North American Journal of Economics and Finance 82 (2026) 102568

Contents lists available at ScienceDirect

North American Journal of Economics and Finance

journal homepage: www.elsevier.com/locate/najef

Constrained portfolio optimization via Artificial Gorilla Troops: Benchmarking against swarm-intelligence metaheuristic algorithms

Vasileios Gkonis [a,*], Ioannis Tsakalos [a], Ilias Kampouris [b]

[a] Department of Accounting and Finance, University of Thessaly, Gaiopolis Campus, Larissa GR-41500, Greece
[b] College of Business Administration, Abu Dhabi University, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Keywords:
Artificial Gorilla Troop Optimizer
Constrained optimization
Metaheuristic algorithms
Portfolio optimization
Sharpe ratio

ABSTRACT

Over the years, the challenge of portfolio optimization has gained increasing attention from scientists and experienced investors, where maximizing returns with the least possible risk is the major goal. In recent times, there has been a significant surge in interest in metaheuristic algorithms across various industries. In this study, we propose the use of the Artificial Gorilla Troops Optimizer (AGTO) for portfolio optimization. We evaluate its effectiveness and compare it against sixteen other swarm intelligence-based metaheuristic algorithms under varying population and epoch sizes. Our evaluation is based on the Sharpe ratio, utilizing a portfolio composed of stocks from the Dow Jones Industrial Average. The results indicate that AGTO demonstrates strong potential as an effective method for optimal portfolio selection. In addition, this work provides valuable insights into metaheuristic algorithms that have seen relatively limited application in the existing portfolio optimization literature.

1. Introduction

Portfolio optimization is a vital topic in finance that has been extensively studied throughout the years. Markowitz (1952) devised one of the first techniques for portfolio optimization in the 1950s. Modern portfolio theory, his analytical framework, notably provided a quantitative approach for determining the optimal allocation of assets in a portfolio. Markowitz's work established the groundwork for portfolio optimization and revolutionized investment decision-making by emphasizing the importance of balancing risks and returns. The challenge of portfolio optimization is to define the optimal asset allocation that maximizes the expected return given a specific level of risk or minimizes the risk given a certain level of expected return. It entails selecting a combination of assets to invest in while considering their predicted returns, volatility, correlation, and other risk variables. Finance makes considerable use of optimization techniques for portfolio selection and risk management. Many mathematical models and algorithms, including mean–variance optimization, conditional value-at-risk, and stochastic programming, have been developed to address this issue.

Wolpert & Macready (1997) initially introduced the "No Free Lunch" (NFL) theorem, which questions the idea that a single optimization method can be applied to all problems and produce optimal solutions. Specifically, it asserts that the success of an optimization process depends on the context and the specific problem being addressed. Over the past two decades, the research community has observed an explosion of interest in metaheuristic algorithms. These algorithms can locate optimal solutions to

* Corresponding author.
E-mail addresses: vgkonis@uth.gr (V. Gkonis), i.tsakalos@uth.gr (I. Tsakalos), ilias.kampouris@adu.ac.ae (I. Kampouris).

https://doi.org/10.1016/j.najef.2025.102568
Received 6 June 2025; Received in revised form 8 November 2025; Accepted 13 December 2025
Available online 15 December 2025
1062-9408/

optimization issues when conventional optimization techniques fail or are overly time-consuming. Their use has been applied to a variety of fields, including engineering, computer science, logistics, and finance, and used for portfolio construction.

Metaheuristic algorithms mimic various phenomena, including animals' behaviour, laws of mathematics and physics, and even human behaviour. According to Mirjalili et al. (2014), there are four key reasons for the increasing development of these types of algorithms. First, these algorithms rely on simple notions, allowing practitioners to replicate numerous natural concepts and suggest new algorithms, as well as alter and enhance existing ones. Secondly, metaheuristics can be applied to various problems without modifying their structure. The researcher merely needs to understand the problem's inputs and evaluate the output. Most of these algorithms use derivation-free processes, which begin with a random value and eliminate the requirement for derivative information. This stochastic tendency can be advantageous over gradient-based optimization methods in situations when the derivative is computationally costly or even unknown. Finally, their stochastic nature prevents stagnation and encourages thorough exploration of the solution space, which is crucial for solving real-world problems that are complex and contain a high number of local optima. Over the years, researchers have published more than 500 metaheuristic algorithms. Most of these algorithms have been deployed in the last decade (Rajwar et al., 2023).

Aside from the algorithm-specific parameters defined by its core architecture, two critical factors influencing the performance of population-based algorithms, are the population size and the number of iterations (epochs). Several studies have evaluated the performance individual algorithms under different configurations (Shi & Eberhart, 1999; Piotrowski, 2017; Piotrowski et al., 2020), whereas others compare multiple ones. Dhal et al. (2021) investigated the impact of population size on the performance of three swarm-based algorithms. Their study evaluated algorithm behaviour across various function types and dimensions, highlighting that population size significantly affects optimization outcomes. The authors emphasize that the optimal population size depends heavily on the algorithm, problem type, and dimensionality.

Agushaka et al. (2023) examined the performance of eleven metaheuristic algorithms across varying population sizes and number of iterations counts on benchmark datasets, concluding that while no algorithm is universally optimal, maintaining sufficient population diversity and iteration depth is key to achieving robust results. In the same vein, Si-Ma et al. (2025) conducted a large-scale evaluation of 123 swarm intelligence algorithms across different problem types, showing that efficient performance for simple, low-dimensional problems is typically achieved within 30 to 80 iterations, whereas more complex problems benefit from higher iteration counts.

Despite these valuable contributions, we believe that a relative gap remains in the literature concerning the systematic assessment of how population and iteration size affect the performance of metaheuristic algorithms, particularly within the practical and widely studied domain of portfolio optimization. Moreover, over the last years the Artificial Gorilla Troops Optimizer (AGTO) (Abdollahzadeh et al., 2021) has shown exceptional performance in a variety of problems, and has also pushed researchers to further evaluate its performance as well as proposed modified approaches (Hussien et al., 2024). To the best of our knowledge there is limited applications of AGTO in financial applications. In this work, we attempt to address the concept of portfolio optimization with the utilization of the AGTO algorithm by optimizing the Sharpe ratio (Sharpe, 1964), which is a popular metric among practitioners and aids in evaluating their investing strategy. We compare AGTO's performance with sixteen other swarm intelligence-based metaheuristic algorithms, applying them to a portfolio composed of stocks from the Dow Jones Industrial Average (DJIA) index. As far as we know, this is the first study to apply AGTO for portfolio optimization, and several of the algorithms included in our comparison have either not been previously used in this context or have only been applied to a limited extent. The results indicate that AGTO performs strongly across a range of population and iteration settings, showing relatively low sensitivity to these parameters while still producing competitive results.

The remainder of this study is organized as follows. Section 2 reviews the existing literature on portfolio optimization using metaheuristic algorithms. Section 3 introduces the swarm intelligence-based metaheuristic algorithms and outlines the formulation of the optimization problem addressed in this work. Section 4 details the experimental set-up and presents the corresponding results. Finally, Section 5 concludes the study and suggests potential directions for future research.

2. Literature review

Dueck and Winker (1992) presented the Threshold Accepting (TA) algorithm, which can optimize portfolios subject to nearly arbitrary constraints and virtually any utility function. The authors claim that, by employing technical analysis, an investor might enhance their portfolio within a realistic model. They conclude that risk is not a standard unit of measurement; hence, portfolios that perform well within one paradigm may underperform in another. Testing TA across numerous scenarios for portfolios comprising bonds from the Federal Republic of Germany, the results revealed the absence of a universal risk measure and underscored the necessity for diverse optimizers for different risk functions. We could claim that they unofficially formulated an NFL theorem for portfolio optimization.

One year later, Arnone et al. (1993) reapproached the concept of portfolio optimization utilizing Genetic Algorithms (GA) (Holland, 1992), stating that stochastic methods could be advantageous when deterministic optimization algorithms underperform, and that GA is capable of exploring extensive search spaces. Mozafari et al. (2011) attempted to address a realistic optimization problem with floor, ceiling, and cardinality constraints. The author introduced the hybrid Improved Particle Swarm Optimization – Simulated Annealing (IPSO-SA) model, which integrates an improved Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995) with a modified Simulated Annealing (SA) (Kirkpatrick et al., 1983) algorithm. IPSO-SA surpassed GA, Tabu Search (TS) (Glover, 1989), SA, and PSO in performance across five benchmark data sets. Mishra et al. (2014) introduced the Non-dominated Sorting Multi-Objective Particle Swarm Optimization (NS MOPSO) to tackle a realistic portfolio asset selection with budget floor and ceiling constraints and cardinality as a multi objective optimization problem. The researchers evaluated the model's performance alongside GA, TS, SA, and PSO over six distinct market indexes, revealing that the proposed method is a viable solution for the constrained portfolio asset selection problem.

Bacanin & Tuba (2014) introduced a modified Firefly Algorithm (FA) (Łukasik & Żak, 2009) for a cardinality constrained mean portfolio optimization problem with entropy constraints. The authors combined the Firefly algorithm (FA) with the Artificial Bee Colony (ABC) optimization algorithm (Karaboga, 2005) to take advantage of ABC's "limit" parameter, enhancing FA's exploration ability. Their methodology surpassed other traditional metaheuristic algorithms, including Genetic Algorithm GA, TS, SA, and PSO. Chen (2015) introduced a Modified Artificial Bee Colony (MABC) algorithm for a possibilistic semi-absolute deviation portfolio, incorporating transaction costs, cardinality, and quantity restrictions. Their numerical trials shown that MABC surpassed the conventional ABC as well as GA, TS, SA, PSO, and Local Search (LS).

