International Journal of Applied Metaheuristic Computing
Volume 15 • Issue 1 • January-December 2024

Optimizing Stock Portfolio Using the Particle Swarm Optimization Algorithm and Assessing PSO and Other Algorithms

Samaneh Mohammadi Jarchelou
https://orcid.org/0009-0004-4076-6286
Department of Statistics, Islamic Azad University, North Tehran Branch, Tehran, Iran

Kianoush Fathi Vajargah
https://orcid.org/0000-0002-2633-3300
Department of Statistics, Islamic Azad University, North Tehran Branch, Tehran, Iran

Parvin Azhdari
https://orcid.org/0000-0001-5091-8555
Department of Statistics, Islamic Azad University, North Tehran Branch, Tehran, Iran

ABSTRACT

When it comes to making financial decisions, choosing stocks is crucial to building a successful portfolio. Stocks are evaluated according to lower risk, and the best stocks are chosen to produce assets that are then utilized to construct the portfolio. In this study, we have compared the integrated particle swarm method to four other algorithms for stock selection and optimization: the genetic algorithm, the Pareto search algorithm, the pattern search algorithm, and quadratic programming in the Matlab toolbox. Six particular stock firms are taken into consideration for this reason during a given time period. First, we will use the aforementioned Matlab toolbox techniques to conduct Markowitz's mean variance model. Additionally, the usual embedded particle swarm methodology and the penalty function approach will be used to create this model. The difference between the averages at ten distinct levels of predicted values will be examined in the next research based on the returns in the chosen portfolios. Statistical tests will be employed to differentiate noteworthy distinctions between the suggested approach and the alternative algorithms. MSC: 65K10,91B05.

KEYWORDS

Markowitz's Model, Quadratic Programming, Genetic Algorithm, Particle Swarm Algorithm, Pareto-Search Algorithm, Pattern-Search Algorithm

1. INTRODUCTION

Numerous models have been developed throughout the course of the twentieth century in an attempt to guide investigators toward the right approach to modelling investments. Sector diversity and stock portfolio optimization have evolved into ideas that are used as tools for decision-making and the growth of financial markets. Since Markowitz's model was released, it has been applied as a useful tool for stock portfolio optimization and has brought about a number of adjustments and advances in the way researchers view investing and stock portfolios. While it may appear straightforward to minimize risk and maximize investment returns, creating an ideal portfolio really involves a number of techniques. Markowitz presented contemporary portfolio theory as a mathematical formula that

DOI: 10.4018/IJAMC.362001

embodies a classical approach (Markowitz 1952). His mean-variance approach gives information about a portfolio's risk by displaying the average anticipated return and variance. A lot of individuals have attempted to expand and alter Markowitz's concept. In the area of stock portfolio optimization, several studies have been conducted using a variety of models as well as numerical and clever techniques. See Kalayci et al. (2019); Ponsich et al. (2012); Thakkar and Chaudhari (2021); Zanjirdar (2020), for an outline of some of these examples. Articles Aranha and Iba (2009); Bazrkar and Hosseini (2023); Chang et al. (2009); Coello et al. (2007); Erwin and Engel- brecht (2023); Lin and Liu (2008); Wright (2006) (also see reviewer articles and books in Goldberg (1989); Gunjan and Bhattacharyya (2022); Zanjirdar (2020)) discuss the use of genetic algorithms, or GA, in stock portfolio optimization. As you shall see below, GA may be used to solve the constrained optimization problem as it is currently integrated into the Matlab toolbox. Furthermore, this software has other algorithms that can address the Markowitz portfolio optimization issue, including pattern-search, Pareto-search, and quadratic programming, or QP for short. However, when it comes to the PSO algorithm, the Matlab toolbox has just one unconstrained optimization toolbox.

When there are no linear restrictions, pattern search automatically searches for a minimum based on an adaptive mesh that is aligned with the coordinate directions. Pattern-search (see the algorithm in Audet and Dennis Jr (2002)) locates a series of locations $x_0, x_1, \dots$ that converge to a minimal point.

The Pareto-search method iteratively looks for non-dominant points by applying pattern-search to a group of points. In every iteration, the Pareto-search meets all linear bounds and restrictions. The method should theoretically converge to locations around the actual Pareto front. See Custódio et al. (2011) for a discussion and proof of convergence, where the proof is used for Lipschitz continuous objective and constraint problems.

Recently, Kennedy and Eberhart (1995) presented a heuristic method called particle swarm optimization. While some PSO research has been done, virtually none of it addresses portfolio optimization for the mean-variance Markowitz model (Cura (2009)). For a complete overview of PSO applications in portfolio optimization, Thakkar and Chaudhari (2021) is a useful resource. Also, refer to recent advancements and applications of PSO in the works Bazrkar and Hosseini (2023); Erwin and Engelbrecht (2023); He and Huang (2014); Kuo and Chiu (2024); Liu and Li (2024); Song et al. (2023).

