Meta-Heuristic Algorithm for Non-convex Risk Parity Portfolio

Rosita Kusumawati, *Member, IAENG,* Dedi Rosadi*, *Member, IAENG,* and Abdurakhman, *Member, IAENG*

***Abstract*—The non-convex Risk Parity (RP) portfolio optimization presents challenges due to the potential presence of multiple local minima, making it difficult to identify the optimal solution. Meta-heuristic algorithms, known for their flexibility, are ideal for addressing this issue, as they effectively balance exploration of new solution spaces with refinement of promising candidates. This study compares the performance of three meta-heuristic algorithms— Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization for continuous domains ($ACO_{\mathbb{R}}$)—in solving the non-convex RP portfolio optimization problem. Using both real-world and simulated datasets, the first empirical study demonstrates the superior performance of PSO. A second study, employing the rolling-window method, evaluates the RP portfolio against the Equally Weighted (EW) and Global Minimum Variance (GMV) portfolios. The results show that, while the RP portfolio does not consistently outperform the others across all metrics, it excels in minimizing Maximum Drawdown (MD) and Value-at-Risk (VaR). This research contributes to the literature by offering a thorough comparison of meta-heuristic algorithms for non-convex RP portfolio optimization and highlighting the RP portfolio's robustness in risk management.**

***Index Terms*—non-convex, risk parity portfolio, genetic algorithm, particle swarm optimization, ant colony optimization.**

I. INTRODUCTION

INVESTORS use stock diversification to mitigate the risk of financial losses in their portfolios [1]. A diversified portfolio consists of carefully chosen equities aimed at reducing overall risk. The first mathematical model for portfolio diversification was the Mean-Variance (MV) Markowitz optimization, introduced by Markowitz in 1952 [2]. The MV portfolio seeks to maximize expected return for a given level of risk or minimize risk for a specified return, determining the optimal weights for each asset based on the covariance matrix and expected returns.

The MV method achieves diversification by reducing risk, but it often results in a concentration of low-risk stocks, which contradicts the principle of diversification. Additionally, it lacks proper indicators for assessing portfolio diversity, limiting investors in constructing a fully diversified portfolio. The 2008 global financial crisis highlighted the need for better diversification strategies [3]. Several methods for measuring diversification have been explored extensively [4]. The Risk Parity (RP) portfolio, which measures diversification by evaluating the risk contribution of each asset, offers a more balanced approach. This method decomposes total portfolio risk into individual asset contributions, with a well-diversified portfolio ensuring that risk is evenly distributed. The RP portfolio allocates capital so that each asset contributes equally to the overall risk [5], [6]. Introduced by [7], the equal risk contribution concept has proven to be more robust than the MV portfolio under various financial conditions [8].

Constructing an RP portfolio is challenging due to its non-convexity. While common numerical optimization techniques like Sequential Quadratic Programming (SQP) and Interior Point Method (IPM) are typically used for non-convex problems [9], these methods can be inefficient and sometimes fail to achieve global convergence [5], [10]. To address this, researchers have explored alternative approaches for optimizing non-convex RP portfolios. The Least-Squares (LS) method was introduced to tackle non-convexity in RP portfolios [11], and the Successive Convex Optimization for Risk Parity (SCRIP) method, which iteratively approximates non-convex problems with convex sub-problems, was also proposed. However, SCRIP does not handle non-convex constraints [12]. Additionally, inexact accelerated gradient descent was suggested for solving the quadratic approximation sub-problem in parallel [13]. Despite these efforts, finding efficient solutions to non-convex RP portfolio optimization problems remains a complex and ongoing area of research.

Meta-heuristic methods offer an effective solution to the challenges of non-convex optimization problems, particularly in complex domains where they excel at exploring the solution space and identifying high-quality solutions. These approaches have been widely applied to the RP portfolio optimization. For example, a hybrid Genetic Algorithm (GA) and Local Search (LoS) algorithm with elitist selection, mutation, and crossover operators has successfully addressed the non-convex RP portfolio optimization problem [14]. An Evolutionary Algorithm (EA) with Adaptive Operator Selection (AOS), which dynamically selects the appropriate operator during the search process, has demonstrated strong performance in solving long-only non-convex RP portfolio optimization problems [15]. Particle Swarm Optimization (PSO), a swarm-based meta-heuristic, has been shown to effectively handle the robust RP portfolio problem, accounting for uncertainties in the covariance matrix [16]. Additionally, Ant Colony Optimization for continuous domains ($ACO_{\mathbb{R}}$) has been compared with the Hall of Fame (HoF) and Differential Evolution (DE) algorithms for the MV portfolio model with a risk parity constraint [17].