Ruiz-Torrubiano & Suárez (2015) developed a memetic technique that integrates genetic algorithms and quadratic programming to address the issue of optimum portfolio selection with cardinality constraints and piecewise linear transaction costs. Their approach works by handling the continuous and combinatorial parts of optimization independently. The modification of the Random Assorting Recombination (RAR) crossover operator (Radcliffe,1993) to alter additional attributes in the genetic representation that indicate the direction of the transactions in each asset during rebalancing, is one of the primary achievements of their research. The progeny produced are always feasible thanks to the extended RAR (eRAR) cross-over, which can manage transaction fees, cardinality restrictions, and minimum trade size restrictions.

Kalayci et al. (2020) introduced a hybrid metaheuristic algorithm for addressing cardinality constrained portfolio optimization, which integrates continuous Ant Colony Optimization (ACO) (Dorigo et al., 2007), GA, and ABC algorithms. The proposed approach incorporates the elitism mechanism from GA, the modification rate from ABC, and the Gaussian formulation from ACO. The findings indicated that the elitism mechanism and modification parameter significantly influenced the convergence performance of the suggested method. Furthermore, a comparison of the computational outcomes on seven publicly accessible benchmark problems with those of alternative approaches in the literature illustrates that the proposed algorithm is very efficient and competitive with state-of-the-art algorithms.

Chen et al. (2021) employed traditional particle swarm optimization and Particle Swarm with Center of Mass Optimization (PSOCM) (Jamous et al., 2015) to assess a diverse sample of S&P 500 companies, contrasting it with the Sequential Least Squares Programming (SLSQP) technique for portfolio selection. Furthermore, the authors enhanced both the Sharpe and Sortino ratios, noting that in some scenarios, PSO and PSOCM outperformed SLSQP in terms of speed and yielded superior outcomes, particularly for the Sortino ratio. Jin et al. (2019) examined a multi-period, tri-objective portfolio optimization problem wherein asset returns are represented as uncertain variables. Utilizing uncertainty theory, they introduced a portfolio model that concurrently incorporates loss-averse utility, liquidity risk, and diversification. A chance constraint was integrated to address investors' safety requirements over the investment horizon. The authors implemented a Self-Adaptive Particle Swarm Optimization (SAPSO) algorithm to address the problem, incorporating a self-adaptive stochastic ranking mechanism aimed at optimizing the balance between exploration and exploitation in the search process.

Mishra et al. (2021) suggested the LASSO – TLBO – SVR, a hybrid model combining Least Absolute Shrinkage and Selection Operator (LASSO), the Teaching Learning-Based Optimization algorithm (TLBO), and Support Vector Regression (SVR) model for optimal portfolio selection. Initially, the LASSO method is employed for feature selection, followed by TLBO algorithm to optimize the hyperparameters of the SVR, ultimately selecting the optimum stock selections. To optimize the stock's weight, the authors utilize the Generalized Reduced Gradient (GRG) Nonlinear technique, GA, and Equal Weight technique. Of the three methodologies, the GA yielded superior outcomes.

The Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014) is a popular metaheuristic algorithm extensively employed in various problems, including portfolio optimization. Ahmad & Shahid (2022) addressed the portfolio selection question by utilizing GWO on datasets from five indices, demonstrating superior performance compared to other metaheuristic algorithms, including GA, TS, and SA. Liu et al. (2021) combined an Enhanced GWO (EGWO) with Support Vector Regression (SVR) and exhibited that their methodology consistently yields excess returns in the American and Chinese markets. Additionally, they contrasted EGWO with four intelligent optimization algorithms: GA, PSO, Gravitational Search Algorithm (GSA) (Rashedi et al., 2009), and Harmony Search (HS) (Geem et al., 2001). The suggested EGWO-SVR attained superior returns in the Chinese A-share market.

The beetle Antennae Search Algorithm (BAS) is inspired by longhorn beetles (Jiang & Li, 2017). The BAS algorithm emulates the functioning of antennae and the random walking of beetles in their natural habitat. BAS has been systematically evaluated in academic literature about portfolio optimization. The Non-linear Activated Beetle Antennae Search (NABAS) introduced by Khan et al. (2022) addresses the non-convex tax-aware portfolio optimization challenge. Empirical investigation using NASDAQ stock data highlighted its superior performance relative to established algorithms like BAS, PSO, and GA. Numerous BAS variants have been explored in the literature, especially for portfolio optimization. (Medvedeva et al., 2021; Mourtas & Katsikis, 2022).

Metaheuristic algorithms are extensively employed in finance and portfolio optimization (Di Tollo & Roli, 2008; Grishina et al., 2017; Saiz et al., 2022; Erwin & Engelbrecht, 2023), with numerous additional significant studies available. Nevertheless, a substantial body of literature is available to provide a comprehensive and insightful evaluation of the existing research. Doering et al. (2019) investigated the status of metaheuristic algorithms in portfolio optimization and risk management, noting a rise in publications in this domain in recent years. A significant conclusion is the dominance of population-based algorithms and the tendency towards the development of hybrid algorithms. The authors ultimately concur with the NFL theorem, as no singularly successful algorithm exists.

Although extensive research has been conducted in portfolio optimization using metaheuristic algorithms, we believe that a gap exists in the literature regarding comprehensive comparisons between well-established and recently introduced metaheuristic algorithms. Most studies tend to benchmark proposed approaches against a limited set of classical metaheuristic algorithms, frequently neglecting comprehensive evaluation involving a wider range of metaheuristic algorithms. This work aims to address that gap by assessing the performance of the AGTO in comparison with both classical and newer metaheuristic algorithms. To ensure a balanced and fair comparison, the evaluation examines performance across several population sizes and numbers of epochs, where also this comparison is usually omitted. Given AGTO's promising performance in recent studies, we consider it a strong candidate for the challenge of portfolio optimization.

Despite the substantial body of work in portfolio optimization using metaheuristic algorithms, existing studies often benchmark against a limited set of methods and rarely explore how population size and epoch configurations influence performance across a wide spectrum of algorithms. Furthermore, emerging metaheuristics such as AGTO, Tuna Swarm Optimization algorithm (TSO), and the Artificial Rabbit Optimizer (ARO) remain underexplored in financial contexts. To address these limitations and guide our investigation, the following hypotheses are proposed:

- **H1**: The Artificial Gorilla Troops Optimizer (AGTO) yields superior performance in constrained portfolio optimization, as measured by the Sharpe ratio, compared to other swarm intelligence-based metaheuristics.
- **H2**: AGTO maintains robust performance across diverse configurations of population size and number of epochs.
- **H3**: In addition to achieving high maximum Sharpe ratios, AGTO exhibits lower performance variance and higher average stability than competing algorithms.
- **H4**: The observed differences in performance between AGTO and other algorithms are statistically significant, as determined by non-parametric testing.

These hypotheses form the foundation for the empirical experiments that follow, wherein we evaluate AGTO and sixteen alternative metaheuristics under multiple constraint settings and data-driven scenarios.

3. Methodology and problem formulation

In this section, we begin with a brief introduction to the benchmark algorithms considered in this study. We then offer a detailed overview of AGTO and conclude with the formulation of the problem addressed in this study.

3.1. Swarm intelligence algorithms

The social structure and behaviour of insects and animals in nature serve as inspiration for swarm intelligence algorithms. Kennedy & Eberhart (1995) introduced PSO, a well-known algorithm, drawing inspiration from the swarm behaviour and dynamic movements of birds and fish. Other established and popular swarm-based algorithms include the ABC, which models the cooperative behaviour of three types of bees: employed, observer, and scout, as they search for food. The Grasshopper Optimizer Algorithm (GOA) (Saremi et al., 2017) simulates grasshoppers' swarming behaviour in nature, both finding food and avoiding predators. The Social Spider Algorithm (SSA) (James & Li, 2015) draws inspiration from the social behaviour of spiders, particularly their cooperative strategies in colony life. Spiders, particularly those in social species, exhibit complex behaviours such as communication through web vibrations, collaborative hunting, and resource sharing. The Antlion Optimization algorithm (ALO) (Mirjalili, 2015) replicates antlions' hunting strategy, which involves hunting and trapping ants in five distinct steps.

The GWO is inspired by the social structure and cooperative hunting behaviour of grey wolves (*Canis lupus*), in which the alpha, beta, delta, and omega ranks work together to capture prey. The Coyote Optimization Algorithm (COA) (Pierezan & Coelho,2018) is based on the social structure of coyotes in nature (Canis latrans). COA, it simulates the sharing of information and adaptation of coyotes in nature, rather than focusing solely on hunting behaviour like GWO. On the contrary, ARO (Wang et al., 2022) mimics rabbits' survival strategies in nature, simulating the detour foraging strategy and random hiding.

The Aquila Optimizer (Abualigah et al., 2021) replicates the natural behaviour of aquilas by employing four distinct strategies to capture their prey. Another widely recognized algorithm is Cuckoo Search (CS) (Yang & Deb, 2009). The algorithm draws inspiration from the fact that certain cuckoo species exhibit brood parasitism. The Harris Hawks Optimization (HHO) (Heidari et al., 2019) algorithm mimics Harris Hawks' grouped attacking behaviour, which employs the "surprise pounce" strategy to surprise their prey while hunting.