The suggested method's steps—data selection, data cleaning and preparation, objective function determination, stock portfolio selection based on GA, pattern search, Pareto search, and QP—are used in this article. These may be found in the Matlab toolbox, and the penalty function mentioned in Section 3 was used to create the PSO algorithm in Matlab. We employ non-parametric statistical tests to examine the numerical findings and present the optimal methods in Section 4. Using the Kruskal-Wallis test, the significance of the lack of differences between solutions produced by these algorithms has been investigated.

The paper presents a comprehensive analysis of portfolio optimization methods using various algorithms. Below are the main findings highlighted throughout the research:

1. Optimizing the stock portfolio using the penalty function method and PSO algorithm.
2. Optimization Algorithms Performance:
    - Algorithm Comparison: The study compares the performance of five algorithms: Particle Swarm Optimization (PSO), Quadratic Programming (QP), Genetic Algorithm (GA), Pareto Search, and Pattern Search.
    - Statistical Testing: The Kruskal-Wallis test was employed to statistically compare the performance of the algorithms against QP, showing no significant difference among the top-performing algorithms (PSO, Pareto Search, and Pattern Search) at a 1% significance level.
    - Effective Algorithms: PSO, QP, and Pareto search algorithms demonstrated superior performance in achieving optimal risk-return ratios compared to GA and Pattern Search.

2. STOCK PORTFOLIO OPTIMIZATION

In the early 1950s, Harry Markowitz established the basic portfolio model on which modern theory is based. He was the first to formally develop the concept of portfolio diversification. Markowitz's model has data or inputs, which are:

1. Expected return per share,
2. The standard deviation of the expected return as a criterion for determining the risk of each share,
3. Covariance, as a measure that shows the alignment between the returns of different stocks.

Markowitz's model was based on the indicators of expected return and risk and portfolio diversification, which is basically a theoretical framework for analyzing risk and return options. According to his theory, an efficient investment portfolio is one that has the highest return or the lowest risk for a certain level of return at a certain level of risk. In formulating his "risk-return" criterion, Markowitz paid special attention to the investment objective. Markowitz's standard mean-variance method for portfolio selection attempts to intercept a frontier, a continuous curve that shows the trade-off between the portfolio's return and risk. In short, the Markowitz optimization model is presented as follows:

$$\min \sigma^2 = \sum_{i=1}^{n} \sum_{i=1}^{n} w_i w_j \sigma_{ij}, \tag{1}$$

s.t.:

$$\sum_{i=1}^{n} w_i \mu_i \geq R_0, \tag{2}$$

$$\sum_{i=1}^{n} w_i = 1, \tag{3}$$

$$w_i \geq 0, i = 1, 2, 3, \ldots, n. \tag{4}$$

In the above model, n is the number of stocks, which indicates the dimensionality of the optimization in the portfolio, σ^2 stands for the variance of the portfolio, which generally refers to portfolio risk, μ_i shows the average return of shares $i = 1, \ldots, n$ over previous days $t = 1, \ldots, T$ and σ_{ij} is the covariance between returns of assets i and j, w_i is the weight of each asset i in the portfolio to be optimized, R_0 is a certain or desired level of return of the investor, and it is considered for the above model, thus, the answer obtained from the objective function is actually the optimal weight of stock selection. The returns of stock collected will be denoted by $R_1, \ldots, R_T$ and the return at day $t = 1, \ldots, T$. Solving this problem for values R_0 lying between $R_{\min} = \min\{R_1, \ldots, R_T\}$ and $R_{\max} = \max \{R_1, \ldots, R_T\}$ obtains all efficient portfolios. The return values R_0 are plotted along with a measure of risk (such as the standard deviation of the optimal weight, etc.), and then a shape called the efficient frontier is obtained.

This model (equations (1)-(4)) is a QP problem in a standard form. Today, a QP problem can be optimally solved using existing software toolbox (see Coello et al. (2007); Cornue- jols and Tütüncü (2006); Gilli et al. (2019)). Matlab software is one of the most powerful mathematical software that has wide applications in other fields as well. In order to solve the above relationship, you can use the algorithms embedded in the Matlab toolbox, such as pattern-search, GA, etc. We compare them with the QP solution in Matlab toolbox. Since the uniqueness of the solution and the convergence of the QP algorithm for problem (1)-(4) have been proven (e.g., see Nocedal and Wright (2006)), other algorithms (classical and intelligent methods) can be compared with it. For this purpose six stocks 'IBM','GOOGL','MSFT','AAPL','YHOO','AMZN' in 22 working days from StartDate='2014-Jan-01' to EndDate='2014-Feb-01' has been investigated. The minimum and maximum average stock returns in this period were $R_{\min} = -0.0018$ and $R_{\max} = 0.0142$, respectively.