This paper contributes to the existing literature by exploring the benefits of applying meta-heuristic methods

Manuscript received July 29, 2024; revised January 11, 2025. This work was supported in part by the Indonesia Endowment Funds for Education (LPDP) and the Center for Higher Education Funding (BPPT) under Beasiswa Pendidikan Indonesia grant number 01877/BPPT/BPI.06/9/2023.

R. Kusumawati is a PhD candidate of Mathematics Department, Universitas Gadjah Mada, Yogyakarta, 55281 Indonesia. (e-mail: rositakusumawati@mail.ugm.ac.id)

D. Rosadi is a professor of Mathematics Department, Universitas Gadjah Mada, Yogyakarta, 55281 Indonesia. (corresponding author to provide phone: (0274) 552243; fax: (0274) 555131; e-mail: dedirosadi@ugm.ac.id)

Abdurakhman is a professor of Mathematics Department, Universitas Gadjah Mada, Yogyakarta, 55281 Indonesia. (e-mail: rachmanstat@ugm.ac.id)

to the non-convex RP portfolio optimization problem and demonstrating the robustness of RP portfolios in risk management. The construction of non-convex RP portfolios is simulated under two scenarios—long-only and long-short constraints—using GA, PSO, and $ACO_{\mathbb{R}}$ on both real and simulated datasets of varying scales. The performance of these methods in solving non-convex problems is evaluated and compared to traditional portfolio strategies such as Equally Weighted (EW) and Global Minimum Variance (GMV) portfolios, using the rolling window approach. The next section introduces the non-convex RP portfolio problem and provides an overview of the GA, PSO, and $ACO_{\mathbb{R}}$ algorithms. Section 3 outlines the study's methodology, followed by the presentation and discussion of results in Section 4, which includes a detailed analysis of the algorithms and portfolio performance. The paper concludes in Section 5.

II. Literature Review

A. Risk Parity Portfolio

The objective of the RP portfolio optimisation problem is to construct a portfolio of assets where the weights are determined primarily on the equal risk contribution of each asset [5], [6]. The quantification of portfolio risk can be expressed in the following manner,

$$\sigma(\mathbf{w}) = \sqrt{\mathbf{w^T \Sigma w}} = \sqrt{\sum_{i=1}^{n} \mathrm{w}_i^2 \sigma_i^2 + \sum_{i=1}^{n} \sum_{j=1, i\neq j}^{n} 2\mathrm{w}_i \mathrm{w}_j \sigma_{ij}} \quad (1)$$

The risk contribution of the i-th asset to the portfolio risk can be stated in the following manner,

$$RC_i = \sigma_i(\mathbf{w}) = \mathrm{w}_i \frac{(\mathbf{\Sigma w})_i}{\sqrt{\mathbf{w^T \Sigma w}}} = \frac{\mathrm{w}_i^2 \sigma_i^2 + \sum_{j=1, j\neq i}^{n} \mathrm{w}_i \mathrm{w}_j \sigma_{ij}}{\sqrt{\mathbf{w^T \Sigma w}}} \quad (2)$$

The Relative Risk Contribution (RRC) of the i-th asset is defined as the ratio between the risk contribution of the i-th asset and the risk of the entire portfolio. This can be expressed as,

$$RRC_i = \frac{RC_i}{\sigma(\mathbf{w})} = \frac{\mathrm{w}_i (\mathbf{\Sigma w})_i}{\mathbf{w^T \Sigma w}} = \frac{\mathrm{w}_i^2 \sigma_i^2 + \sum_{j=1, j\neq i}^{n} \mathrm{w}_i \mathrm{w}_j \sigma_{ij}}{\sigma^2(\mathbf{w})} \quad (3)$$

and $\sum_{i=1}^{n} RRC_i = 1$. To give an example, a RP portfolio consist of 6 stocks will have an equal risk contribution for each assets and an RRC of $1/6$ for each asset.