Humpback whales' bubble-net hunting strategy served as the inspiration for the Whale Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016). Similarly, spiral and parabolic foraging describe the hunting behaviour of tunas, which the TSO (Xie et al., 2021) mimics. The Sailfish Optimization Algorithm (SOA) (Shadravan et al., 2019) draws inspiration from sailfish, the fastest fish in the ocean, and emulates their cooperative hunting behaviour. Another popular algorithm is Marine Predator Algorithms (MPA) (Faramarzi et al., 2020), a metaheuristic optimization algorithm inspired by the hunting strategies of marine predators, particularly those at the top of the oceanic food chain, such as sharks, orcas, and other large predators.

3.2. Artificial Gorilla Troops

This section provides a comprehensive overview of AGTO, as it is the suggested metaheuristic algorithm for portfolio optimization. As extremely intelligent primates, gorillas display complex social structures, emotional capacities, and cognitive skills, such as tool-making and a sense of the past and future.

Gorillas live in complex social structures led by a dominant silverback male, who guides troop movements, resolves conflicts, protects the group, and maintains social order. Blackbacks, which are younger male gorillas, provide additional protection for the troop and can assume leadership if the silverback dies. Female gorillas form strong bonds with silverbacks for protection and reproduction, while males compete intensely for dominance, though all-male groups can form peaceful alliances. AGTO mimics these dynamics by structuring its search process around exploration and exploitation phases. During exploration, three operators enhance the algorithm's ability to search diverse areas of the optimization space. In the exploitation phase, two mechanisms intensify the search around local optima.

The algorithm simulates gorilla troop behaviour using three types of solutions: X (current position), GX (candidate positions), and the silverback (best solution), with only one silverback per population representing the optimal solution. If a GX outperforms its corresponding X, it replaces it; otherwise, it's stored in memory. Inspired by how gorillas seek better food sources and group positions, weaker solutions are moved away from while better ones are approached, allowing AGTO to effectively balance exploration and exploitation and adaptively navigate complex search spaces.

3.2.1. Exploration phase

Gorillas live in groups under the leadership of a silverback, whose commands are obeyed. However, gorillas may leave their group and migrate to different locations, known or unknown. In the AGTO algorithm, gorillas are considered candidate solutions, with the best candidate solution at each optimization operation stage representing the silverback.

Three mechanisms are used in the exploration phase: *migration to an unknown location*, *migration toward a known location*, and *moving toward other gorillas*. These mechanisms are selected based on a general procedure involving a parameter named $p \in [0,1]$ which must set before optimization procedure and a random number $rand\ U(0,1)$. If r and <p the *migration to an unknown location* is selected. If r $\geq$ 0.5, the mechanism of *movement toward other gorillas* is selected, and if r $<$ 0.5, migration to a known location is selected. Each mechanism provides specific benefits to the AGTO algorithm's performance. The first mechanism enables the algorithm to extensively explore the problem space, the second improves AGTO's exploration performance, and the third strengthens AGTO's ability to escape local optimal points (Eq. (1)).

$$GX(t+1)\begin{cases} (UB-LB)\times r_1 + LBrand < p \\ (r_2 - C)\times X_r(t) + L\times Hrand \geq 0.5 \\ X(i) - L\times (L\times X(t) - GX_r(t)\,) + r_3\times (X(t) - GX_r(t))), rand < 0.5 \end{cases} \tag{1}$$

where:

$$r_1, r_2, r_3\ U(0,1)$$

$$LB\ : Lower\, bound$$

$$UB : Upper\, bound$$

In Eq. (1) $GX(t+1)$ denotes the gorilla candidate position in the next iteration and. GX_r is one randomly selected gorilla from the group. C (Eq. (2)) takes into account the current iteration (*It*), the overall value of iterations (*MaxIt*) to execute the process of optimization, and variable F (Eq. (3)).

$$C = F\times\left(1-\frac{It}{MaxIt}\right) \tag{2}$$

$$F = cos(2\times r_4) + 1, r_4 \in [0,1] \tag{3}$$

As the authors have visually shown in Eq. (2), early phases of the optimization process gives values with abrupt changes across a wide time frame; however, this interval of change gets smaller in the last stages.

Next, the L parameter (Eq. (4)) is calculated, which simulates the silverback's leadership. Gorillas lack adequate experience in the early phases of group leadership which may lead them to take poor decisions. However, silverbacks attain enough experience over the "iterations", which results in a strong and stable leadership.

$$L = C\times l, l\ U(-1,1) \tag{4}$$

Finally, H and Z, which is a random variable, are calculated using Eq. (5) and Eq. (6) respectively

$$H = Z\times X(t) \tag{5}$$

$$Z = [-C, C] \tag{6}$$

3.2.2. Exploitation phase

The AGTO algorithm's exploitation phase involves two mechanisms. The selection of the two phases is determined by the values of C, and W parameter which should be set before the optimization process. The silverback gorilla leads the group, makes decisions, directs the gorillas toward food sources, and is responsible for their safety and well-being. All gorillas in the group follow the silverback's decisions. If $C \geq W$ then the *Follow the Silverback* (Eq. (7)) mechanism is selected:

$$GX(t+1) = L \times M + (X(t) - X_{\text{silverback}}) + X(t) \tag{7}$$

$$M = \left(|\frac{1}{N} \sum_{i=1}^{N} GX_i(t)|^g \right)^{\frac{1}{g}} \tag{8}$$

$$g = 2^L \tag{9}$$

However, the silverback may weaken, get aged which will ultimately leading to death, other blackback male gorillas may become the group leader. Moreover, even if silverback is healthy, the blackbacks may exhibit signs of rebelization which leads to engage with the silverback and seek to the domination of the group. In such scenarios, the *competition for adult females* (Eq. (10)) mechanism is selected. The pseudocode of the algorithm, as outlined by Abdollahzadeh et al. (2021), is provided in the Appendix A.

$$GX(i) = X_{\text{silverback}} - (X_{\text{silverback}} \times Q - X(t) \times Q) \times A \tag{10}$$

$$Q = 2 \times r_5 - 1 \tag{11}$$

$$A = \beta \times \mathrm{E} \tag{12}$$

$$\mathrm{E} = \begin{cases} N_1, rand \geq 0.5 \\ N_2, rand \geq 0.5 \end{cases} \tag{13}$$

3.3. Constraint satisfaction and portfolio selection

3.3.1. Portfolio constraints

According to the investor's preferences and investment strategy, a number of portfolio selection constraints come into play. The budget constraint is the most fundamental, ensuring that the total weights of the portfolio's assets add to one. Risky portfolios can be categorized as restricted or unrestricted based on the presence or absence of certain constraints. Restricted portfolios prohibit short selling, resulting in non-negative asset weights, while unrestricted portfolios allow short selling. Sector constraints may also be considered, particularly when the investor aims to create a diverse portfolio. In this case, it is advisable to use weights as safeguards to enhance diversification and mitigate sector risk.

Moreover, cardinality constraints are significant, as investors favor portfolio simplification by restricting the number of assets and reducing transaction costs. Such constraints transform the portfolio optimization problem into a mixed integer programming problem, so complicating the computational process of identifying the optimal solution. In alignment with portfolio diversity and simplification, an investor may consider incorporating floor and ceiling limits. These types of constraints could be considered as bounds during the optimization process. Maintaining a weight below 5 % for several assets may be prohibitively expensive and impossible, while individual assets exceeding 50 % would lead to excessive concentration and higher risk.

3.3.2. Restricted short-selling portfolio optimization

Introduced by William Sharpe in 1966, the Sharpe ratio (Eq. (14)) remains a prevalent measure among investors to evaluate the performance of assets in relation to their risk. It computes the excess return of a portfolio over the risk-free rate per unit of risk, as determined by the standard deviation of the portfolio.

$$SharpeRatio = \frac{R_p - r_f}{\sigma_p} \tag{14}$$

where R_p is the expected portfolio return, r_f is the risk-free rate, and σ_p is the standard deviation of the portfolio's returns.

An investor seeks to maximize their investment for a given level of risk according to the Sharpe ratio (Eq. (15)). In this work, we consider the case of restricted portfolio with budget constraint (Eq. (16)), which ensures that the total sum of the portfolio's asset weights equals one, representing a full allocation of available capital. Additionally, the portfolio is restricted from short selling (Dupačová & Kopa, 2014; Pagnoncelli et al., 2017; Jin et al., 2019; Mishra et al., 2021), meaning that the lower bound of asset weights cannot be negative (Eq. (17)). From an optimization perspective, it can be formulated as follows:

$$\text{Maximize}: \frac{R_p - r_f}{\sigma_p}, \tag{15}$$

Subject to:

$$\sum_{i=1}^{N} w_i = 1, \tag{16}$$

$$0 \leq w_i \leq 1, \tag{17}$$

where N is the total number of assets, and w_i corresponds to the weight of i_{th} asset in the portfolio. One of the necessary restrictions for the specific portfolio optimization problem is that the sum of the weights must be equal to one. To achieve this, we use the normalization approach (Eq. (18) of Pai & Michel (2014), where each respective weight W_i in normalized as follows:

$$W_i^{(norm)} = \frac{(W_i - W_{min})}{\sum_{i=1}^{N}(W_i - W_{min})} \tag{18}$$

4. Experimental analysis

4.1. Dataset and evaluated algorithms

Experimental evaluation was conducted using historical DJIA stock data from Yahoo Finance, dated between January 1, 2023, and August 23, 2024 (Table 1). Price series were aligned on common U.S. trading days, and daily log returns were computed. For the respective stocks we optimize the Sharpe ratio which we report it on annual basis, and the risk-free rate is set to 5 %. In total, seventeen metaheuristic algorithms were employed for optimization, which are presented in Section 3 (Table 2).

