with financial applications. These approaches improve solution quality and efficiency while addressing complex portfolio and investment problems.

Nakagami Distribution

The Nakagami distribution is a probability distribution, It was first proposed in 1960.It is a method developed for small-scale fading modeling, and it is one of the most popular distributions for modeling and is used in wireless signal and radio wave propagation. The two parameters with set balancing values defined the height, steepness, and concaveness of the probability density curve.

The probability density function (PDF) formula is:

$PDF = 2\left(\frac{\mu}{w}\right)^{\mu}\frac{1}{\Gamma(\mu)}x^{(2\mu-1)}e^{\frac{-\mu x^2}{w}}$ (4)

μ is the shape parameter.

$\Gamma(\mu)$ Gamma function evaluated at μ.
x represents the random variable being evaluated
ω is the scale or spread parameterwhere ($\omega > 0$ for all $x > 0$) and directs the extension of the distribution.

Portfolio Optimization Problem

Mean-Variance (MV) was proposed (Coello and Cortes, 2005; Markowitz, 1952; Zitzler and Thiele, 1999) as the process of risk indicated as variance, opposed to expected return. Optimal portfolio selection could be mapped to a quadratic type optimization problem which tries to achieve maximizes the portfolio return objective and minimizes the portfolio risk objective.

N indicates the number of the asset to invest, μ_k indicates the average return of the i'th stock x_i indicates the weight variable which allocated to i'th stock and σ_{jk} indicates the covariance between to the i'th and k'th stocks
The formula return value is defined in the following

$$r = \sum_{i=1}^{N} x_i \mu_i \qquad (5)$$

And the risk of the portfolio is given as follows:

$$\sigma^2 = \sum_{i=1}^{N} \sum_{k=1}^{N} x_i x_k \sigma_{ik} \qquad (6)$$

MV could be expressed as minimizing the $\sigma^2$with regard to r. Therefore, the formulation is written as:

$$\min \sum_{j=1}^{N} \sum_{k=1}^{N} x_j x_k \sigma_{jk} \qquad (7)$$

Subject to $\sum_{i=1}^{N} x_i \mu_i = r,$ (8)

$$\sum_{j=1}^{D} x_j = 1 \qquad 0 \le x_j \le 1$$

$$, j = 1, \dots, D$$

Some scientists improve and provide a multi-objective portfolio optimization problem (MOPOP) which is used in practice and real world. Two objectives could be defined for the model and given as:

$$\text{Max } r = \sum_{i=1}^{N} x_i \mu_i \qquad (9)$$

$$Min \sigma^2 = \sum_{i=1}^{N} \sum_{k=1}^{N} x_i x_k \sigma_{ik} \qquad (10)$$

s.t.
$\varepsilon_i q_i \le x_i \le \delta_i q_i$Lower and upper percentage of allocation for each stock

$$\sum_{i=1}^{N} x_i = 1 \qquad (11)$$

$$\sum_{i=1}^{N} q_i = K \qquad (12)$$

$$q_i \in \{0,1\} \qquad (13)$$

Eq.10 presents the sum of weight for each stock, and Eq.11 also shows the number of cardinalities K based on the Eq.12 which is decision variable . Eq.13 is constraint in decision variable. We used this model to solve the portfolio problem in this literature and attempted to provide an adaptable algorithm that gives the best result in any cardinality constraint.

The contribution of this work stems from the fact that it provides the Naka_FA_PSO algorithm for cardinality-constrained portfolio optimization. This study has taken steps to provide the selected quality of stock by increasing the search power and the diverse solutions in the search space. The PSO algorithm was redesigned to move the generated solutions to the best ones and escape from the local trap. This type of movement has been designed as a new modeling form with equilibrium near and far solutions from each other. The characteristics of the FA algorithm and Nakagami distribution, considering the concept of absorption, have been used to create the best solution that tries to maintain the coherence and dispersion of the solution.Integration of those algorithms gives an improved design that is capable of solving problems.

METHODS

In this study, a hybrid multi-objective optimization approach is proposed by combining Particle Swarm Optimization (PSO) and Firefly Algorithm (FA), reinforced with Nakagami distribution to maintain solution diversity and prevent premature convergence. The use of the Nakagami distribution allows the algorithm to better maintain the stochastic characteristics of potential solutions, thereby improving both exploration and exploitation during the optimization process. To evaluate the performance of the proposed hybrid algorithm (PSO-FA with Nakagami distribution), experiments were conducted using the OR-library benchmark dataset. Each algorithm was run in 30 independent executions to ensure the reliability and robustness of the results. The results were then analyzed and compared based on various performance metrics, such as accuracy, diversity, and stability, to validate the effectiveness of the proposed approach compared to existing algorithms.

Parameters and Objective Functions

In this study, Eqs. 9 and 10 are considered as the objective function. Furthermore, the cardinality size is 10, the lower bound of variables is 0, and the upper bound is 0.5. To select stocks, we define Pattern-set as a variable to select (1) and ignore (0) stocks, which is a binary list. In order to estimate the proper stocks based on the cardinality, we need to design an algorithm to estimate optimum solutions. The optimum solutions should provide spread and convergence of pareto fronts in space solutions.In this study, we seek to provide a solution to estimate suitable solutions, which are provided from a balanced and more optimal distribution, and also the proposed algorithm should provide a more reasonable solution in terms of execution time during execution in relation toaverage scale data.