The RP portfolio allocates each asset in a manner that ensures each asset contributes an equal amount of risk to the overall portfolio risk. The RP portfolio optimization problem can be formulated as Least Square optimization problem, which can be represented as follows,

$$\text{minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\mathrm{w}_i (\mathbf{\Sigma w})_i - \mathrm{w}_j (\mathbf{\Sigma w})_j\right)^2 \quad (4)$$

subject to

$$\boldsymbol{e^T}\mathbf{w} = 1 \quad (5)$$

Equation (4) minimize the total quadratic difference between the risk contribution of asset i and asset j whilst Equation (5) ensures that the sum of asset weights is equal to one. The equivalent form of the RP portfolio problem in Equation (4) − (5) is as follows,

$$\text{minimize} \sum_{i=1}^{n} \left(\mathrm{w}_i (\mathbf{\Sigma w})_i - \frac{\sum_{j=1}^{n} \mathrm{w}_j (\mathbf{\Sigma w})_j}{n}\right)^2 \quad (6)$$

subject to

$$\boldsymbol{e^T}\mathbf{w} = 1 \quad (7)$$

B. Meta-Heuristic for The Non-Convex Risk Parity Portfolio

Non-convex optimization problems often feature complex objective functions with multiple local minima or non-smooth landscapes, making them difficult to solve using traditional gradient-based methods, which typically only find local optima. Meta-heuristic algorithms are designed to efficiently explore large, complex solution spaces while avoiding local minima. These methods do not require convexity and use stochastic or heuristic approaches to provide approximate solutions by balancing exploration and exploitation of the search space. As a result, meta-heuristics are widely used to tackle non-convex problems in fields such as machine learning, operations research, and engineering [18], [19], [20], [21], [22], [23], [24], [25].

Meta-heuristic algorithms are classified into two groups: those based on metaphors and those not. Non-metaphor algorithms do not rely on simulations for the solution search strategy, while metaphor-based algorithms simulate natural events, human behavior, or mathematical processes. Algorithms inspired by biological evolution, in particular, use various biological metaphors, with different approaches to representing solutions. These algorithms are typically grouped into three main paradigms: evolutionary systems, swarm intelligence, and immune systems [26]. This article focuses specifically on the evolutionary systems and swarm intelligence paradigms.

1) Genetic Algorithm: A Genetic Algorithm (GA) is a meta-heuristic optimization method inspired by biological evolution. Introduced by John Holland in the 1960s [27], GA operates on a population of candidate solutions, known as chromosomes, represented as $C_i^g = \{c_1^g, c_2^g, ..., c_n^g\}$ where $c_i^g \in \mathbb{R}$ or $c_i^g \in \mathbb{Z}$ and g is the generation number, with each chromosome corresponding to a potential solution. The algorithm iteratively improves these solutions through selection, crossover, and mutation. At each generation, solutions are evaluated using a fitness function, and the fittest individuals are selected to produce offspring, enhancing the survival of the most promising solutions. This process continues across generations, with the population evolving toward an optimal or near-optimal solution.

The GA can be outlined in the following steps: (i) determine the variable types, encoding scheme, fitness function, and genetic operators with their respective probabilities; (ii) initialize a random population of chromosomes; (iii) perform selection, where individuals with higher fitness values are more likely to be chosen as parents for the next generation; (iv) perform crossover, which combines genetic material from two parent chromosomes selected in the previous step to create offspring with potentially improved characteristics; (v) perform mutation, which introduces small random alterations to the offspring's genes, maintaining genetic diversity and preventing premature convergence; and (vi) iteratively repeat the selection, crossover, and mutation processes until a stopping condition, such as a predefined number of generations or achieving a satisfactory fitness level, is met.