The respective metaheuristic algorithms have been chosen to be executed for a combination of three epochs (30, 50, 100) and four population sizes (10, 30, 50, 100). All algorithms were implemented using the MEALPY package (Van Thieu & Mirjalili, 2023). Furthermore, this study follows the repeated execution of each algorithm for a total of 30 iterations as with other works (Mirjalili et al., 2014, Mirjalili,2 015, Mirjalili & Lewis, 2016, Abdollahzadeh et al., 2021) during which the highest, average, and standard deviation has been recorded and preserved. Reporting of average and standard deviation can provide information about the stability of the performance of each respective algorithm (Mirjalili, 2015). However, due the stochastic nature of metaheuristic algorithms, statistical test should be considered in order to get a wider understanding of their performance (Derrac et al., 2011). Accordingly, conduct the Wilcoxon-rank sum test to further ensure comparative statistical significance (Derrac et al., 2011; Mirjalili & Lewis, 2013; Mirjalili, 2015). The Wilcoxon rank-sum test is a non-parametric statistical method, which identifies if there is a statistically significant difference between two groups. In the following section, we report the findings and emphasize the highest and average values observed for each outcome.

4.2. Results

The highest Sharpe ratio recorded in the experiments was 2.273. Initially, we will examine the highest, second highest, and lowest performing algorithms based on their maximum and average scores. Ultimately, we will present a comparative analysis of the algorithms' efficacy. In our initial experiment (Table 3), we employed a total of 30 epochs. As expected, choosing a population size of 10 would likely lead to weaker results due to the limited number of agents involved. AGTO attained the highest Sharpe ratio of 2.216. TSO attained the second-highest performance with a Sharpe ratio of 2.164, and GOA recorded the lowest performance at 0.961. AGTO outperforms all other metaheuristic algorithms in average performance, with ABC attaining the second-best results. Furthermore, GOA exhibited the lowest average performance with a Sharpe ratio of 0.722.

Subsequently, we employed a population size of 30. The overwhelming majority of the algorithms exhibited superior performance, as expected. AGTO and TSO attained the highest and second-highest Sharpe ratios of 2.266 and 2.245, respectively. In this context, CSA exhibited the poorest performance. All algorithms exhibited superior average performance compared to the prior experiment. In the following instances, with population sizes of 50 and 100, AGTO and TSO attained the highest Sharpe ratio scores of 2.270 and 2.273, respectively. However, AGTO achieved delivering a higher mean Sharpe ratio and a lower run-to-run standard deviation in comparison to TSO. For a population size of 50, MPA ranks second with a Sharpe ratio of 2.238, while for a population size of 100, WOA achieves the second-best performance with a Sharpe ratio of 2.261. Nonetheless, AGTO had greater average performance in both circumstances. In both instances, BA had the lowest performance, both in average and peak metrics.

Table 1
Tickers of DIJA.

Stock	Ticker	Stock	Ticker
3 M	MMM	IBM	IBM
American Express	AXP	Intel	INTC
Amgen	AMGN	Johnson & Johnson	JNJ
Amazon	AMZN	JPMorgan Chase	JPM
Apple	AAPL	McDonald's	MCD
Boeing	BA	Merck	MRK
Caterpillar	CAT	Microsoft	MSFT
Chevron	CVX	Nike	NKE
Cisco	CSCO	Procter & Gamble	PG
Coca-Cola	KO	Salesforce	CRM
Disney	DIS	Travelers	TRV
Dow	DOW	United Health Group	UNH
Goldman Sachs	GS	Verizon	VZ
Home Depot	HD	Visa	V
Honeywell	HON	Walmart	WMT

Table 2
Evaluated metaheuristic algorithms.

Algorithm
Artificial Bee Colony (ABC)
Artificial Gorilla Troop (AGTO)
Ant Lion Optimizer (ALO)
Aquila Optimizer (AO)
Artificial Rabbit Optimizer (ARO)
Bat Algorithm (BA)
Coyote Optimization Algorithm (COA)
Cuckoo Search Algorithm (CSA)
Grasshopper Optimization Algorithm (GOA)
Grey Wolf Optimizer (GWO)
Harris Hawk Optimizer (HHO)
Marines Predator Algorithm (MPA)
Particle Swarm Optimizer (PSO)
Sailfish Optimizer (SFO)
Social Spider Algorithm (SSA)
Optimization Algorithm (WOA)

Table 3
Optimization results utilizing 30 epochs.

	Maximum				Average				Standard deviation			
	Population size				Population size				Population size			
	10	30	50	100	10	30	50	100	10	30	50	100
ABC	2.029	1.929	1.955	1.987	1.815	1.854	1.894	1.914	**0.068**	0.046	0.036	0.039
AGTO	***2.216***	**2.266**	**2.270**	**2.273**	**1.892**	**2.173**	**2.228**	**2.246**	0.239	0.107	0.050	0.027
ALO	2.023	2.099	2.192	2.258	1.612	1.877	1.976	2.062	0.208	0.133	0.130	0.115
AO	1.887	1.951	2.055	2.014	1.142	1.353	1.365	1.450	0.345	0.273	0.285	0.208
ARO	2.087	2.213	2.213	2.238	1.758	1.956	2.029	2.131	0.215	0.122	0.124	0.062
BA	1.036	1.115	1.115	1.115	0.754	0.860	0.901	0.943	0.112	0.110	0.095	0.084
COA	1.835	1.722	2.008	1.897	1.368	1.550	1.676	1.705	0.206	0.129	0.142	0.112
CSA	1.076	1.097	1.228	1.192	0.911	0.981	1.017	1.069	0.073	0.060	0.069	0.058
GOA	0.961	1.188	1.387	1.722	0.722	0.969	1.139	1.452	0.122	0.103	0.125	0.118
GWO	1.901	2.118	2.196	2.226	1.779	2.059	2.130	2.200	0.074	**0.035**	**0.035**	**0.019**
HHO	2.109	2.166	2.166	2.260	1.588	1.846	1.902	2.007	0.221	0.205	0.177	0.148
MPA	1.702	2.165	2.238	2.240	1.267	1.716	1.943	2.037	0.157	0.180	0.156	0.102
PSO	1.219	1.437	1.419	1.476	1.096	1.202	1.254	1.307	0.068	0.086	0.062	0.060
SFO	1.501	1.581	1.590	1.762	1.149	1.278	1.364	1.446	0.162	0.126	0.115	0.148
SSA	1.531	1.647	1.695	1.901	1.286	1.466	1.550	1.636	0.122	0.094	0.079	0.085
TSO	2.164	2.245	**2.270**	**2.273**	1.802	2.062	2.104	2.168	0.227	0.148	0.126	0.107
WOA	2.101	2.231	2.231	2.261	1.640	1.893	1.968	2.057	0.258	0.228	0.172	0.158

Note: The table presents the performance of each metaheuristic algorithm based on experiments conducted over 30 epochs for each population size.

Subsequently, we employed 50 epochs with the same population sizes (Table 4). Assigning a population size of 10 yielded a Sharpe ratio of 2.241 for AGTO. AGTO outperformed the 30 epochs with the identical population size. Unlike the prior studies, ARO ranks second. In terms of average performance, ABC and GWO received the highest scores, recording 2.020 and 2.013, respectively. GOA attained the lowest Sharpe ratio for both maximum and average performance. AGTO once more attained the greatest Sharpe ratio (2.272) utilizing populations of 30 and 50. In the preceding scenario, ARO attained the second highest Sharpe Ratio scores of 2.257 and 2.266, respectively. In these cases, BA exhibited the poorest performance, recording a Sharpe ratio of 1.115. Regarding average performance, GWO exhibited superior results, while AGTO ranked second. In both cases, BA exhibited the worse performance on average. AGTO and WOA attained the greatest Sharpe ratio of 2.273 with a population size of 100. However, AGTO delivered a higher mean Sharpe across runs and the lowest standard deviation in comparison to WOA. The BA algorithm produced the poorest results. Consistent with the preceding two situations, GWO and AGTO attained the highest average performances.

In our concluding experiment, we employed a total of 100 epochs (Table 5). With the population size set to 10, we noted that ARO achieved the maximum performance with a Sharpe ratio of 2.269, while AGTO ranked second with a Sharpe ratio of 2.260. In this instance, GOA exhibited the poorest performance. Regarding average performance, GWO and ABC exhibited superior results. When the population size was fixed at 30, AGTO attained the highest Sharpe ratio of 2.273, succeeded by ARO and MPA at 2.271. In terms of average performance, GWO and AGTO exhibited superior results. Subsequently, we employed a population size of 50. AGTO, ARO, and MPA attained a Sharpe ratio of 2.273, closely followed by TSO at 2.272, while BA recorded the lowest Sharpe ratio of 1. 115. Compared with ARO and MPA, AGTO exhibited the lowest standard deviation of performance, indicating greater stability. GWO and AGTO had superior performance on average. With a population size of 100, four metaheuristic algorithms attained the highest Sharpe ratio: AGTO, ARO, MPA, and TSO. However, AGTO was the most consistent, exhibiting the lowest run-to-run standard deviation among

Table 4
Optimization results utilizing 50 epochs.