3. RESEARCH MODEL AND ITS IMPLEMENTATION

In the Matlab toolbox, four methods, Pareto-search, QP, GA, pattern-search, are completely designed as a function to solve the model (1)-(4), while the particleswarm function in Matlab is only for unconstrained problem. So, we propose a standard PSO algorithm to solve Markowitz's model as follows:

3.1 Stock Portfolio Optimization Based on Particle Swarm Algorithm and Its Implementation

The particle swarm optimization algorithm was proposed for the first time by Eberhart and Kennedy (1995). In developing this method, it was inspired by the group flight of birds, the group swimming of fish, and their social life lives, which were formulated using a series of simple mathematical relationships. It is a population-based meta-heuristic optimization method that PSO can be apply to a wide range of problems (see Clerc (2013); Coello et al. (2007); Mercangöz (2021)). Like any meta-initiative method, this algorithm also starts by creating an initial random population, which is called a group of particles. The characteristics of each particle in the group are determined based on a set of parameters whose optimal values should be determined. In this method, each particle will represent a point in the problem solution space. Each of the particles has a memory, that is, it remembers the best position it reaches in the search space. Therefore, the movement of each particle takes place in two directions: first, to the best position that it has taken so far, and then to the best position that the particles have taken so far. Therefore, in this method, the change in position of each particle in the search space will be influenced by its own experience and knowledge, as well as that of its neighbors. See reference Mercangöz (2021) for a good and appropriate book on optimal portfolio optimization using the PSO algorithm. In this research, with the aim of using the well-known PSO algorithm, by entering the penalty coefficient in the objective function, it was tried to include both risk and return criteria in the objective function.

Definition 3.1 (Nesterov (2018)) *A continuous function P is called a penalty function for a closed set $Q \subseteq \mathbb{R}^n$ if*

1. $P(x) = 0$ for all $x \in Q$.
2. $P(x) > 0$ for all $x \notin Q$.

Therefore, the presented model can be rewritten as:

$$\min Z = \sum_{i=1}^{n}, \sum_{j=1}^{n} w_i w_j \sigma_{ij} + P(w_1, \ldots, w_n) \tag{5}$$

s.t.:

$$\sum_{i=1}^{n} \omega_i = 1 \tag{6}$$

$$0 \leq w_i \leq 1, i = 1,2,3,\ldots,n \tag{7}$$

where,

$$P(w_1, \ldots, w_n) = \max\left\{0, 1 - \frac{\sum_{i=1}^{n} w_i \mu_i}{R_0}\right\},$$

is the penalty function for the following set

$$Q = \left\{w = (w_1, \ldots, w_2), \sum_{i=1}^{n} w_i = 1, \sum_{i=1}^{n} w_i \mu_i \geq R_0\right\} \tag{8}$$

The above relationship is a nonsmooth constrained optimization relationship, and that we use the standard PSO algorithm in the search spaces (6) and (7) to solve it as follows:

Particle position updates and transformations /velocity updates can be represented by Eqs. (9) and (10), respectively. These are the two equations that govern PSO:

$$\nu W_i^d[t+1] = \omega \nu W_i^d[t] + c_1 \times r_1 \times \left(\text{P}best\, W_i^d[t] - W_i^d[t]\right) + c_2 \times r_2 \times \left(\text{G}best\, W_i^d[t] - W_i^d[t]\right), \tag{9}$$

$$W_i^d[t+1] = W_i^d[t] + \nu W_i^d[t+1], \tag{10}$$

such that $W_i^d[t] = \left(w_i^1[t], \ldots, w_i^d[t]\right)$ the positions (along the d dimension, which is equal to the number of stocks) of particle i at time "t" (previous position) and "$t+1$" (current position); $\nu W_i^d[t+1]$ and $\nu W_i^d[t]$ are the velocity of particle i (along dimension d) at time "t" and "$t+1$", respectively. P$best\, W_i^d[t]$ is the personal best particle position for the ith particle (along the d dimension) at time t. G$best\, W_i^d[t]$ is the global position of the best particle at time t. The random numbers r_1 and r_2 are two distinct random uniform numbers between 0 and 1 to add a random property to the particle move. Coefficient ω is the inertia weight used to obtain a trade-off between global exploration (toward the global best) and local exploitation (toward the personal best). Accelerated coefficients c_1 and c_2 are two constant positive numbers known as learning factors and are called cognitive learning and social learning factors Eberhart and Kennedy (1995), respectively. The results are calculated based on a "merit criterion" after each time period. Over time, particles accelerate towards particles that have a higher fitness criterion and are in the same communication group. Although each method works well for a range of problems, this method has shown much success in solving continuous optimization problems. The following algorithm was used to solve models (5)-(7):