A GA offers a flexible approach to portfolio optimization, including unconstrained problems that use downside risk measures and those involving various risk metrics such as mean-variance, semi-variance, mean absolute deviation, and skewness with cardinality restrictions [28], [29]. It can also address complex scenarios, such as incorporating market capitalization and cardinality constraints [30]. When fuzzy numbers are used to manage uncertainties in stock returns, the GA is efficient for portfolio optimization [31], [32]. Additionally, the GA handles anomalous data and deviations from normal distribution assumptions, excelling with cardinality constraints [33], [34]. Over time, the GA evolves to solve multi-objective optimization problems, with the Non-dominated Sorting Genetic Algorithm II (NSGA-II) proving effective for such tasks [35], [36]. Its performance is further enhanced through hybrid approaches, such as combining the GA with the ϵ-constraint method for optimizing the Mean Absolute Semi-Deviation (MASD)-Skewness portfolio under uncertainty [37], or with the Firefly Algorithm (FA) for Mean-Variance-Skewness optimization [38]. Additionally, hybrid GA with Local Search (LS) is successfully applied to long-only and short-term portfolio problems [14].

GA utilize various selection techniques, such as linear-rank selection, non-linear rank selection, proportional (roulette wheel) selection, tournament selection, fitness proportional selection with linear scaling, and fitness proportional selection with Goldberg's sigma truncation scaling [39]. Crossover, a key GA operator, combines two parent chromosomes to create a new offspring chromosome with a crossover probability, denoted as P_X. The GA employs several crossover methods, including single-point crossover, multi-point crossover, linear crossover, uniform crossover, global uniform crossover, queen-bee crossover, arithmetic and average crossover, simplex crossover, geometrical crossover, direction-based crossover, heuristic crossover, flat crossover, blend crossover, simulated binary crossover, Laplace crossover, parent-centric normal crossover, and unimodal normal distribution crossover [40], [15]. Mutation, another GA operator, helps prevent the population from becoming homogeneous and stagnating by randomly altering the genes of a selected chromosome with a mutation probability, denoted as P_Y.

There is a wide array of operator choices for producing offspring. Nevertheless, GA operators used here are as follows,

- **Fitness Linear Scaling Selection** Fitness proportional selection with linear scaling follows the same basic principles as the roulette wheel selection method, but with an added fitness linear scaling step [39]. This method is used in GA to refine the selection process by adjusting the fitness values of individuals in the population. The primary aim is to prevent the most fit individuals from dominating the selection too early and to facilitate a more gradual progression toward the optimal solution. In this approach, fitness proportional selection with linear scaling can be thought of as a weighted random sampling, where individuals are selected based on their scaled fitness relative to the total scaled fitness of the entire population. The process begins by calculating the fitness value f_i for each individual i, followed by the computation of the mean fitness of the population, as shown in the equation:

$$\bar{f} = \sum_{i=1}^{N} f_i \tag{8}$$

where N is the number of individuals in the population. Next, the maximum and minimum fitness is computed i.e.

$$f_{max} = \max\left\{f_1, f_2, ..., f_N\right\} \tag{9}$$

$$f_{min} = \min\left\{f_1, f_2, ..., f_N\right\} \tag{10}$$

Then, scaling parameters are defined to adjust the fitness values using the following formulas respectively,

$$a = \frac{f_{max} - f_{min}}{f_{max} - \bar{f}} \tag{11}$$

$$b = f_{min} - a\bar{f} \tag{12}$$

Here, the constants a and b are selected in such a way that the scaled fitness values preserve the relative order of the original fitness values while ensuring that no scaled fitness values become negative. The scaled fitness values are then computed using the formula:

$$f_i^s = af_i + b \tag{13}$$

Next, the total fitness of the population is calculated based on the scaled fitness values, as shown in the following equation:

$$F_{total}^s = \sum_{i=1}^{N} f_i^s \tag{14}$$

The selection probability p_i and the cumulative probability distribution C_i for each individual are then determined using the respective formulas:

$$p_i = \frac{f_i^s}{F_{total}^s} \tag{15}$$

$$C_i = \sum_{j=1}^{i} f_j^s \tag{16}$$

Next, select a number of k individuals which satisfy $C_{k-1} \le r \le C_k$ based on random numbers uniformly distributed between 0 and 1 that is generated, where is C_0 defined as 0 for convenience.