	Maximum				Average				Standard deviation			
	Population size				Population size				Population size			
	10	30	50	100	10	30	50	100	10	30	50	100
ABC	2.093	2.100	2.136	2.163	**2.020**	2.053	2.071	2.084	**0.044**	0.024	0.025	0.025
AGTO	**2.241**	**2.272**	**2.272**	**2.273**	1.963	2.205	**2.243**	2.258	0.264	0.069	0.036	0.024
ALO	2.148	2.216	2.248	2.248	1.759	1.981	2.060	2.119	0.215	0.137	0.114	0.099
AO	1.883	2.117	2.143	1.760	1.151	1.359	1.400	1.453	0.320	0.319	0.240	0.176
ARO	2.182	2.257	2.266	2.262	1.967	2.153	2.215	2.242	0.185	0.118	0.065	0.021
BA	1.036	1.115	1.115	1.115	0.759	0.860	0.902	0.943	0.109	0.110	0.094	0.084
COA	1.863	1.952	2.084	2.063	1.530	1.718	1.785	1.828	0.187	0.127	0.132	0.112
CSA	1.090	1.215	1.313	1.203	0.937	1.013	1.047	1.088	0.075	0.059	0.092	0.057
GOA	0.970	1.235	1.450	1.824	0.734	1.013	1.209	1.551	0.122	0.106	0.136	0.134
GWO	2.102	2.243	2.255	2.266	2.013	**2.217**	**2.243**	**2.261**	0.059	**0.013**	**0.009**	**0.003**
HHO	2.109	2.180	2.232	2.261	1.700	1.895	2.008	2.090	0.209	0.179	0.168	0.113
MPA	1.715	2.246	2.245	2.272	1.393	1.945	2.141	2.235	0.133	0.170	0.119	0.032
PSO	1.383	1.398	1.454	1.507	1.160	1.246	1.301	1.353	0.083	0.078	0.070	0.093
SFO	1.501	1.581	1.590	1.786	1.155	1.282	1.374	1.455	0.158	0.124	0.115	0.153
SSA	1.539	1.779	1.791	1.875	1.382	1.571	1.632	1.752	0.082	0.090	0.083	0.073
TSO	2.226	2.250	2.260	2.261	1.926	2.108	2.121	2.184	0.197	0.142	0.117	0.084
WOA	2.101	2.228	2.243	**2.273**	1.650	1.897	1.970	2.077	0.245	0.185	0.183	0.151

Note: The table presents the performance of each metaheuristic algorithm based on experiments conducted over 50 epochs for each population size.

Table 5
Optimization results utilizing 100 epochs.

	Maximum				Average				Standard deviation			
	Population size				Population size				Population size			
	10	30	50	100	10	30	50	100	10	30	50	100
ABC	2.243	2.243	2.246	2.245	2.216	2.226	2.232	2.235	**0.012**	0.009	0.008	0.004
AGTO	2.260	**2.273**	**2.273**	**2.273**	2.005	2.232	2.252	2.264	0.274	0.045	0.025	0.012
ALO	2.114	2.167	2.231	2.232	1.833	2.038	2.077	2.123	0.136	0.082	0.103	0.087
AO	1.950	2.164	2.164	2.164	1.274	1.473	1.529	1.556	0.298	0.256	0.266	0.223
ARO	**2.269**	2.271	**2.273**	**2.273**	2.089	2.203	2.238	2.265	0.146	0.099	0.076	0.021
BA	1.036	1.115	1.115	1.115	0.759	0.860	0.903	0.943	0.109	0.110	0.094	0.084
COA	2.020	2.140	2.131	2.132	1.765	1.933	1.969	2.005	0.179	0.099	0.093	0.082
CSA	1.090	1.215	1.313	1.263	0.976	1.045	1.095	1.124	0.049	0.053	0.067	0.060
GOA	0.995	1.331	1.585	1.993	0.761	1.098	1.327	1.693	0.124	0.123	0.142	0.147
GWO	2.254	2.269	2.269	2.271	**2.227**	**2.265**	**2.267**	2.270	0.017	**0.002**	**0.001**	**0.001**
HHO	2.100	2.228	2.257	2.261	1.764	1.928	2.011	2.079	0.225	0.205	0.156	0.122
MPA	2.052	2.271	**2.273**	**2.273**	1.697	2.217	2.253	**2.271**	0.193	0.093	0.074	0.006
PSO	1.311	1.435	1.494	1.559	1.189	1.292	1.321	1.381	0.078	0.066	0.067	0.080
SFO	1.501	1.581	1.610	1.839	1.161	1.299	1.400	1.473	0.155	0.109	0.128	0.156
SSA	1.679	1.872	1.924	2.054	1.541	1.709	1.770	1.869	0.086	0.094	0.100	0.086
TSO	2.246	2.259	2.272	**2.273**	2.029	2.112	2.192	2.243	0.140	0.126	0.077	0.043
WOA	2.176	2.261	2.261	2.231	1.742	1.918	1.987	2.070	0.215	0.193	0.191	0.130

Note: The table presents the performance of each metaheuristic algorithm based on experiments conducted over 100 epochs for each population size.

them, in the vast majority of cases. Moreover, GWO delivered the next-highest Sharpe ratio. Regarding average performance, MPA achieved the highest Sharpe ratio of 2.271, while GWO followed closely with a Sharpe ratio of 2.270. In the context of 100 epochs, BA had the poorest performance across all instances regarding both the maximum and average Sharpe ratio. Appendix A presents a condensed table summarizing the results for all configurations of epochs and population size.

4.3. Statistical analysis

To strengthen the validity of our findings, we performed the Wilcoxon rank-sum test, with the corresponding results displayed in Table 6, which reports p-values comparing each algorithm against AGTO for each combination of epoch and population size. In 192 comparisons, AGTO had greater performance in 175 instances relative to the other algorithms. Utilizing 30 epochs revealed statistically significant differences in nearly all instances between the performance of AGTO and the other algorithms. The sole exception arose with a population size of 10, where no statistically significant difference was detected between AGTO and TSO.

With 50 epochs and a population size of 10, we found that no definitive conclusions could be made about the superiority of AGTO compared to ABC, ARO, GWO, and TSO. Moreover, no conclusive results were noted for the statistical efficacy of AGTO in comparison to GWO for population sizes of 50 and 100. In the context of 100 epochs, a definitive conclusion about the comparative performance of

Table 6
Wilcoxon rank-sum test results per epoch and population size.

	30				50				100			
	10	30	50	100	10	30	50	100	10	30	50	100
	p-value				p-value				p-value			
ABC	1.2E-02	3.3E-10	2.9E-11	2.9E-11	8.4E-01	5.8E-10	2.9E-11	2.9E-11	5.3E-06	6.8E-03	9.2E-06	3.3E-10
ALO	7.0E-06	1.5E-09	2.3E-10	5.3E-10	1.5E-04	1.3E-08	1.6E-09	2.5E-10	1.1E-04	5.8E-11	8.6E-11	3.2E-11
AO	1.6E-09	3.5E-11	2.9E-11	2.9E-11	7.0E-10	3.9E-11	3.5E-11	2.9E-11	2.3E-09	5.8E-11	2.9E-11	2.9E-11
ARO	5.7E-03	4.5E-08	1.1E-09	3.6E-10	5.7E-01	4.8E-02	8.8E-04	2.4E-05	3.7E-01	3.1E-01	8.4E-01	1.7E-01
BA	3.2E-11	2.9E-11	2.9E-11	2.9E-11	3.2E-11	2.9E-11	2.9E-11	2.9E-11	3.2E-11	2.9E-11	2.9E-11	2.9E-11
COA	2.8E-09	2.9E-11	2.9E-11	2.9E-11	2.5E-08	2.9E-11	2.9E-11	2.9E-11	1.1E-05	3.2E-11	2.9E-11	2.9E-11
CSA	3.5E-11	2.9E-11	2.9E-11	2.9E-11	3.9E-11	2.9E-11	2.9E-11	2.9E-11	6.4E-11	2.9E-11	2.9E-11	2.9E-11
GOA	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11
GWO	4.3E-04	1.5E-06	3.5E-08	5.3E-10	6.7E-01	4.4E-02	8.6E-02	2.7E-01	1.7E-06	5.8E-02	1.8E-01	1.8E-01
HHO	3.7E-06	7.4E-09	7.0E-11	6.4E-10	1.3E-05	4.0E-10	7.0E-10	1.7E-10	3.7E-05	4.8E-10	4.4E-10	7.8E-11
MPA	7.7E-10	3.0E-10	2.3E-10	1.0E-10	1.6E-09	1.6E-08	1.1E-07	1.7E-04	1.9E-06	2.4E-01	2.3E-01	5.9E-01
PSO	4.0E-10	2.9E-11	2.9E-11	2.9E-11	5.3E-10	2.9E-11	2.9E-11	2.9E-11	4.8E-10	2.9E-11	2.9E-11	2.9E-11
SFO	4.0E-10	2.9E-11	2.9E-11	2.9E-11	4.4E-10	2.9E-11	2.9E-11	2.9E-11	4.8E-10	2.9E-11	2.9E-11	2.9E-11
SSO	9.3E-10	2.9E-11	2.9E-11	2.9E-11	2.1E-09	2.9E-11	2.9E-11	2.9E-11	1.4E-08	2.9E-11	2.9E-11	2.9E-11
TSO	9.8E-02	2.5E-04	4.0E-06	3.3E-04	1.8E-01	7.5E-04	6.3E-07	4.5E-08	4.8E-01	2.4E-06	2.7E-05	4.1E-04
WOA	8.9E-05	1.5E-07	2.1E-09	1.6E-08	7.5E-06	1.2E-09	5.8E-10	8.1E-09	2.1E-05	2.3E-09	1.6E-09	2.9E-11

Note: The table presents the p-values of Wilcoxon Rank-Sum Test of each metaheuristic versus AGTO for each combination of epoch and population sizes.

AGTO, ARO, GWO, and MPA in terms of statistical significance could not be established. As indicated in Tables 3–5, AGTO attained the greatest Sharpe ratio in most situations, highlighting its strong overall performance. Nonetheless, as will be examined in the subsequent part, GWO, TSO, ARO, MPA, and ABC showed solid performance in specific scenarios, underscoring their competitiveness.