Strategy and Algorithm Foundation

For solving, in the first step, cardinality constrained (CC) requires to consider. We perform this with two inner and outer strategies. In the inner strategy,with the help of stock data relative to each other, using the following formula.

$$\psi_i = \frac{\mu_i}{\frac{1}{D}\sum_{k=1}^{D} \sigma_{ik}} \tag{14}$$

here $\psi_i$indicates the return and risk relationship of the ithstock.D, μ, $and\ \sigma$ denote stock number, return values, and covariance, respectively.

The higher value shows the potential stock for selecting step. Roulette wheel algorithm based on the mentioned formula provides a set of elite and potential stock. To describe the detail of Roulette wheel algorithm: Suppose there are p individuals with fitness valuesf_1,f_2,f_3, ...,f_n the probability of those are as follows:

$$p_i=\frac{f_i}{\sum_{k=1}^{p} f_k} \tag{15}$$

High probability rather than low probability has a higher chance of being selected in the next generation. In the outer strategy, combined algorithms generate new solutions and then call one of the remove and add strategies to set the cardinality. There is a pattern set for valuing each stock. It decreases or increases according to the amount of the reward, which is performed with the help of the remove strategy and the add strategy. Apart from this strategy, the obtained non-dominated solution from the mutation operator, along with the reward value, also affects this vector.

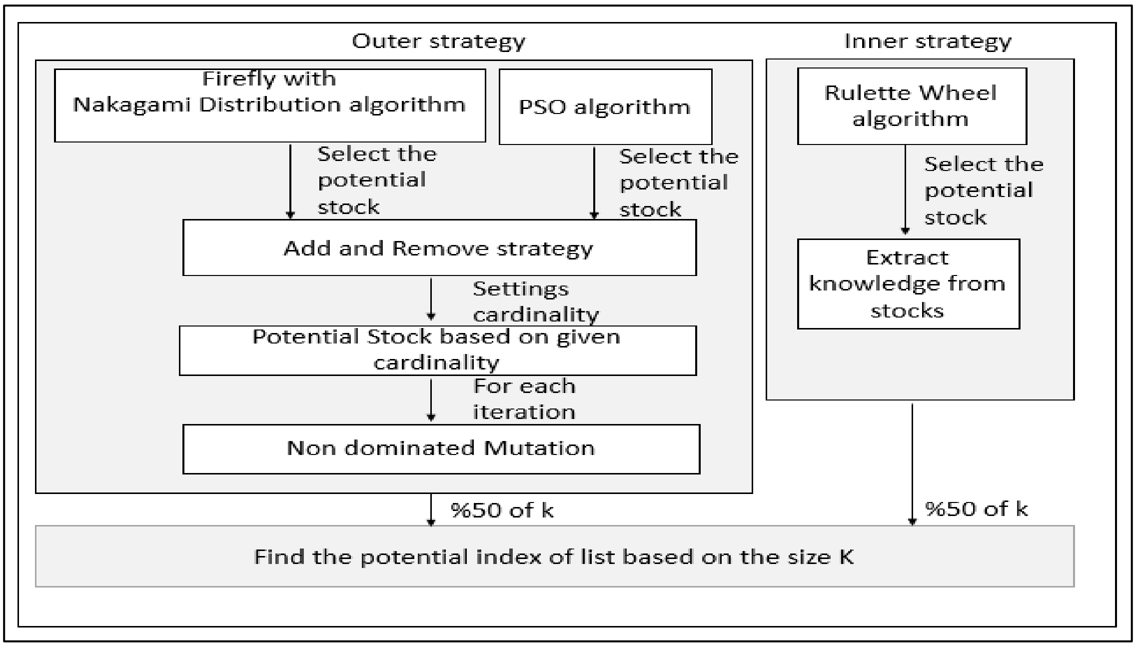


Figure1.Schema of NAKA-FA-PSO Algorithm for selecting stock.

Figure 1 illustrates the NAKA-FA-PSO Algorithm scheme for stock selection based on cardinality size. Here, there are two strategies used, namely the inner strategy and the outer strategy. In the outer strategy, two algorithms are applied: PSO and Firefly with Nakagami distribution. Meanwhile, in the inner strategy, the Roulette Wheel algorithm is used. All algorithms are developed and modified to select potential stocks. Each strategy accounts for half of the selected stocks. In the outer strategy, after selecting half of the stocks, a mutation operation is applied to increase diversity. Finally, a solution is provided based on the proportion of both strategies and the given cardinality measure.