- **Arithmetic Crossover** Arithmetic Crossover operator that defines a linear combination of two chromosomes [40]. Two chromosomes, selected randomly for crossover, C_y^g and C_x^g may produce two off-springs, C_i^{g+1} and C_j^{g+1}, which is a linear combination of their parents i.e,

$$C_i^{g+1} = \alpha C_i^g + (1-\alpha)\, C_j^g \tag{17}$$

$$C_j^{g+1} = (1-\alpha)\, C_i^g + \alpha C_j^g \tag{18}$$

For each gene i, a random number r uniformly distributed in the interval $[0, 1]$ is generated. The crossover process for gene c_i will be carried out if

$r \leq P_X$.

- **Uniform Random Mutation** Uniform mutation is a GA operator used to introduce variability into the population by randomly altering the genes of an individual. This operator randomly selects one or more genes and assigns them new values within a predefined range. For each gene i, a random number r uniformly distributed in the interval $[0, 1]$ is generated. The mutation process for gene c_i will be carried out if $r \leq P_Y$. For gene c_i in individual $\mathbf{c} = (c_1, c_2, ..., c_n)$ selected for mutation, a new value $c_i \sim \text{UNIF}(c_{min}, c_{max})$ is constructed.

In GA for non-convex portfolio optimization, the goal is to identify the best chromosome as the optimal solution. Each chromosome in the g^{th} generation, represented as $C_i^g = \{c_1^g, c_2^g, ..., c_n^g\}$, corresponds to the weight of asset i for $i = 1, 2, ..., n$, with $c_i^g \in \mathbb{R}$. The objective function evaluates candidate solutions, and in non-convex optimization, the best chromosome minimizes the objective function (6). The fitness function ranks chromosomes by their optimality, favoring better solutions for reproduction through selection, crossover, and mutation. The solution must also meet the equality constraint (7), which can be handled using methods like penalty functions, decoders, or task-specific operators [41]. In this case, a penalty term is introduced to penalize deviations from the equality constraint, guiding the algorithm to find solutions that closely satisfy it. The fitness function (19), which minimizes the sum of deviations in risk contributions and the absolute difference between total weights and 1, is used.

$$f = \sum_{i=1}^{n} \left(\mathrm{w}_i (\mathbf{\Sigma w})_i - \frac{\sum_{j=1}^{n} \mathrm{w}_j (\mathbf{\Sigma w})_j}{n} \right)^2 + \left| \sum_{i=1}^{n} \mathrm{w}_i - 1 \right| \tag{19}$$

2) Particle Swarm Optimization: PSO algorithm, introduced by Eberhart and Kennedy in 1995, is a metaphor-based metaheuristic inspired by the swarm behavior observed in nature, such as in birds, ants, and bees [42]. It uses a population of particles to explore potential solutions in a random, yet guided search. Each particle, representing a possible solution, has a position and velocity in a multidimensional space, with its position updated over iterations based on its own best solution called personal best (Pbest) and the best solution among all the particles called global best (GBest). The algorithm's process of evolution relies on a fitness function to evaluate solution quality. Known for its simplicity, fast convergence, and minimal parameter requirements, PSO has become widely used and has evolved over time, both in applications and algorithmic refinement. The position and velocity of particle i at iteration t are represented as:

$$\mathrm{x}_i^t = \left\{ \mathrm{x}_{i1}^t, \mathrm{x}_{i2}^t, ..., \mathrm{x}_{ij}^t, ..., \mathrm{x}_{iD}^t \right\} \tag{20}$$

$$\mathrm{v}_i^t = \left\{ \mathrm{v}_{i1}^t, \mathrm{v}_{i2}^t, ..., \mathrm{v}_{ij}^t, ..., \mathrm{v}_{iD}^t \right\} \tag{21}$$