1. Suppose $t = 0$. Randomly generate an initial population with a population of M and corresponding velocities. For each $1 \leq i \leq M$, put P$best\, W_i^d[0] = W_i^d[0]$, then calculate the value of G$best$ $W_i^d[0]$ using equation (11).

$$\text{G}best\, W_i^d[0] = \underset{W_i^d[0], 1 \leq i \leq M}{\operatorname{argmin}} Z\left(W_i^d[0]\right). \tag{11}$$

2. According to relations (9) and (10), update the value of the speed and position of the particles. Normalize $W_i^d[t]$ in order to place it in the search spaces (6) and (7), that is,

$$W_i^d[t] \leftarrow \left(\frac{w_i^1[t]}{\|W_i^d[t]\|}, \ldots, \frac{w_i^d[t]}{\|W_i^d[t]\|}\right),$$

where $\|W_i^d[t]\| = \sqrt{(w_i^1[t])^2 + \cdots + (w_i^d[t])^2}$.

3. According to equation (12), for each $1 \leq i \leq M$, compare the objective function value of particle $W_i^d[t+1]$ with the objective function value of P*best* $W_i^d[t]$. If the objective function value of this particle is less than the objective function value, then $W_i^d[t+1]$ is the best personal experience of the *i*th particle in step $t+1$; otherwise, the best personal experience of the *i*th particle in step $t+1$ is the same as the best personal experience of the *i*th particle in step *t*.

$$Pbest\,W_i^d[t+1] = \begin{cases} W_i^d[t+1], & Z\left(W_i^d[t+1]\right) < Z\left(Pbest\,W_i^d[t]\right), \\ Pbest\,W_i^d[t], & otherwise. \end{cases} \tag{12}$$

4. According to (13), in step $t+1$, select the best personal experience of a particle as G*best* W_i^d $[t+1]$, which has the lowest objective function value of $Z(\cdot)$ among the best personal experiences of all particles.

$$\mathrm{G}best\,W_i^d[t+1] = \underset{\mathrm{P}best W_i^d[t+1], 1 \leq i \leq M}{\operatorname{argmin}} Z\left(\mathrm{P}best\,W_i^d[t]\right). \tag{13}$$

5. Check the completion condition of the algorithm. If the number of iterations of the algorithm has reached the maximum number of iterations, declare $Z\left(\mathrm{G}best\,W_i^d[t+1]\right)$ as the optimal or suboptimal solution in point $W_i^d[t+1]$ and exit the program; otherwise, put $t \leftarrow t+1$ and go to step 2.

The initial population was considered to be $M = 20$ particles, and the algorithm stops after at most 100 iterations. The cognitive learning and social learning factors were 1.4962, and the inertia weight was 0.7298. The initial velocity of the particles was considered to be zero (in Clerc and Kennedy (2002), it is shown that these parameters create the conditions (velocity) of convergence, and they showed that system explosions can be controlled simply by using these proper values of the constriction coefficient).

4. COMPARISON OF ALGORITHMS WITH DIFFERENT INPUT INFORMATION IN FORMING THE BASKET

In order to compare the results of several algorithms and determine the best algorithm among them, a strong and reliable tool is needed. We use the method of non-parametric statistical tests. These methods are mathematical guidelines for hypothesis testing including the Kruskal-Wallis test, Wilcoxon sign rank test, Friedman test, etc., (see also Derrac et al. (2011)), and their most obvious feature is that they do not require any assumptions about the probability distribution and the tested variables. We compare each algorithm with the QP algorithm that is embedded in the Matlab toolbox for model (1). For this comparison, the Kruskal-Wallis test was used in the Matlab. We study and examine the results of solving each of these algorithms for the ten expected values as follows:

$$R_0 = 0.0100, 0.0105, 0.01100, 0113, 0.0119, 0.0124, 0.0128, 0.0133, 0.0138, 0.0142. \tag{14}$$

For instance, the calculated values of the stock portfolios at the expected return $R_0 = 0.0138$ with each algorithm were obtained as follows:

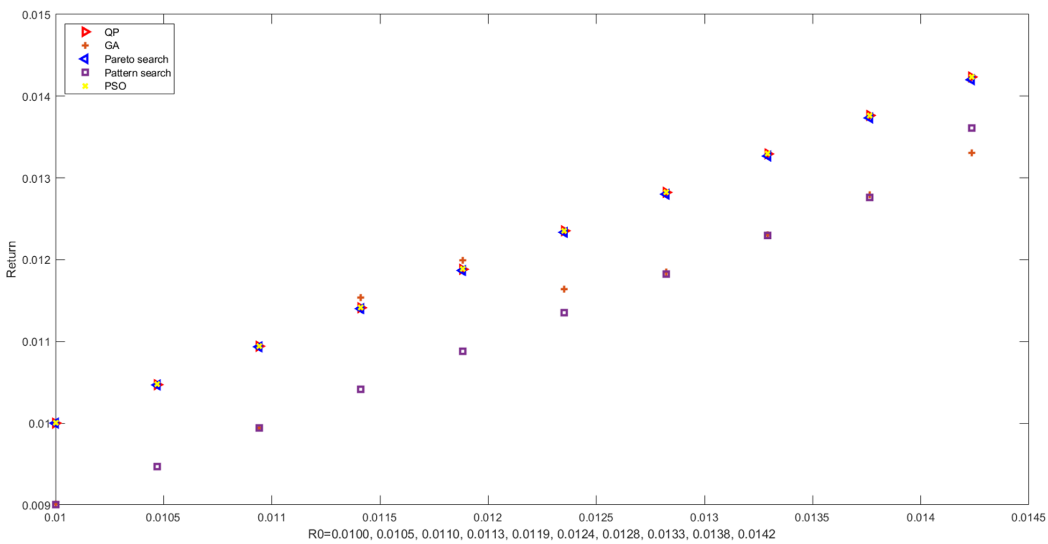
Table 1 shows the results of stock portfolios of the algorithms QP, GA, Pareto-search, and pattern-search based on Markowitz's models (1)-(4) and PSO with models (5)-(7) at the level $R_0 = 0.0138$.

Fig. 1 below shows the results of stock portfolio returns, which are based on the results of the above-mentioned algorithms. In the figure, it can be seen that the results of the stock returns obtained

Table 1. Solutions $w_i, i = 1,\ldots,6$ at the level $R_0 = 0.0138$

QP with model (1)-(4)	GA with model (1)-(4)	Pareto-search with model (1)-(4)	Pattern-search with model (1)-(4)	PSO with model (5)-(7)
.30457E-07	0.033328672	0.001808418	0.031379568	0.000727368
.090577288	0.13388711	0.093706388	0.227230279	0.090634994
.30172E-07	0.019912129	0.000440368	4.88783E-05	0
.54557E-08	0.007529587	5.31501E-08	2.71349E-07	6.82648E-05
.232115455	0.093396319	0.224282883	0.104983821	0.229460012
.677306761	0.711158264	0.679761996	0.637356633	0.679109362

Figure 1. The result of the algorithm solution at R_0 levels



from solving the QP, Pareto, and PSO algorithms are very close, and their results are the same as the expected values. While this is not the case for GA and Pareto-search algorithms, in fact, the efficiency obtained from the resulting solutions for these two algorithms was less than the expected value.

Fig. 2 shows the efficient frontier of all algorithms based on the variance of the solutions separately, and Fig. 3 shows the efficient frontier of all the algorithms based on the variance of the solutions and also the ”estimatePortRisk” in the Matlab toolbox altogether.

4.1 The Results of Hypothesis Testing and the Presentation of Findings

For this purpose, the average results of all the solutions of the algorithms at the levels (14) intended for testing.

The Kruskal-Wallis test, or Kruskal-Wallis H-test (named after two scientists, William Kruskal and Alan Wallis), is a non-parametric test that we have used in this research to examine several sample groups with different numbers of samples. This test is a generalized form of the Yeoman-Whitney test, which only has the ability to compare two sample groups. The parametric mode of this test is a one-way analysis of variance.