Algorithm NaKa_FA_PSO

```
Definition:
NP              // population size
NRep            // archive size
k               // cardinal size

Begin
1:
      Rep = ∅,// archive
Fn = 0; // number of evaluation function
      Lb=ϕ; // lower bound
      Up=ϕ; // upper bound
      scale=|Up − Lb|;
Pattern_set=ϕ // General pattern
POS_select=ϕ  //  Binary trace for particle pattern
```

$y = 2$ // light absorption coefficient
$B_0 = 2$ // light intensity

2: ψ= Calculate the inner strategy as formula (14)and Calculate the outer strategy RouletteWheelSelection (ψ,k/2) as formula (15)

3: For i=1 to NP
4: Randomly initialize the particle variable POS_i,
5: Randomly initialize the velocity variable v_i,
6: Randomly initialize the selecting stock of particle variable POS_select_i,
7: idx= Calculate the formula B RouletteWheelSelection(ψ,k/2)
8: Set the elements of $\boldsymbol{POS\_Select_i}$ to one based on the idx(index)
9: [Position, Index]=**Call Algorithm Allocation_Number (k,lb,ub, pattern_set,** $\boldsymbol{POS\_Select_i}$)
10: normaliz_position = Evaluate the normal value for (Position, Index)
11: POS_i.Cost= Evaluate the fitness value for normaliz_position
12: End For

13: While not stop
14:For i=1 to NP
15: If rand<0.3
16: $V_i^{(k+1)} = wv_i^{(k)} + c_1 r_1 (p_{best}^{(k)} - POS_i^{(k)}) + c_2 r_2 (Leader1_{best}^{(k)} - POS_i^{(k)}) + c_2 r_2 (Leader2_{best}^{(k)} - POS_i^{(k)})$;
17: $POS_i^{t+1} = POS_i^t + V_i^{(k+1)}$;
18:Else
17: $r = mean(\sqrt{sum(x_i^t - Leader1_{best}^{(k)})^2} + \sqrt{sum(x_i^t - Leader1_{best}^{(k)})^2})$;
18: Pd= $2\left(\frac{\mu}{w}\right)^{\mu} \frac{1}{\Gamma(\mu)} x^{(2\mu-1)} e^{\frac{-\mu x^2}{w}}$; $w = 0.5$, $\mu = 1$
19: R= Generated Random (Pd, $p_{best}^{(k)}$);
20: $Beta = B_0 e^{-yr^2}$;
21: $Steps = (\sqrt{scale.R} * 10^{-3})$;
22: $x_i^{t+1} = x_i^t + Beta * (p_{best}^{(t)} - POS_i^{(t)}) + (p_{best}^{(t)} - Leader1_{best}^{(t)}) + Steps$;
23: End If
24: idx= Calculate the formula B RouletteWheelSelection(ψ,k/2)
25: Set the elements of $\boldsymbol{POS\_Select_i}$ to one based on the idx
26: Position=**Call Algorithm Allocation_Number (k,lb, ub,pattern_set,**$\boldsymbol{POS\_Select_i}$**)**
27: normaliz_position = Evaluate the normal value for (Position, Index) as (9,10) formula
28: POS_i.Cost= Evaluate the fitness value for normaliz_position
29:If rand<pm
30: xnew=**call Standard MutateGenetic**
31: Set the elements of xnewto one with based on the random idx
32: Position= **Call Algorithm Allocation_Number (k,lb, ub,pattern_set, xnew)**
33: normaliz_position = Evaluate the normal value for (Position,Index) as (9,10) formula
34: POS_i.Cost= Evaluate the fitness value for normaliz_position
35:End If
36: Rep0= Specify New Members for New Resository
37: Rep-old=[Rep0 Rep-old]
38: Rep-old= Save Non-DminatedMemebrs for the Repository
39: Update Grid and Check if Repository is Full

```
40:End For
41: End While
End
```

Figure 2 Algorithm NaKa_FA_PSO

Figure 2 indicates the pseudo-code for the main scheme. In the first step of the algorithm, the calculation involves determining the relationship between return and risk using Eq. 14 and establishing the probability distribution of gains based on Eq. 15. In the second step, start with solutions which is generated by uniform distribution lines (3-12). In the third step, combined algorithm and propose a new model generator for solutions. Based on the PSO model attempts to preserve the elite, which in leaders are integrated with local best and particles. It only works 0.3 of random and remains of it regards to firefly and Nakagami distribution. Nakagami distribution aids in generating diversity and spreading solutions. The distribution Nakagami designed the step of the firefly algorithm and sometimes proposed near and far solutions and maintained a variety of solutions in the search space. Firefly algorithm is modified and redesigned with Nakagami distribution. Absorption and attractiveness are the advantages of firefly, which maintains and controls of solutions in the potential space lines (14-23).

All described so far related to generated solutions, and then the Allocation_Number algorithm Figure 3, the Add algorithm Figure 4, and the Remove algorithm Figure 5 adjusted and satisfied the cardinality problem based on the solutions. In Figure 3, line 1, initialization is done, and in line 2, calculations are made based on the number of ones. From line 3 to 9, it is related to the removal of extra positions, which is related to the algorithm Remove Remove algorithm, and from line 10 to 17, they also show the positions that are lost, which does this to the Add algorithm. In Figure 4, from line 2 to 4, the selection process performs position indices, and from line 5 to 12, it corresponds to the selected shape modeled by a Gaussian distribution in order to add position indices. In Figure 5, from lines 2 to 4, the selection process performs position indices, and from lines 5 to 12, it relates to the selected pattern modeled by a Gaussian distribution in order to remove position indices. In order to trace the potential of each stock, it is used pattern_set, which is mentioned before. To set the reward ters could be 0.5 and -0.5 for add and remove strategy, respectively. This parameter determines the chance of success for each stock.