PSO algorithm can be outlined in the following steps: (i) initialize algorithm parameters, including cognitive and social acceleration constants (c_1, c_2), swarm size (N), and initial position and velocity (x_1^0, v_i^0); (ii) define the objective function to be optimized; (iii) evaluate the fitness of each particle and identify Pbest and Gbest based on fitness values; (iv) update the velocity and position of each particle using the following equations [43]:

$$\mathrm{v}_{ij}^{t+1} = K \left(\mathrm{v}_{ij}^t + c_1 r_1 \left(p_{ij}^t - \mathrm{x}_{ij}^t \right) + c_2 r_2 \left(g_{ij}^t - \mathrm{x}_{ij}^t \right) \right) \tag{22}$$

$$\mathrm{x}_{ij}^{t+1} = \mathrm{x}_{ij}^t + \mathrm{v}_{ij}^{t+1} \tag{23}$$

where constriction factor $K = 0.25 \cos\left(\frac{\pi t}{t_{max}}\right) + \frac{5}{8}$, $\mathrm{v}_{ij}^t, \mathrm{x}_{ij}^t \in \mathbb{R}$, t is the number of iterations, r_1 and r_2 are two uniformly distributed random numbers in $[0, 1]$, and p_{ij}^t, g_{ij}^t represent Pbest and Gbest for the j-th dimension; (v) recalculate the fitness at each particle's new position, update Pbest if the current fitness is superior, and update Gbest if a better Pbest is found. These steps are repeated until convergence or when the particle positions become uniform.

There have been several significant contributions to the application of Swarm Intelligence (SI) algorithms, specifically PSO in solving portfolio optimization problems [44]. For instance, [45] developed PSO algorithm to tackle MV optimization problem with cardinality constraints and compared its performance with other algorithms such as GA, Tabu Search (TS), and Simulated Annealing (SA). The results showed that no single algorithm was superior across all investment strategies; however, PSO delivered the most optimal solutions for low-risk investments. In the context of the Sharpe Ratio (SR) portfolio model, PSO outperformed GA in both long-only and long-short scenarios [46]. An extension of PSO, the Non-dominated Sorting Multi-objective Particle Swarm Optimization (NS-MOPSO) algorithm, demonstrated excellent performance in the MV portfolio model, effectively managing multiple objectives and constraints, including cardinality, minimum lot size, and asset weights limits [47], [48]. [49] extended this work by introducing risk parity constraints, while [50] used the Differential Evolution (DE) algorithm to address the same problem. Additionally, [16] successfully applied PSO to the Robust RP convex portfolio optimization problem, which focuses on managing uncertainty in the covariance matrix parameters.

PSO for the non-convex RP portfolio search for the particle position in multidimensional as the optimal solution of the problem. Every particle position represents a potential solution for a problem. Each particle starts off in multidimensional space at a random point. The position of i^{th} particle from t^{th} iteration is as follows, $\mathrm{x}_i^t = \left\{ \mathrm{x}_{i1}^t, \mathrm{x}_{i2}^t, ..., \mathrm{x}_{ij}^t, ..., \mathrm{x}_{in}^t \right\}$ where x_{ij}^t represent the weight of asset j for $j = 1, 2, ..., n$ and $\mathrm{x}_{ij}^t \in \mathbb{R}$. The optimal solution has the minimal fitness value of equation (19).

3) Ant Colony Optimization: In 1991, M. Dorigo introduced ACO algorithm, inspired by the behavior of ants searching for food, to solve combinatorial optimization prob-

TABLE I: $ACO_{\mathbb{R}}$ Solution Archives

S_1	S_1^1	S_1^2	...	S_1^n	$f(S_1)$	h_1
S_2	S_2^1	S_2^2	...	S_2^n	$f(S_2)$	h_2
...	...	...	...	...	...	...
S_l	S_l^1	S_l^2	...	S_l^n	$f(S_l)$	h_l
...	...	...	...	...	...	...
S_k	S_k^1	S_k^2	...	S_k^n	$f(S_k)$	h_k
	G^1	G^2	...	G^n		

lems [51]. ACO is a metaphor-based meta-heuristic that simulates a colony of ants as they seek the optimal path between their nest and a food source. As ants move, they deposit pheromones along their path, with more pheromones accumulating on frequently traveled routes. This pheromone trail influences other ants to follow these paths, reinforcing the optimal solution over time. The algorithm leverages this collective behavior to find the best solution by favoring paths with higher pheromone concentrations.