Figure 2. The result of the algorithm solution at R_0 points

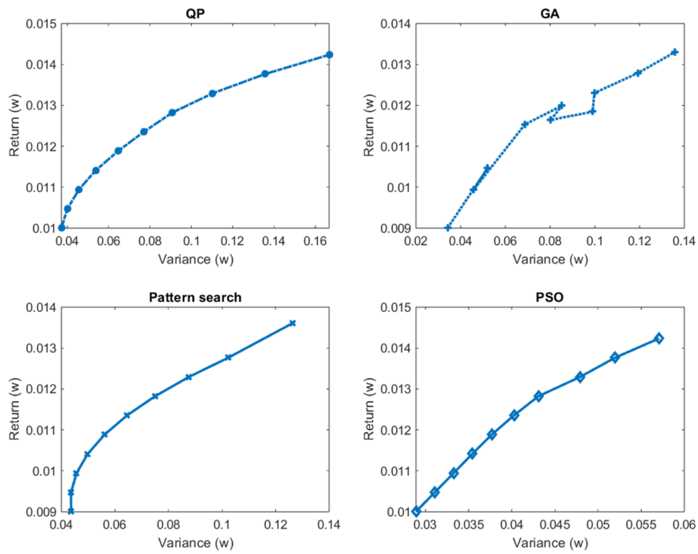
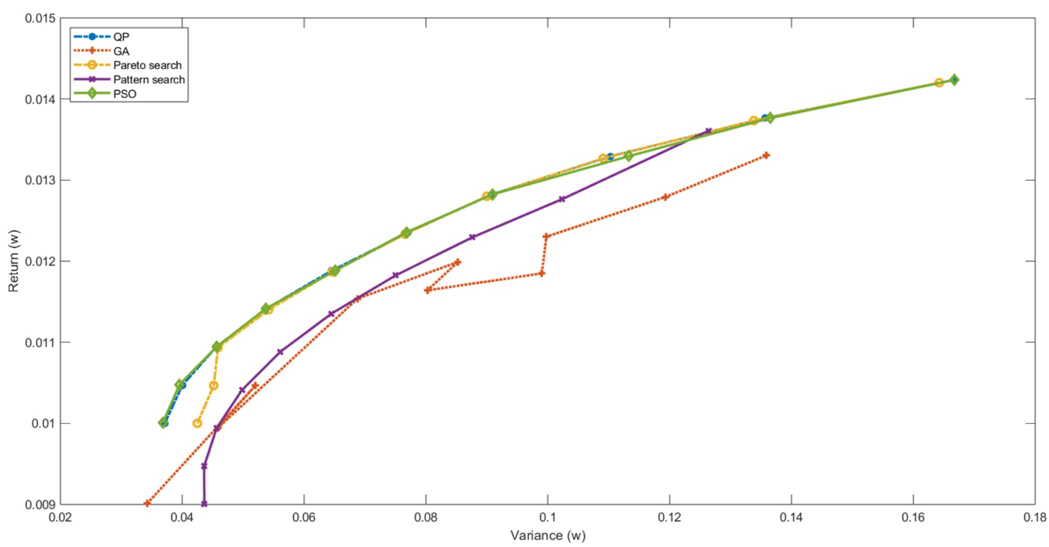


Figure 3. The results of the return of algorithm solution and variance (w_i)



$$\begin{cases} H_0\text{: There is no significant difference between the returns in} \\ \text{the selected and optimal stock portfolios determined by the two algorithms.} \\ H_1\text{: There is a significant difference between the returns in} \\ \text{selected and optimal stock portfolios determined by two algorithms.} \end{cases}$$

In the following tables, we will compare each of the algorithms with the QP algorithm and finally decide whether to reject or accept the null hypothesis.

Table 2. Mean of solutions $w_i, i = 1,...,6$ at the levels (14)

QP	GA	Pareto-search	Pattern-search	PSO
.30457E-07	0.012631181	0.001808418	0.031379568	0.004449216
.090577288	0.113605157	0.093706388	0.227230279	0.076969685
.30172E-07	0.016770969	0.000440368	4.88783E-05	0.001428935
.54557E-08	0.017909771	5.31501E-08	2.71349E-07	6.78529E-06
.232115455	0.120735336	0.224282883	0.104983821	0.237857369
.677306761	0.717726287	0.679761996	0.637356633	0.679288009

Table 3. Comparison of QP and GA

source	SS	degrees of freedom	mean squared	χ^2	Probability $> \chi^2$
Columns	8.333	1	8.333	0.64	0.4233
Error	134.667	10	13.4667		
total	143	11			

Table 4. Comparison of QP and Pareto-search

source	SS	degrees of freedom	mean squared	χ^2	Probability $> \chi^2$
columns	1.333	1	1.333	0.1	0.7488
error	141.667	10	14.1667		
total	143	11			

In the following table, we will compare all the algorithms together.

Finally, according to the values obtained from tables 4-6 and the critical value, which has significant values, it can be decided to reject the one hypothesis and accept the null hypothesis for

Figure 4. The results of the return of algorithm solution and estimatePortRisk(w_i)