Algorithm Allocation_Number(k,xpos,lb, ub, pattern_set,POS_Select_i)

```
Definition:
Θ_1 // reward for Adding
Θ_2// reward for removing
    k // cardinal size
xpos // particle position

Begin
1:  POS_select= ϕ //  Binary trace for particle pattern
    SET= ϕ // preprocessed list
Add_size  // the number of selected index
Remove_size // the number of selected index
    POS= ϕ // particle position
Pattern_set= ϕ // General pattern
POS_select= ϕ //  Binary trace for particle pattern
2:  count= Calculates the number of element from POS_select  which is one
3:If count>k
4:        selectid=[ ];
5:Remove_size =count- k;
6:        idx1=Find the index of (POS_select_i(:)>0);
7:   [pattern_set,Selectid] = Call Algorithm Remove (SET, Remove_size,
Pattern_set, POS_select, Θ_2)
8:        set the index of POS_Select_i to zero based on the Selected position
```

```
        xpos=xpos.*POS_Selecti;
Else Ifcount<k
        selectid=[ ];
        ADD_size =k -count;
        Find the element of POS_selecti list which is greater than zero
        list1(idx1)=[ ];
    [pattern_set, Selectid] = Call Algorithm Add (SET, ADD_size, Pattern_set, POS_select, Θ1)
        set the index of POS_Selectito one based on the Selected position
        xpos=xpos.*POS_Selecti;
End IF
End
```

Figure3.Algorithm Allocation_Number.

```
AlgorithmAdd (SET, ADD_size, Pattern_set, POS_select, Θ1)
Begin
C=[ ]; Θ1 // the specified value of changing
If sum(Pattern_set)==0
     Selectid= generate random number index with considering Add_size
     pattern_set(Selectid)=1;
Else
    R=pattern_set.* Gaussian(0, 1);
    C=R/sum(R);
    idxd= Sort C by ascending
    idxd2=SET(idxd);
    Selectid=idxd2(1:ADD_SIZE);
    pattern_set(Selectid))=pattern_set(Selectid)+Θ1;
End If
End
```

Figure 4. Algorithm Add

```
AlgorithmRemove(SET, Remove_size, Pattern_set, POS_select, Θ2)
Begin
C=[]; Θ2 // the specified value of changing
If sum(Pattern_set)==0
    Selectid= generate random number index with
    considering Add_size
    pattern_set(Selectid)=1;
Else
    R=pattern_set.* Gaussian(0, 1);
    C=R/sum(R);
    idxd= Sort C by descending
    idxd2=SET(idxd);
    Selectid=idxd2(1:REVMOVE_SIZE);
    pattern_set(Selectid)=pattern_set(Selectid)+Θ2;
End If
End
```

Figure 5. Algorithm Remove

RESULT

Experimental study

a. Implementation and Test problem

In order to confirm the performance and efficiencies of the proposed algorithm, the statistical analysis is performed by utilizing Excel 2013. Competitive test problems were used in this study. The problem is five publicly easy-reach benchmark problems presented in OR-Library (Yang, 2020). The details are mentioned below, and specified stock numbers are reported in Table 1.

1. Hong Kong Hang Seng: The Hang Seng Index (HSI) is the primary stock market index for the Hong Kong Stock Exchange (HKEX). It represents the performance of the 50 largest and most liquid companies listed on the HKEX. The Hang Seng dataset would include historical data on the HSI's constituents, enabling analysis of stock prices, trading volumes, and other relevant financial indicators for these companies.
2. German Dow Jones Indices (DAX) 100: The 100 largest and most frequently traded companies on the exchange in Frankfurt are expressed by the DAX, which is the principal stock market index in Germany. It would be feasible to investigate these 100 companies's stock prices, market capitalizations, volume of trading, and other financial variables employing historical data gathered from the DAX dataset, which would be accessible.
3. The 100 largest companies listed on the London Stock Exchange constitute the British Financial Times Stock Exchange (FTSE) 100, an index of the stock market. These components' historical data would be incorporated into the FTSE 100 dataset, enabling it feasible to investigate their stock prices, market capitalizations, trading volumes, and other financial variables.
4. Standard & Poor's (S&P) 100 in the United States of America: The S&P 100, occasionally named as the OEX, is an index of the stock market that evaluates the performance of 100 significant, renowned American businesses operating in different industries. The past information for these companies stock prices, market capitalizations, volume of trading, and other financial indicators would be incorporated in the S&P 100 dataset.
5. The Nikkei 225, which reflects the 225 stocks that are traded the most often on the Tokyo Stock Exchange, is the principal index of the stock market in Japan. These 225 constituents' past information from the Nikkei dataset would allow it feasible to investigate their stock prices, market capitalizations, trading volumes, and other financial factors.

Table 1. Test problem from OR-Library

Stock Market Index Datasets	Name (N)	Stocks Number
Hong Kong Hang Seng	P1	31
German DeutscherAktienindex (Dax) 100	P2	85
British Financial Times Stock Exchange (FTSE)100	P3	89
United States Standard & Poor's (S&P) 100	P4	98
Japanese Nikkei Stock Average (Nikkei)-225	P5	225

b. Performance Metrics

In order to prove the efficiency of the NAKA_FA_PSO Fa algorithm, three well-known indicators are used and compared with other algorithms. These indicators are used to compare the distribution, spread, and proximity of the predicted value. Scientists and practitioners are able to assess the effectiveness of different multi-objective optimization algorithms and techniques using GD, IGD, and HV. These metrics offer a systematic method to evaluate and rank different approaches by evaluating the quality, diversity, and coverage of solutions. Each indicator is defined in detail below.