The $ACO_{\mathbb{R}}$ algorithm is an enhanced version of ACO designed for continuous optimization problems, in contrast to the original ACO, which is used for combinatorial problems [52]. The algorithm proceeds as follows: (i) set the algorithm parameters, including the number of ants per iteration (m), the solution archive dimensions (k), and the search locality, represented by q. As q decreases, the solution becomes more favorable and is more likely to be selected due to its increased weight. Each ant represents a solution, denoted as $S_l = \left\{S_l^1, S_l^2, ..., S_l^i, ..., S_l^n\right\}$, with each S_l^i constrained by $S_l^i \in [S_l^{min}, S_l^{max}]$. (ii) Generate k ants randomly using a uniform distribution. These solutions are then sorted in ascending order based on their fitness values, and stored in an archive with a defined structure, as shown in Table I. Better solutions are ranked lower. A weight and probability are assigned to each solution as follows:

$$h_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \tag{24}$$

$$p_l = \frac{h_l}{\sum_{r=1}^{k} h_r} \tag{25}$$

(iii) In this step, m new ants are randomly constructed based on a Gaussian kernel PDF sampled as defined below:

$$G^i(x) = \sum_{l=1}^{k} h_l g_l^i(x) = \sum_{l=1}^{k} h_l \frac{1}{\sigma_l^i\sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2{\sigma_l^i}^2}} \tag{26}$$

for $i = 1, 2, ..., n$ where $\sigma_l^i = \xi \sum_{e=1}^{k} \frac{|S_e^i - S_l^i|}{k-1}$ and $\mu^i = \left\{\mu_1^i, \mu_2^i, ..., \mu_k^i\right\} = \left\{S_1^i, S_2^i, ..., S_k^i\right\}$. Based on the value of p_l, one of the functions $g_l^i(x)$ is chosen for each new ant, and a solution for every variable is constructed using that function. (iv) Finally, the fitness values of the m newly created solutions are evaluated, and they are added to the archive in chronological order. To maintain the archive size, the m solutions with the lowest fitness values are removed.

This method has made significant contributions to the portfolio optimization problem. $ACO_{\mathbb{R}}$ method outperforms PSO in terms of performance on the MV portfolio model with cardinality constraints, particularly for low-risk investments [53]. [17] furthered the analysis by incorporating risk parity limitations. The $ACO_{\mathbb{R}}$ algorithm demonstrates excellent performance on the MV portfolio model with multi-objective functions [54], [55], [56].

The $ACO_{\mathbb{R}}$ for the non-convex RP portfolio search for an ant as the optimal solution of the problem. Every ant represents a potential solution for a problem. The representation of l^{th} ant is as follows, $S_l = \left\{S_l^1, S_l^2, ..., S_l^i, ..., S_l^n\right\}$, where S_l^i represent the weight of asset i for $i = 1, 2, ..., n$ and $S_l^i \in \mathbb{R}$. The optimal solution has the minimal fitness value of equation (19).

TABLE II: Parameters for GA

Parameter	Value
Number of population (N)	100
Crossover probability (P_X)	0.9
Mutation probability (P_Y)	0.01
Elitism	10
Maximum iterations	500

III. METHODOLOGY

This study evaluates the performance of GA, PSO, and $ACO_{\mathbb{R}}$ in optimizing the non-convex RP portfolio problem. It focuses on constructing long-only and long-short non-convex RP portfolios using both simulated and real-world datasets, across small and large-scale scenarios. The study also compares the RP portfolio's performance with that of the EW and GMV portfolios. A detailed methodology is outlined in Figure 1. All computations were coded in R and run on an HP Pavilion 14-bf0xx Laptop (Intel(R) Core(TM) i5-7200U, 2701 MHz, with 8GB RAM).

The optimal asset weights for the long-only non-convex RP portfolio are determined using GA with parameters outlined in Table II. In GA, balancing exploitation and exploration is crucial, with many implementations favoring a high crossover probability and a low mutation probability. The optimal mutation probability typically ranges from 0.005 to 0.01, while the crossover probability is ideally between 0.75 and 0.95 [57], [58]. In this study, the parameters were selected through experimentation: the crossover probability is set to 0.9 to promote exploration, while a lower mutation probability of 0.01 is used to refine existing solutions and prevent excessive exploitation.