4.4. *Comparative analysis of algorithms' performance*

The findings show that five of the evaluated algorithms failed to achieve a Sharpe ratio above 2 in any scenario. — BA, CSA, GOA, PSO, SFO. Furthermore, SSA successfully exceeded a Sharpe ratio above 2, solely in the most computationally demanding case of 100 epochs and population size. The six previously mentioned algorithms showed weaker average performance. Notably, while AO reached a high maximum Sharpe ratio, its average performance was relatively weak. ABC and ALO demonstrated consistent performance throughout many epochs and population sizes. Overall, ABC outperformed ALO alone in the context of 100 epochs across all population size combinations and in the instance of 30 epochs with 10 population size.

In the other instances, ALO demonstrated superiority. The HO demonstrated stability in outcomes across all combinations of epochs and population size. Nonetheless, performance exhibited an average deficiency in stability. Conversely, WOA seems to surpass HHO in most cases regarding the attainment of the maximum Sharpe ratio. However, regarding average performance, HHO somewhat surpasses WOA.

Concerning GWO, it demonstrated reliable performance, showing the lowest standard deviation across runs in most configurations. Nonetheless, it never attained the optimal Sharpe ratio. In contrast, regarding average performance, it exhibited consistent results and surpassed the other algorithms in the majority of instances. The other algorithms attained a maximum Sharpe ratio of 2.273. Upon selecting 100 epochs, MPA attained the optimal Sharpe ratio in two instances. However, with a population size of 10, MPA exhibited inferior performance relative to AGTO, ARO, and TSO. Setting the epoch number and population size at 100 yielded the optimal average performance across all methods.

In our examination of ARO and TSO performance, it is evident that TSO surpassed ARO in achieving the highest Sharpe ratio while employing 30 epochs. This tendency was also found with 50 epochs and a population size of 10. With the exception of cases involving 100 epochs and 100 population sizes, where a tie was noted, ARO surpassed TSO in all other instances. Regarding average performance, TSO exceeded only ARO in the scenario of 30 epochs. Notably, TSO equaled AGTO in scenarios involving 30 epochs and population sizes of 50 and 100, with both algorithms achieving the optimal Sharpe ratio. Ultimately, it is evident that AGTO demonstrated the most consistent performance.

In all instances, except for 100 epochs and a population size of 10, ARO exceeded it. AGTO not only attained superior performance but also consistently produced a Sharpe ratio over 2,260, except for 30 epochs and 10 population sizes. In this case, the performance exhibited potential. AGTO delivered the most consistent average performance across 30 epochs. In the remaining scenarios, it still performed well; yet it was marginally inferior to other methods.

Although AGTO was initially suggested as the method for portfolio optimization, it would be unfair to overlook the noteworthy performance of alternative metaheuristic algorithms. Such an example is TSO, which, as far as we know, is the first one that has been used for the concept of portfolio optimization. It is evident that certain cases were matched with AGTO. Nevertheless, the mean performance was inferior to that of AGTO. Furthermore, both ARO and MPA had remarkable performances. As far as we know, the application of these algorithms in portfolio optimization is limited. Furthermore, the findings indicated that the widely utilized GWO performs well and offers greater stability on average. However, none of the examples succeeded in achieving the optimal Sharpe ratio.

In conclusion, AGTO exhibited optimal performance in all instances, apart from one. Generally, one of the most formidable challenges that a practitioner or researcher may encounter is to address the trade-off between the number of epochs and population sizes. AGTO has reduced sensitivity to the selection of these components, as indicated by the attainment of the greatest Sharpe ratio in five out of twelve instances and a Sharpe ratio of no less than 2.270 in eight instances. Literature typically refrains from employing a sample size of 10. We have acquired significant insights on the performance of the assessed models. To that end, the results demonstrate that indeed the AGTO can be considered a viable algorithm for solving the case of optimal portfolio selection, since a researcher or practitioner can obtain optimal results across various combinations of epochs and population sizes.

The empirical findings of this study provide strong support for the hypotheses proposed at the end of the literature review. Hypothesis 1 (H1) is confirmed through consistent results showing that AGTO outperformed all sixteen competing metaheuristic algorithms in terms of maximum Sharpe ratio across many tested configurations. Hypothesis 2 (H2) is also substantiated, as AGTO maintained robust performance across different combinations of population sizes and epoch counts, with minimal degradation in outcome quality, demonstrating reduced sensitivity to parameter tuning. Regarding Hypothesis 3 (H3), AGTO exhibited high average performance and low standard deviation in most scenarios, suggesting a favorable balance between performance and stability. Lastly, Hypothesis 4 (H4) is validated through the Wilcoxon rank-sum tests, which revealed statistically significant differences in AGTO's performance when compared to most other algorithms, particularly at higher population and epoch levels.

Collectively, these findings demonstrate that AGTO is not only an effective optimizer for constrained portfolio problems but also a consistently reliable one under varying experimental conditions. The results highlight AGTO's potential as a practical tool for both academic researchers and financial practitioners aiming to implement metaheuristic-driven portfolio optimization.

This study is well aligned with previous research that employs metaheuristic algorithms for portfolio optimization, particularly those focusing on constrained scenarios, such as Mozafari et al. (2011), Mishra et al. (2014), and Bacanin & Tuba (2014). Like these works, it addresses practical constraints (e.g., no short-selling, budget limits) and adopts performance metrics such as the Sharpe ratio. However, it diverges meaningfully by expanding the comparative scope: while most prior studies benchmark a proposed method against a narrow set of classical algorithms, this paper evaluates sixteen distinct swarm intelligence-based metaheuristics, including lesser-studied approaches like TSO, ARO, and MPA. In doing so, it offers a broader, more up-to-date empirical foundation. Moreover, while earlier work (Agushaka et al., 2023; Si-Ma et al., 2025) emphasized the role of population size and iteration count, this study uniquely integrates those parameters into a financial optimization context. The inclusion and empirical evaluation of AGTO—a novel algorithm not yet applied to portfolio optimization in existing literature—represents a direct contribution to both computational finance and metaheuristic development. Thus, the present work not only consolidates and extends existing findings but also introduces new directions by integrating algorithmic innovation with rigorous benchmarking.

4.5. Policy and practical implications

The findings of this study carry several important implications for both policy-makers and financial practitioners. First, the superior and consistent performance of AGTO under constrained portfolio settings—such as no short-selling and full-budget allocation—suggests its practical applicability within regulatory-compliant investment frameworks. For institutional investors such as pension funds, mutual funds, and sovereign wealth vehicles that are bound by such constraints, AGTO offers a computationally efficient and stable alternative to traditional optimization methods.

Second, the reduced sensitivity of AGTO to hyperparameter tuning (i.e., population size and number of iterations) has significant implications for firms seeking scalable and robust algorithmic solutions. This attribute makes AGTO well-suited for integration into automated portfolio construction systems in the FinTech and robo-advisory sectors, where computational overhead and real-time responsiveness are critical.

From a policy standpoint, the adoption of transparent, repeatable, and non-deterministic optimization tools like AGTO aligns with emerging calls for increased algorithmic transparency and explainability in financial services. Regulatory bodies aiming to ensure fair and accountable use of AI in investment management may find AGTO's stochastic yet controlled architecture to be a favorable model for compliance-focused algorithm deployment.

Finally, the inclusion and benchmarking of newer metaheuristics—many of which had limited or no prior application in financial contexts—highlight the need for continuous reassessment of industry-standard models. Policy-makers and academic institutions supporting digital finance research may use this study as a basis for promoting innovation in quantitative finance curricula and regulatory sandboxes.

5. Conclusions and future research

This study proposes the use of Artificial Gorilla Troops Optimizer (AGTO) as a potential approach for portfolio optimization. We compared AGTO with sixteen other metaheuristic algorithms for a variety of scenarios with different epochs and population sizes for the concept of optimizing a portfolio of stocks from DJIA. The results demonstrated that AGTO is able to provide optimal results in almost all cases. Moreover, in terms of average performance, it consistently provides stable results, if not the best. This work not only proposes a well-known metaheuristic algorithm for the concept of portfolio optimization, which, as far as we know, this work is probably the first one to employ it for this task but moreover provides a rich comparison of various metaheuristic algorithms in this regard.

Furthermore, we gained valuable insights not only for popular metaheuristics algorithms among researchers, but we also evaluated the performance of algorithms that have limited application in the existing literature, specifically the concept of portfolio optimization tasks such as TSO. Another contribution of this work is the analysis of these algorithms' performance in relation to the number of epochs and population. The appropriate selection of these components generally affects the optimization performance. It's worth noting that in almost all combinations, except in one case, AGTO produced the maximum Sharpe ratio among other algorithms.

Given this, we can conclude that AGTO can provide optimal results for a variety of combinations, of population size and number of epochs. Future directions are to perform similar experiments on a larger dataset, as well as exploring more complex portfolio optimization problems. Furthermore, future research could involve comparing a greater number of metaheuristic algorithms for portfolio optimization, with the aim of further evaluating their performance.

CRediT authorship contribution statement

Vasileios Gkonis: Writing – original draft, Software, Methodology, Conceptualization. **Ioannis Tsakalos:** Writing – review & editing, Validation, Supervision, Project administration. **Ilias Kampouris:** Writing – review & editing, Validation, Supervision, Project administration.

Funding

The authors received no specific funding for this work.

Declaration of competing 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.

Appendix A

..