The Generational Distance (GD) metric provided by Veldhuizen and Lamont (1998); Yelği (2025) calculates the distance of the predicted values from the Pareto Front. In Eq.16, d_i is the distance calculated by Euclid between each solution in the obtained front and its nearest neighbor in the reference Pareto front, and n is the total number of solutions in the obtained front.

$$GD = \frac{\left(\sum_{i=1}^{n} d_i^2\right)^2}{n} \qquad (16)$$

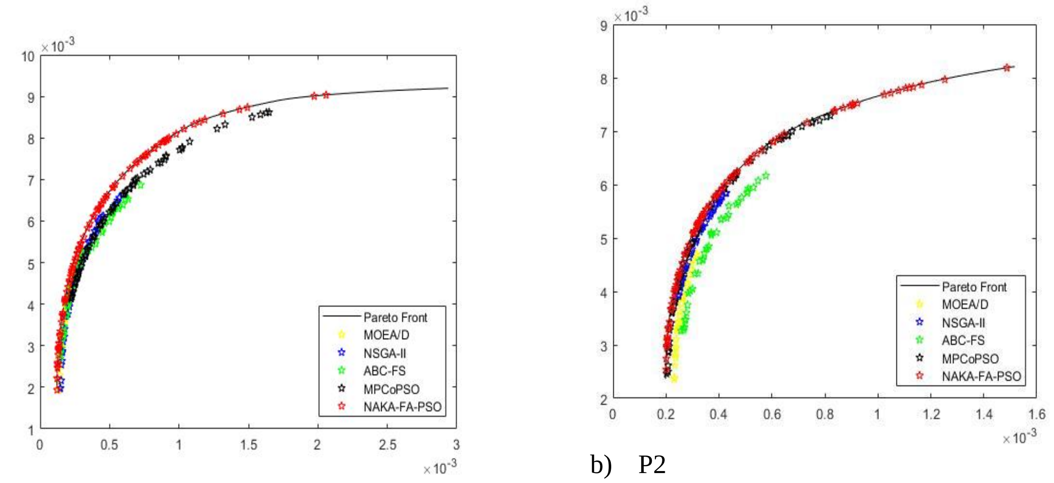
From other indicators, Inverted Generational Distance (IGD) indicator shows the diversity and convergence of the solution (Yelghi, 2024). In Eq. 17P is a group of uniformly distributed reference points selected from the Pareto optimum front, P is the objective values of a set of non-dominated solutions obtained by any algorithm, and d(x,y) is the Euclidean distance between points x and y. IGD evaluates the degree of convergence and diversity of the solution set P by measuring the average minimum distance between each point in P and those in P.

$$IGD(p,p^*) = \frac{\sum e \in P^* \min_{y \in p} d(x,y)}{|P^*|} \qquad (17)$$

Another quality indicator, Hypervolume (HV) can be used to measure and presents that approximates the Pareto front well with regard to both diversity and convergence. In Eq. 18,the area covered by a solution set P with consideration for a set of specified reference points R in the objective space is referred to as the solution set's HV value (P) (Yelghi et al., 2024).

$$HV(P,R) = \lambda(H(P,R)) \qquad (18)$$

Where H(P,R)={$z \epsilon Z$ | $\exists x \epsilon P, \exists r \epsilon R$: $f(x) \leq x \leq r$} and λ is Lebesgue measure