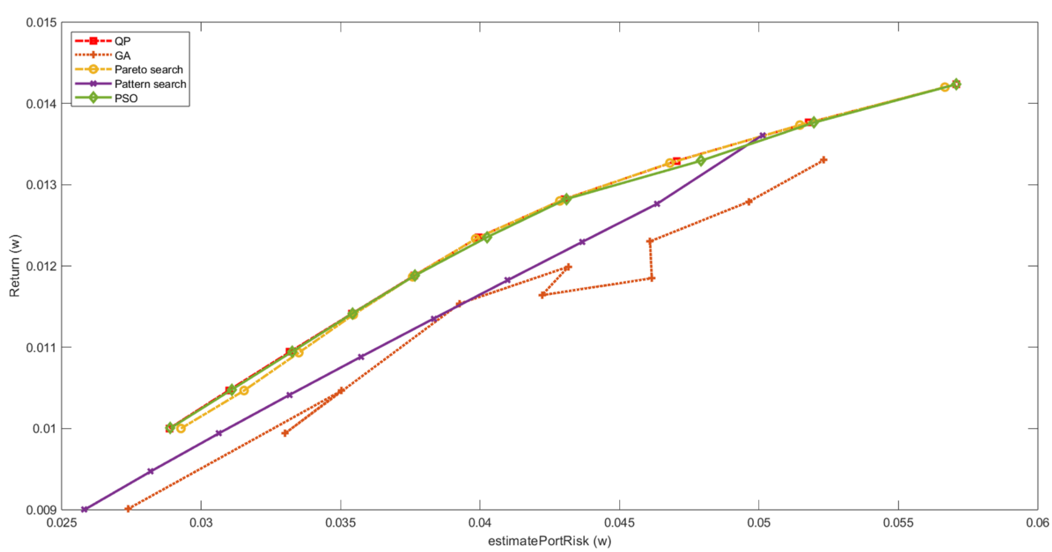


Table 5. Comparison of QP and pattern-search

source	SS	degrees of freedom	mean squared	χ^2	Probability > χ^2
Column	3	1	3	0.23	0.631
Error	140	10	14		
total	143	11			

Table 6. Comparison of QP and PSO

source	SS	degrees of freedom	mean squared	χ^2	Probability > χ^2
Columns	1.333	1	1.333	0.1	0.7488
Error	141.667	10	14.1667		
Total	143	11			

Table 7. Comparison of all algorithm together

source	SS	degrees of freedom	mean squared	χ^2	Probability > χ^2
Columns	52	4	13	0.67	0.9546
Error	2195.5	25	87.82		
Total	2247.5	29			

a significant difference between the returns of Pareto-search, pattern-search, and PSO algorithms in comparison to QP, and from table 3, it can be decided to accept the one hypothesis and reject the null hypothesis for a significant difference between the returns between GA and QP. In table 7, the comparison of all algorithms together shows that there is no significant difference between the returns in the selected and optimal stock portfolios determined by all algorithms together.

5. CONCLUSION AND DISCUSSION

Markowitz's model showed that the most important factors in choosing the optimal stock portfolio are the two factors of return and risk. The results of this research show that the optimization methods based on the defined objective function sought to select the stock portfolio that has the highest return and the lowest risk. As mentioned earlier, the purpose of this research is to investigate Markowitz's model based on the embedded PSO algorithm and four algorithms (QP, GA, Pareto search, and pattern search) in the Matlab toolbox and their comparison. Therefore, six specific stock companies were tested during a specific period of time in order to investigate this issue. In this regard, we first examined the Markowitz mean variance model using the penalty function method and the standard PSO algorithm. Then, the Kruskal- Wallis test at the 1% level was used to compare the two algorithms. In order to test the hypothesis, each algorithm was compared with QP. The statistical tests related to the results show that there is no significant difference between the average of solutions at levels (14) obtained from the PSO, Pareto-search, and pattern-search with QP, while this was not the case with the GA. In comparison, a successful algorithm is one that has a lower amount of risk, or, in other words, has reached the best return with the least risk. As seen in the results, PSO, QP, and Pareto search algorithms perform better. The results indicate that there is no significant difference between the proposed method of the PSO algorithm and the two methods of QP and Pareto-search algorithm, while pattern-search algorithm and GA differed slightly from the three above-mentioned algorithms in comparison.

FUNDING

No funding was used in this study

AVAILABILITY OF DATA AND MATERIALS

The data that supports the findings of this study is openly available at Google Finance and Yahoo Finance.

COMPETING OF INTEREST

The authors declare that they have no competing interests.