Algorithm 1 Pseudocode of AGTO

```
% GTO setting
Inputs : The population size N and maximum number of iterations T and parameters β and p
Outputs : The location of Gorilla and its fitness value
% initialization
Initialize the random population Xi (i = 1, 2, ···, N)
Calculate the fitness values of Gorilla
% Main Loop
while (stopping condition is not met) do
    Update the C using Equation (2)
    Update the L using Equation (4)
    % Exploration phase
  for (each Gorilla (Xi)) do
      Update the location Gorilla using Equation (1)
  endfor
    % Create group
    Calculate the fitness values of Gorilla
    if GX is better than X, replace them
    Set Xsilverback as the location of silverback (best location)
    % Exploitation phase
  for (each Gorilla (Xi)) do
    if (|C| ≥ 1) then
          Update the location Gorilla using Equation (7)
      Else
          Update the location Gorilla using Equation (10)
    Endif
  endfor
  % Create group
    Calculate the fitness values of Gorilla
    if New Solutions are better than previous solutions, replace them
    Set Xsilverback as the location of silverback (best location)
endwhile
Return XBestGorilla, bestFitness
```

Table A1
Summary of optimization results by epoch and population size.

	Epochs											
	30				50				100			
	Population Size											
	10	30	50	100	10	30	50	100	10	30	50	100
	Maximum											
ABC	2.029	1.929	1.955	1.987	2.093	2.100	2.136	2.163	2.243	2.243	2.246	2.245
AGTO	***2.216***	**2.266**	**2.270**	**2.273**	**2.241**	**2.272**	**2.272**	**2.273**	2.260	**2.273**	**2.273**	**2.273**
ALO	2.023	2.099	2.192	2.258	2.148	2.216	2.248	2.248	2.114	2.167	2.231	2.232
AO	1.887	1.951	2.055	2.014	1.883	2.117	2.143	1.760	1.950	2.164	2.164	2.164
ARO	2.087	2.213	2.213	2.238	2.182	2.257	2.266	2.262	**2.269**	2.271	**2.273**	**2.273**
BA	1.036	1.115	1.115	1.115	1.036	1.115	1.115	1.115	1.036	1.115	1.115	1.115
COA	1.835	1.722	2.008	1.897	1.863	1.952	2.084	2.063	2.020	2.140	2.131	2.132
CSA	1.076	1.097	1.228	1.192	1.090	1.215	1.313	1.203	1.090	1.215	1.313	1.263
GOA	0.961	1.188	1.387	1.722	0.970	1.235	1.450	1.824	0.995	1.331	1.585	1.993
GWO	1.901	2.118	2.196	2.226	2.102	2.243	2.255	2.266	2.254	2.269	2.269	2.271
HHO	2.109	2.166	2.166	2.260	2.109	2.180	2.232	2.261	2.100	2.228	2.257	2.261
MPA	1.702	2.165	2.238	2.240	1.715	2.246	2.245	2.272	2.052	2.271	**2.273**	**2.273**
PSO	1.219	1.437	1.419	1.476	1.383	1.398	1.454	1.507	1.311	1.435	1.494	1.559
SFO	1.501	1.581	1.590	1.762	1.501	1.581	1.590	1.786	1.501	1.581	1.610	1.839
SSA	1.531	1.647	1.695	1.901	1.539	1.779	1.791	1.875	1.679	1.872	1.924	2.054
TSO	2.164	2.245	**2.270**	**2.273**	2.226	2.250	2.260	2.261	2.246	2.259	2.272	**2.273**
WOA	2.101	2.231	2.231	2.261	2.101	2.228	2.243	**2.273**	2.176	2.261	2.261	2.231
	Average											
ABC	1.815	1.854	1.894	1.914	**2.020**	2.053	2.071	2.084	2.216	2.226	2.232	2.235
AGTO	**1.892**	**2.173**	**2.228**	**2.246**	1.963	2.205	**2.243**	2.258	2.005	2.232	2.252	2.264
ALO	1.612	1.877	1.976	2.062	1.759	1.981	2.060	2.119	1.833	2.038	2.077	2.123
AO	1.142	1.353	1.365	1.450	1.151	1.359	1.400	1.453	1.274	1.473	1.529	1.556
ARO	1.758	1.956	2.029	2.131	1.967	2.153	2.215	2.242	2.089	2.203	2.238	2.265
BA	0.754	0.860	0.901	0.943	0.759	0.860	0.902	0.943	0.759	0.860	0.903	0.943
COA	1.368	1.550	1.676	1.705	1.530	1.718	1.785	1.828	1.765	1.933	1.969	2.005
CSA	0.911	0.981	1.017	1.069	0.937	1.013	1.047	1.088	0.976	1.045	1.095	1.124
GOA	0.722	0.969	1.139	1.452	0.734	1.013	1.209	1.551	0.761	1.098	1.327	1.693
GWO	1.779	2.059	2.130	2.200	2.013	**2.217**	**2.243**	**2.261**	**2.227**	**2.265**	**2.267**	2.270
HHO	1.588	1.846	1.902	2.007	1.700	1.895	2.008	2.090	1.764	1.928	2.011	2.079
MPA	1.267	1.716	1.943	2.037	1.393	1.945	2.141	2.235	1.697	2.217	2.253	**2.271**
PSO	1.096	1.202	1.254	1.307	1.160	1.246	1.301	1.353	1.189	1.292	1.321	1.381
SFO	1.149	1.278	1.364	1.446	1.155	1.282	1.374	1.455	1.161	1.299	1.400	1.473
SSA	1.286	1.466	1.550	1.636	1.382	1.571	1.632	1.752	1.541	1.709	1.770	1.869
TSO	1.802	2.062	2.104	2.168	1.926	2.108	2.121	2.184	2.029	2.112	2.192	2.243
WOA	1.640	1.893	1.968	2.057	1.650	1.897	1.970	2.077	1.742	1.918	1.987	2.070
	Standard deviation											
ABC	**0.068**	0.046	0.036	0.039	**0.044**	0.024	0.025	0.025	**0.012**	0.009	0.008	0.004
AGTO	0.239	0.107	0.050	0.027	0.264	0.069	0.036	**0.024**	0.274	0.045	0.025	0.012
ALO	0.208	0.133	0.130	0.115	0.215	0.137	0.114	0.099	0.136	0.082	0.103	0.087
AO	0.345	0.273	0.285	0.208	0.320	0.319	0.240	0.176	0.298	0.256	0.266	0.223
ARO	0.215	0.122	0.124	0.062	0.185	0.118	0.065	0.021	0.146	0.099	0.076	0.021
BA	0.112	0.110	0.095	0.084	0.109	0.110	0.094	0.084	0.109	0.110	0.094	0.084
COA	0.206	0.129	0.142	0.112	0.187	0.127	0.132	0.112	0.179	0.099	0.093	0.082
CSA	0.073	0.060	0.069	0.058	0.075	0.059	0.092	0.057	0.049	0.053	0.067	0.060
GOA	0.122	0.103	0.125	0.118	0.122	0.106	0.136	0.134	0.124	0.123	0.142	0.147
GWO	0.074	**0.035**	**0.035**	**0.019**	0.059	**0.013**	**0.009**	**0.003**	**0.017**	**0.002**	**0.001**	**0.001**
HHO	0.221	0.205	0.177	0.148	0.209	0.179	0.168	0.113	0.225	0.205	0.156	0.122
MPA	0.157	0.180	0.156	0.102	0.133	0.170	0.119	0.032	0.193	0.093	0.074	0.006
PSO	0.068	0.086	0.062	0.060	0.083	0.078	0.070	0.093	0.078	0.066	0.067	0.080
SFO	0.162	0.126	0.115	0.148	0.158	0.124	0.115	0.153	0.155	0.109	0.128	0.156
SSA	0.122	0.094	0.079	0.085	0.082	0.090	0.083	0.073	0.086	0.094	0.100	0.086
TSO	0.227	0.148	0.126	0.107	0.197	0.142	0.117	0.084	0.140	0.126	0.077	0.043
WOA	0.258	0.228	0.172	0.158	0.245	0.185	0.183	0.151	0.215	0.193	0.191	0.130

Data availability

The data was acquired via Yahoo Finance (https://finance.yahoo.com/).

References

Abdollahzadeh, B., Soleimanian Gharehchopogh, F., & Mirjalili, S. (2021). Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *International Journal of Intelligent Systems, 36*(10), 5887–5958. https://doi.org/10.1002/int.22535

Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering, 157*, Article 107250. https://doi.org/10.1016/j.cie.2021.107250

Agushaka, J. O., Ezugwu, A. E., Abualigah, L., Alharbi, S. K., & Khalifa, H. A. E. W. (2023). Efficient initialization methods for population-based metaheuristic algorithms: A comparative study. *Archives of Computational Methods in Engineering, 30*(3), 1727–1787. https://doi.org/10.1007/s11831-022-09850-4

Ahmad, G., & Shahid, M. (2022). Towards portfolio selection in stock markets using grey wolf optimization approach. In *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*. https://doi.org/10.1109/icdabi56818.2022.10041457

Arnone, S., Loraschi, A., & Tettamanzi, A. (1993). A genetic approach to portfolio selection. *Neural Network World, 3*(6), 597–604.

Bacanin, N., & Tuba, M. (2014). Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint. *The Scientific World Journal, 2014*(1), Article 721521. https://doi.org/10.1155/2014/721521

Chen, W. (2015). Artificial bee colony algorithm for constrained possibilistic portfolio optimization problem. *Physica A: Statistical Mechanics and its Applications, 429*, 125–139. https://doi.org/10.1016/j.physa.2015.02.060

Chen, R. R., Huang, W. K., & Yeh, S. K. (2021). Particle swarm optimization approach to portfolio construction. *Intelligent Systems in Accounting, Finance and Management, 28*(3), 182–194. https://doi.org/10.1002/isaf.1498

Derrac, J., García, 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. https://doi.org/10.1016/j.swevo.2011.02.002

Dhal, K. G., Das, A., Sahoo, S., Das, R., & Das, S. (2021). Measuring the curse of population size over swarm intelligence-based algorithms. *Evolving Systems, 12*, 779–826. https://doi.org/10.1007/s12530-019-09318-0

Di Tollo, G., & Roli, A. (2008). Metaheuristics for the portfolio selection problem. *International Journal of Operations Research, 5*(1), 13–35.