a) P1

b) P2

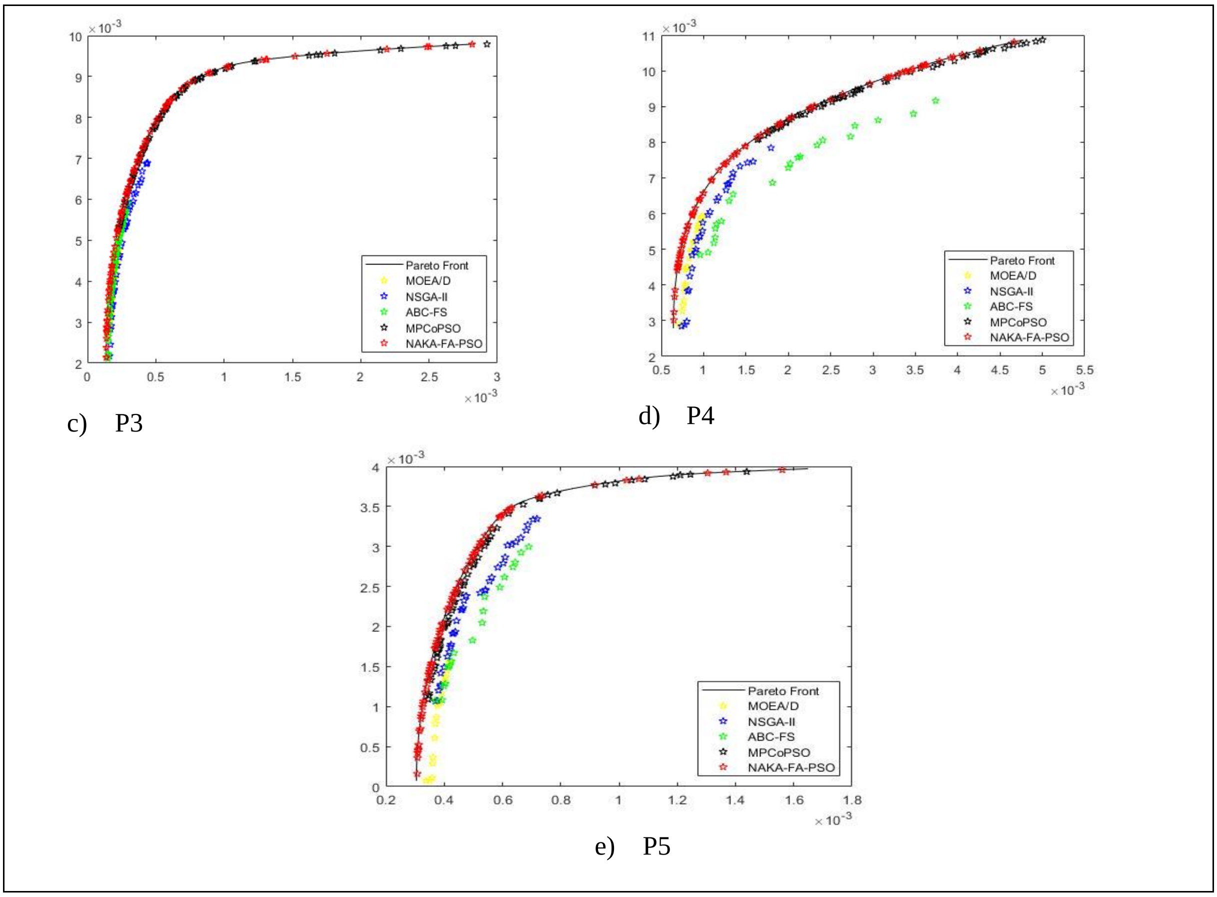


Figure 6. The convergence result of algorithms on five datasets a) P1 b) P2 c) P3 d) P4 e)P5

Table 2. The IGD index value for each algorithm

	IGD									
	NAKA-FA-PSO		MPCoPSO		ABC-FS		NSGA-II		MOEA /D	
N	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
P1	**5.92E-06**	3.64E-12	**2.56E-06**	1.22E-12	3.22-05	9.12-07	6.20-06	6.25-08	2.35-06	6.25-12
P2	**5.30E-08**	3.81E-11	5.30E-07	2.37E-10	4.61-07	7.14-12	3.22-04	1.95-05	3.68-07	1.81-14
P3	**6.53E-07**	4.76E-12	4.62E-06	4.68E-11	6.51-06	9.12-10	9.81-04	4.52-10	3.94-06	2.36-13
P4	**3.78E-07**	5.12E-12	5.78E-06	3.84E-11	7.14-05	6.51-07	1.84-05	6.35-06	4.25-06	9.17-13
P5	**2.76E-07**	5.55E-12	4.81E-06	2.31E-10	6.25-04	3.23-06	6.59-05	6.15-09	2.67-06	5.61-12

Table 3. The GD index value for each algorithm

	GD									
	NAKA-FA-PSO		MPCoPSO		ABC-FS		NSGA-II		MOEA /D	
N	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
P1	3.92E-07	3.64E-16	3.93E-07	5.76E-08	4.62-06	6.16-05	6.32-07	7.14-07	6.91-07	3.66-10
P2	**4.84E-09**	3.81E-12	4.35E-08	3.56E-09	6.25-08	7.12-10	9.54-05	6.23-06	3.25-08	7.11-09
P3	**4.71E-08**	4.76E-15	2.85E-07	3.34E-11	7.14-07	4.16-11	1.62-05	9.11-09	7.14-07	8.05-10
P4	**5.16E-08**	5.12E-13	9.74E-07	3.93E-10	2.35-06	6.98-09	6.74-06	9.08-07	6.85-07	6.14-09
P5	**7.43E-08**	5.55E-14	2.74E-07	4.76E-08	7.12-05	4.35-05	6.32-06	7.52-08	9.32-07	7.21-10

Table 4. The HV index value for each algorithm

	HV									
	NAKA-FA-PSO		MPCoPSO		ABC-FS		NSGA- II		MOEA /D	
N	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
P1	**4.78E-02**	2.94E-04	4.98E-03	4.76E-04	6.25-05	4.62-04	4.74-04	6.84-04	7.36-08	6.41-04

P2	**6.14E-03**	6.48E-03	3.63E-03	5.56E-04	4.25-05	6.74-05	9.14-03	6.35-03	6.17-07	4.35-03
P3	**8.16E-03**	3.36E-04	1.82E-04	6.34E-03	3.91-06	8.26-03	7.25-04	7.84-04	9.18-07	7.45-04
P4	**3.20E-02**	8.54E-02	9.86E-03	8.93E-04	2.57-05	9.74-04	6.14-03	1.17-04	8.19-08	3.69-04
P5	**2.94E-03**	1.98E-04	6.47E-04	6.76E-06	6.64-06	4.67-05	6.17-04	9.82-04	5.36-07	4.78-03