PROCESSING DATES

11, 2024

This manuscript was initially received for consideration for the journal on 08/16/2024, revisions were received for the manuscript following the double-anonymized peer review on 10/31/2024, the manuscript was formally accepted on 11/11/2024, and the manuscript was finalized for publication on 11/22/2024

CORRESPONDING AUTHOR

Correspondence should be addressed to Kianoush Fathi Vajargah; fathi_kia10@yahoo.com

REFERENCES

Aranha, C., & Iba, H. (2009). The memetic tree-based genetic algorithm and its application to portfolio optimization. *Memetic Computing*, *1*(2), 139–151. DOI: 10.1007/s12293-009-0010-2

Audet, C., & Dennis, J. E.Jr. (2002). Analysis of generalized pattern searches. *SIAM Journal on Optimization*, *13*(3), 889–903. DOI: 10.1137/S1052623400378742

Bazrkar, M. J., & Hosseini, S. (2023). Predict stock prices using supervised learning algorithms and particle swarm optimization algorithm. *Computational Economics*, *62*(1), 165–186. DOI: 10.1007/s10614-022-10273-3

Chang, T.-J., Yang, S.-C., & Chang, K.-J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications*, *36*(7), 10529–10537. DOI: 10.1016/j.eswa.2009.02.062

Clerc, M. (2013). *Particle Swarm Optimization*. ISTE. Wiley.

Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, *6*(1), 58–73. DOI: 10.1109/4235.985692

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (*Vol. 5*). Springer.

Cornuejols, G., & Tütüncü, R. (2006). *Optimization methods in finance* (*Vol. 5*). Cambridge University Press. DOI: 10.1017/CBO9780511753886

Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis Real World Applications*, *10*(4), 2396–2406. DOI: 10.1016/j.nonrwa.2008.04.023

Custódio, A. L., Madeira, J. A., Vaz, A. I. F., & Vicente, L. N. (2011). Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, *21*(3), 1109–1140. DOI: 10.1137/10079731X

Derrac, J., Garca, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3–18. DOI: 10.1016/j.swevo.2011.02.002

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, pages 39–43. Ieee. DOI: 10.1109/MHS.1995.494215

Erwin, K., & Engelbrecht, A. (2023). Multi-guide set-based particle swarm optimization for multi-objective portfolio optimization. *Algorithms*, *16*(2), 62. DOI: 10.3390/a16020062

Gilli, M., Maringer, D., & Schumann, E. (2019). *Numerical methods and optimization in finance*. Academic Press.

Goldberg, D. (1989). 'geneticatgorithms in search, optimization& machine [learning," addison- wes] ey.

Gunjan, A., & Bhattacharyya, S. (2022). A brief review of portfolio optimization techniques. *Artificial Intelligence Review*, 1–40.

He, G., & Huang, N. (2014). A new particle swarm optimization algorithm with an application. *Applied Mathematics and Computation*, *232*, 521–528. DOI: 10.1016/j.amc.2014.01.028

Kalayci, C. B., Ertenlice, O., & Akbay, M. A. (2019). A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Systems with Applications*, *125*, 345–368. DOI: 10.1016/j.eswa.2019.02.011

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, *volume 4*, pages 1942–1948. IEEE. DOI: 10.1109/ICNN.1995.488968

Kuo, R., & Chiu, T.-H. (2024). Hybrid of jellyfish and particle swarm optimization algorithm-based support vector machine for stock market trend prediction. *Applied Soft Computing*, *154*, 111394. DOI: 10.1016/j.asoc.2024.111394

Lin, C.-C., & Liu, Y.-T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research*, *185*(1), 393–404. DOI: 10.1016/j.ejor.2006.12.024

Liu, X., & Li, A.-D. (2024). An improved probability-based discrete particle swarm optimization algorithm for solving the product portfolio planning problem. *Soft Computing*, *28*(3), 2535–2562. DOI: 10.1007/s00500-023-08530-0

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77–91.

Mercangöz, B. A. (2021). Portfolio optimization. In *Applying Particle Swarm Optimization: New Solutions and Cases for Optimized Portfolios*, pages 15–27. Springer. DOI: 10.1007/978-3-030-70281-6_2

Nesterov, Y. (2018). Lectures on convex optimization, volume 137 of *Springer Optimization and Its Applications*. Springer, Cham. Second edition of [MR2142598]. DOI: 10.1007/978-3-319-91578-4

Nocedal, J., & Wright, S. J. (2006). Quadratic programming. *Numerical optimization*, pages 448–492.

Ponsich, A., Jaimes, A. L., & Coello, C. A. C. (2012). A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation*, *17*(3), 321–344. DOI: 10.1109/TEVC.2012.2196800

Song, Y., Liu, Y., Chen, H., & Deng, W. (2023). A multi-strategy adaptive particle swarm optimization algorithm for solving optimization problem. *Electronics (Basel)*, *12*(3), 491. DOI: 10.3390/electronics12030491

Thakkar, A., & Chaudhari, K. (2021). A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization. *Archives of Computational Methods in Engineering*, *28*(4), 2133–2164. DOI: 10.1007/s11831-020-09448-8

Wright, J. N. S. J. (2006). Numerical optimization.

Zanjirdar, M. (2020). Overview of portfolio optimization models. *Advances in mathematical finance and applications*, 5(4):419–435.