Doering, J., Kizys, R., Juan, A. A., Fito, A., & Polat, O. (2019). Metaheuristics for rich portfolio optimisation and risk management: Current state and future trends. *Operations Research Perspectives, 6*, Article 100121. https://doi.org/10.1016/j.orp.2019.100121

Dorigo, M., Birattari, M., & Stutzle, T. (2007). Ant colony optimization. *IEEE Computational Intelligence Magazine, 1*(4), 28–39. https://doi.org/10.1109/MCI.2006.329691

Dueck, G., & Winker, P. (1992). New concepts and algorithms for portfolio choice. *Applied Stochastic Models and Data Analysis, 8*(3), 159–178. https://doi.org/10.1002/asm.3150080306

Dupačová, J., & Kopa, M. (2014). Robustness of optimal portfolios under risk and stochastic dominance constraints. *European Journal of Operational Research, 234*(2), 434–441. https://doi.org/10.1016/j.ejor.2013.06.018

Erwin, K., & Engelbrecht, A. (2023). Meta-heuristics for portfolio optimization. *Soft Computing, 27*(24), 19045–19073. https://doi.org/10.1007/s00500-023-08177-x

Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine predators algorithm: A nature-inspired metaheuristic. *Expert systems with applications, 152*, Article 113377. https://doi.org/10.1016/j.eswa.2020.113377

Grishina, N., Lucas, C. A., & Date, P. (2017). Prospect theory–based portfolio optimization: An empirical study and analysis using intelligent algorithms. *Quantitative Finance, 17*(3), 353–367. https://doi.org/10.1080/14697688.2016.1149611

Glover, F. (1989). Tabu search—part I. *ORSA Journal on Computing, 1*(3), 190–206.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation, 76*(2), 60–68. https://doi.org/10.1177/003754970107600201

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems, 97*, 849–872. https://doi.org/10.1016/j.future.2019.02.028

Holland, J. H. (1992). Genetic algorithms. *Scientific American, 267*(1), 66–73.

Hussien, A. G., Bouaouda, A., Alzaqebah, A., Kumar, S., Hu, G., & Jia, H. (2024). An in-depth survey of the artificial gorilla troops optimizer: Outcomes, variations, and applications. *Artificial Intelligence Review, 57*(9), 246. https://doi.org/10.1007/s10462-024-10838-8

James, J. Q., & Li, V. O. (2015). A social spider algorithm for global optimization. *Applied Soft Computing, 30*, 614–627. https://doi.org/10.1016/j.asoc.2015.02.014

Jamous, R. A., Tharwat, A. A., El Seidy, E., & Bayoumi, B. (2015). A new particle swarm with center of mass optimization. *International Journal of Engineering Research and Technology, 4*(5), 312–317.

Jiang, X., & Li, S. (2017). Beetle antennae search without parameter tuning (BAS-WPT) for multi-objective optimization. https://arxiv.org/abs/1710.10724.

Jin, X., Chen, N., & Yuan, Y. (2019). Multi-period and tri-objective uncertain portfolio selection model: A behavioral approach. *The North American Journal of Economics and Finance, 47*, 492–504. https://doi.org/10.1016/j.najef.2018.06.005

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

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report-TR06. *Department of Computer Engineering, Engineering Faculty*. Erciyes University.

Khan, A. T., Cao, X., Brajevic, I., Stanimirovic, P. S., Katsikis, V. N., & Li, S. (2022). Non-linear activated beetle antennae search: A novel technique for non-convex tax-aware portfolio optimization problem. *Expert Systems with Applications, 197*, Article 116631. https://doi.org/10.1016/j.eswa.2022.116631

Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, pp. 1942-1948. Doi: 10.1109/ICNN.1995.488968.

Kirkpatrick, S., Gelatt, C. D., Jr, & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Liu, M., Luo, K., Zhang, J., & Chen, S. (2021). A stock selection algorithm hybridizing grey wolf optimizer and support vector regression. *Expert Systems with Applications, 179*, Article 115078. https://doi.org/10.1016/j.eswa.2021.115078

Łukasik, S., Żak, S. (2009). Firefly algorithm for continuous constrained optimization tasks. Computational Collective Intelligence: Semantic Web, Social Networks and Multiagent Systems, Wroclaw, Poland, 5-7 October. Doi: 10.1007/978-3-642-04441-0_8.

Markowitz, H. (1952). Portfolio selection. *Journal of Finance, 7*, 77–91. https://doi.org/10.2307/2975974

Medvedeva, M. A., Katsikis, V. N., Mourtas, S. D., & Simos, T. E. (2021). Randomized time-varying knapsack problems via binary beetle antennae search algorithm: Emphasis on applications in portfolio insurance. *Mathematical Methods in the Applied Sciences, 44*(2), 2002–2012. https://doi.org/10.1002/mma.6904

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation, 9*, 1–14. https://doi.org/10.1016/j.swevo.2012.09.002

Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software, 83*, 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010

Mishra, S. K., Panda, G., & Majhi, R. (2014). A comparative performance assessment of a set of multiobjective algorithms for constrained portfolio assets selection. *Swarm and Evolutionary Computation, 16*, 38–51. https://doi.org/10.1016/j.swevo.2014.01.001

Mourtas, S. D., & Katsikis, V. N. (2022). V-Shaped BAS: Applications on large portfolios selection problem. *Computational Economics, 60*(4), 1353–1373. https://doi.org/10.1007/s10614-021-10184-9

Mozafari, M., Jolai, F., & Tafazzoli, S. (2011). A new IPSO-SA approach for cardinality constrained portfolio optimization. *International Journal of Industrial Engineering Computations, 2*(2), 249–262. https://doi.org/10.5267/j.ijiec.2011.01.004

Mishra, S., Padhy, S., Mishra, S. N., & Misra, S. N. (2021). A novel LASSO–TLBO–SVR hybrid model for an efficient portfolio construction. *The North American Journal of Economics and Finance, 55*, Article 101350. https://doi.org/10.1016/j.najef.2020.101350

Pagnoncelli, B. K., Cifuentes, A., & Denis, G. (2017). A two-step hybrid investment strategy for pension funds. *The North American Journal of Economics and Finance, 42*, 574–583. https://doi.org/10.1016/j.najef.2017.09.001

Pai, G. V., & Michel, T. (2014). Metaheuristic multi-objective optimization of constrained futures portfolios for effective risk management. *Swarm and Evolutionary Computation, 19*, 1–14. https://doi.org/10.1016/j.swevo.2014.08.002

Pierezan, J., & Coelho, L. D. S. (2018). Coyote optimization algorithm: A new metaheuristic for global optimization problems. In *2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro*. https://doi.org/10.1109/CEC.2018.8477769

Piotrowski, A. P. (2017). Review of differential evolution population size. *Swarm and Evolutionary Computation, 32*, 1–24. https://doi.org/10.1016/j.swevo.2016.05.003

Piotrowski, A. P., Napiorkowski, J. J., & Piotrowska, A. E. (2020). Population size in particle swarm optimization. *Swarm and Evolutionary Computation, 58*, Article 100718. https://doi.org/10.1016/j.swevo.2020.100718

Radcliffe, N. J. (1993). Genetic set recombination. In D. Whitley (Ed.), *Foundations of Genetic Algorithms 2* (pp. 203–219). San Mateo, CA: Morgan Kaufmann. https://doi.org/10.1016/B978-0-08-094832-4.50019-2.

Rajwar, K., Deep, K., & Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges. *Artificial Intelligence Review, 56*(11), 13187–13257. https://doi.org/10.1007/s10462-023-10470-y

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*(13), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

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

Saiz, M., Lostumbo, M. A., Juan, A. A., & Lopez-Lopez, D. (2022). A clustering-based review on project portfolio optimization methods. *International Transactions in Operational Research, 29*(1), 172–199. https://doi.org/10.1111/itor.12933

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence, 80*, 20–34. https://doi.org/10.1016/j.engappai.2019.01.001

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance, 19*(3), 425–442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x

Shi, Y., & Eberhart, R. C. (1999, July). Empirical study of particle swarm optimization. In Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 3, pp. 1945-1950). IEEE.

Si-Ma, S., Liu, H. M., Zhan, H. X., Liu, Z. F., Guo, G., Yu, C., & Hu, P. C. (2025). Efficient maximum iterations for swarm intelligence algorithms: A comparative study. *Artificial Intelligence Review, 58*(3), 87. https://doi.org/10.1007/s10462-024-11104-7

Van Thieu, N., & Mirjalili, S. (2023). MEALPY: An open-source library for latest meta-heuristic algorithms in Python. *Journal of Systems Architecture, 139*, Article 102871. https://doi.org/10.1016/j.sysarc.2023.102871

Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W. (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 114*, Article 105082. https://doi.org/10.1016/j.engappai.2022.105082

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

Xie, L., Han, T., Zhou, H., Zhang, Z. R., Han, B., & Tang, A. (2021). Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization. *Computational Intelligence and Neuroscience, 2021*(1), Article 9210050. https://doi.org/10.1155/2021/9210050

Yang, X. S., Deb, S., (2009). Cuckoo search via Lévy flights. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9-11 December. Doi: 10.1109/NABIC.2009.5393690.