Table 5. Time Complexity

Algorithms	**Time Complexity**
MOEA/D	O(MNT)+3S
NSGA-II	O(MN^2)+3S
ABC-FS	O(MNEOA)+3S
MPCoPSO	O(MN^2)+ 4S
NAKA-FA-PSO	O($M(U+N^2)$)+3S

DISCUSSION

In this work, ten assets (with a predefined cardinality size) have been used to evaluate the proposed method together with other comparative algorithms. To solve the mean-variance portfolio optimization problem (POP) with cardinality constraints, the experimental process consists in thirty independent runs and 50,000 function evaluations (NFEs). Standard benchmark datasets taken from the OR-library have been applied to evaluate the algorithms.

Three performance measures– IGD (inverted generational distance), GD (generational distance), and HV (hypervolume)–have been used to assess the quality of the acquired solutions. Tables 2, 3, and 4's significant principles show that the suggested method surpasses all other ones. HV assesses how well the produced solutions cover the objective space while IGD and GD offer insights into how closely the generated solutions approximate the true Pareto front.

While the MPCoPSO algorithm has produced rather close results, the statistical analysis results–shown in Table 2–show that the proposed method has attained better performance than other algorithms. Lower values in this measure point to better performance. Table 3 shows that among the datasets p2, p3, p4, and p5, the mean values have been the best together with reasonable deviations. Furthermore, the MPCoPSO method has produced rather more consistent results with little variations. Table 4 shows that a better solution with more coverage of the solution space comes from a higher HV value. As can be observed across all test cases the suggested method has effectively given strong coverage and convergence. Furthermore displaying competitive performance is MPCoPSO, the second-best performing method. Based on these evaluation criteria, the comparative study shows generally that the suggested algorithm shows more robustness and accuracy than the other ones.

Figure 6 shows on several datasets (P1 to P5) the convergence of several algorithms on the Pareto front. This figure clearly shows that the NAKA_FA_PSO method is intended to produce several solutions, so guaranteeing optimality and robustness. Either formally or informally, many research projects involve considering time complexity as a regular feature. Nevertheless, in this work the informal complexity analysis has been omitted since the variations in execution platforms produce different computational performances, resulting in it inappropriate for exact reporting.

Table 5 shows that the NAKA-FA-PSO method shows a relative superiority over other algorithms apart from the MOEA/D method. Regarding this: M stands for the number of objectives; N for the population size; T for the number of weight vectors; E for the number of employed, or firefly solutions; O for the number of onlooker solutions; A for the number of abandoned solutions; U for the mutation operator; S for the chosen strategy for the process of problem-solving. The results show generally that the suggested method is quite convergent, applicable, and dependable for the portfolio choosing problem. Using an HP LAPTOP-5J21O90K, all experiments were run on an Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80GHz with 8GB of RAM. MATLAB R2018b contained the algorithms under Windows 10.

CONCLUSION

In this paper, we proposed Nakagami distribution with a hybrid of FA and PSO to solve the cardinality constraint portfolio optimization problem. The proposed algorithm provides a proper solution for the success of the tradeoff between exploration and exploitation in obtaining the optimum solution and demonstrates how to keep convergence and dispersion in each iteration in solution space. The Proposed Algorithm and Compared Algorithms have been performed cardinality size with ten stocks with 30 run times and compared to each other. The experimental results showed that our proposed algorithm outperforms the others. The characteristics of the algorithm, including convergence, stability, dispersion, and reliability, have been demonstrated with statistical and metric methods. However, the proposed algorithm will have an proper performance compared to the growth of the data over time, and the morphology and complexity of the data will be future problems that should be investigated. In the future, we will try to provide a completely general approach for such NP-Complete problems that has flexibility and provides optimal solutions with certainty.

REFERENCES

Anagnostopoulos, K. P., & Mamanis, G. (2011). The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications, 38*(11), 14208–14217. https://doi.org/10.1016/j.eswa.2011.04.233

Chang, T.-J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research, 27*(13), 1271–1302. https://doi.org/10.1016/S0305-0548(99)00074-X

Coello, C. A. C., & Cortes, N. C. (2005). Solving Multiobjective Optimization Problems Using an Artificial Immune System. *Genetic Programming and Evolvable Machines, 6*(2), 163–190. https://doi.org/10.1007/s10710-005-6164-x

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197. https://doi.org/10.1109/4235.996017

Estrada-Padilla, A., Gómez-Santillán, C., Joaquín Fraire-Huacuja, H., Cruz-Reyes, L., Rangel-Valdez, N., Lucila Morales-Rodríguez, M., & José Puga-Soberanes, H. (2023). GRASP/Δ: An efficient algorithm for the multi-objective portfolio optimization problem. Expert Systems with Applications. *Expert Systems with Applications*, 211, 118647. https://doi.org/10.1016/j.eswa.2022.118647

Jorion, P. (1996). Risk2: Measuring the Risk in Value at Risk. *Financial Analysts Journal, 52*(6), 47–56. http://www.jstor.org/stable/4479959

Junhui, F., Weiguo, Z., Qian, L., & Qin, M. (2009). Nonlinear futures hedging model based on skewness risk and kurtosis risk. *Systems Engineering, 27*(10), 44–48.

Kalayci, C. B., Ertenlice, O., Akyer, H., & Aygoren, H. (2017). An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization. *Expert Systems with Applications*, 85, 61–75. https://doi.org/10.1016/j.eswa.2017.05.018

Kalayci, C. B., Polat, O., & Akbay, M. A. (2020). An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization. *Swarm and Evolutionary Computation*, 54, 100662. https://doi.org/10.1016/j.swevo.2020.100662

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95 - *International Conference on Neural Networks*, 4, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Konno, H., & Yamazaki, H. (1991). Mean-Absolute Deviation Portfolio Optimization Model and Its Applications to Tokyo Stock Market. *Management Science*, *37*(5), 519–531. https://doi.org/10.1287/mnsc.37.5.519

Lv, M., Wang, J., Wang, S., Gao, J., & Guo, H. (2024). Developing a hybrid system for stock selection and portfolio optimization with many-objective optimization based on deep learning and improved NSGA-III. *Information Sciences*, 670, 120549. https://doi.org/10.1016/j.ins.2024.120549

Markovitz, H. (1959). Portfolio selection: Efficient diversification of investments. *Yale University Press*.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7, 77–91.

Merton, R. C. (1969). Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, *51*(3), 247–257. https://www.jstor.org/stable/1926560

Mirjalili, S., Saremi, S., Mirjalili, S. M., & Coelho, L. dos S. (2016). Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, 47, 106–119. https://doi.org/10.1016/j.eswa.2015.10.039

Morteza, H., Jameii, S. M., & Sohrabi, M. K. (2023). An improved learning automata based multi-objective whale optimization approach for multi-objective portfolio optimization in financial markets. *Expert Systems with Applications*, 224, 119970. https://doi.org/10.1016/j.eswa.2023.119970

Qin, S., Zhang, S., Wang, J., Liu, S., Guo, X., & Qi, L. (2024). Multiobjective Multiverse Optimizer for Multirobotic U-Shaped Disassembly Line Balancing Problems. *IEEE Transactions on Artificial Intelligence*, *5*(2), 882–894. https://doi.org/10.1109/TAI.2023.3266187

Qu, B. Y., Zhou, Q., Xiao, J. M., Liang, J. J., & Suganthan, P. N. (2017). Large-Scale Portfolio Optimization Using Multiobjective Evolutionary Algorithms and Preselection Methods. *Mathematical Problems in Engineering*, *2017*(1). https://doi.org/10.1155/2017/4197914

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *The Journal of Risk*, *2*(3), 21–41. https://doi.org/10.21314/JOR.2000.038

Ruiz-Torrubiano, R., & Suárez, A. (2015). A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs. *Applied Soft Computing*, 36, 125–142. https://doi.org/10.1016/j.asoc.2015.06.053

Silva, Y. L. T. V., & Silva, N. E. F. (2023). A hybrid non-dominated sorting genetic algorithm with local search for portfolio selection problem with cardinality constraints. *Exacta*, *22*(3), 788–819. https://doi.org/10.5585/2023.22046

Song, Y., Han, L., Zhang, B., & Deng, W. (2024). A dual-time dual-population multi-objective evolutionary algorithm with application to the portfolio optimization problem. *Engineering*

Applications of Artificial Intelligence, 133, 108638. https://doi.org/10.1016/j.engappai.2024.108638

Veldhuizen, D. A. V., & Lamont, G. B. (1998). Multi-objective evolutionary algorithm research: A history and analysis. *In Department of Electrical and Computer Engineering*. Wright-Patterson AFB, OH: Air Force.

Wang, Z., Zhang, X., Zhang, Z., & Sheng, D. (2022). Credit portfolio optimization: A multi-objective genetic algorithm approach. *Borsa Istanbul Review*, *22*(1), 69–76. https://doi.org/10.1016/j.bir.2021.01.004

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

Yang, X. S. (2020). Nature-inspired optimization algorithms. *Academic Press*.

Yelghi, A. (2024). Estimation single output with a hybrid of ANFIS and MOPSO_HS. *Sakarya University Journal of Computer and Information Sciences*, *7*(1), 112–126. https://doi.org/10.35377/saucis...1414742

Yelghi, A., & Köse, C. (2018). A modified firefly algorithm for global minimum optimization. *Applied Soft Computing*, 62, 29–44. https://doi.org/10.1016/j.asoc.2017.10.032

Yelghi, A., Yelghi, A., Apan, M., & Tavangari, S. (2024). Computing intelligence in capital market (Studies in Computational Intelligence (Vol. 1154). *Springer Nature*.

Yelği, A. (2025). Survivor optimizer: A competitive strategy for enhanced search efficiency. *Ain Shams Engineering Journal*, *16*(9), 103561. https://doi.org/10.1016/j.asej.2025.103561

Zhao, H., Chen, Z.-G., Zhan, Z.-H., Kwong, S., & Zhang, J. (2021). Multiple populations co-evolutionary particle swarm optimization for multi-objective cardinality constrained portfolio optimization problem. *Neurocomputing*, 430, 58–70. https://doi.org/10.1016/j.neucom.2020.12.022

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257–271. https://doi.org/10.1109/4235.797969