Using the same parameters as before, GA is also employed to determine the optimal weights for the non-convex RP long-short portfolio. The maximum limit on the asset weight allocated to short positions varies among investors, making it challenging to set a standard percentage. Short positions are generally riskier than long positions due to the potential for unlimited losses if the shorted asset's price rises significantly [59]. As a result, financial advisors often recommend limiting the proportion of a portfolio allocated to short positions, typically capping it at 0% to 5%, or up to 10% for more risk-tolerant investors. In this study, the lower and upper bounds for each stock weights are set at -0.3 and 1, respectively.

The optimal weights of the long-only non-convex RP portfolio are calculated using PSO with R, using the parameters specified in Table III. In this study, the cognitive component (c_1) and social component (c_2) are both set to 2.05, with a higher cognitive component promoting individual exploration based on personal experiences, and a larger social component encouraging particles to follow the group's optimal solution, thus enhancing resource exploitation [60]. These parameter

TABLE III: Parameters for PSO

Parameter	Value
Number of particle (N)	100
c_1	2.05
c_2	2.05
Maximum iterations	500

TABLE IV: Parameters for $ACO_{\mathbb{R}}$

Parameter	Value
Number of ant (N)	100
q	0.01
k	10
ξ	0.0001
Maximum iterations	500

values are determined through experimentation to effectively balance exploration and exploitation. To determine the optimal weights for the long-short non-convex RP portfolio, the PSO algorithm is applied using the same parameters as those in Table III. The lower and upper weight limits for each stock are the same as those used in GA, specifically -0.3 and 1.

Using `R` and the parameters of $ACO_{\mathbb{R}}$ in Table IV , the optimal weights of the long-only non-convex RP portfolio are determined. The parameter values are chosen through trial and error to balance exploration and exploitation. The q value is set to 0.01 to promote more exploration, as higher q values broaden the Gaussian kernel. Meanwhile, the ξ parameter is set to a very low value of 0.0001, focusing on local search and exploitation by reducing the standard deviation for perturbation. The same parameters are applied to calculate the optimal asset weights for the non-convex RP long-short portfolio. The lower and upper weight limits for each stock are set at -0.3 and 1, respectively, consistent with the GA and PSO settings.

For the real-world datasets, the first dataset consists of daily returns of the adjusted closing prices for 30 companies listed on the Jakarta Islamic Index (JII) over the period from January 5, 2022, to January 5, 2024, obtained from Yahoo! Finance using `quantmod` library [61]. Several stocks were excluded due to data issues.

To assess the scalability of the algorithms, a second dataset is created using daily adjusted closing prices of 80 companies listed on the Indonesia Stock Exchange (IDX). The covariance matrix is calculated using historical stock price data from the same time-frame. Here, several stocks were also excluded due to data issues.

The JII and IDX stock data, which focus on large-cap stocks, are used because the data is readily available, there is high stock liquidity in the market (facilitating portfolio adjustments for investors), and the proportion of JII and IDX stocks relative to total market capitalization is relatively high, increasing the likelihood that the research findings will be applicable to a broad market segment.

As in the real-world datasets, two different dimensionality configurations are considered in the simulation setup. Asset returns are generated using a multivariate normal distribution with a mean of 0, and the covariance matrix is the sample covariance matrix of daily returns from the constituents of the JII and IDX indices for the period from January 5, 2022, to January 5, 2024. For small and large scalability scenarios, a noise factor is added to each asset, distributed as a univariate normal with a variance of 0.1 times the variance of each asset.

To evaluate how well the meta-heuristic methods optimize the non-convex RP portfolio, two performance metrics are used: Mean Squared Error (MSE) and the Herfindahl-Hirschman Index (HHI). MSE calculates the mean of squared differences between the actual and the target value of the RRC of each asset. For a portfolio containing n assets, the target RRC for each asset is $1/n$. In contrast, the HHI measures the concentration of risk within the portfolio and is defined as follows [6], [49],

$$h(\mathrm{w}) = \sum_{i=1}^{n} \left(\frac{\mathrm{w}_i(\mathbf{\Sigma w})_i}{\mathbf{w^T \Sigma w}} \right)^2 \tag{27}$$

If the index value of a portfolio consisting of n assets is 1, it indicates that the portfolio is highly concentrated in a single asset. In contrast, when each asset contributes equally to the portfolio's risk, the index value equals $1/n$.

RP portfolio performance is analyzed using a rolling-window method, which updates the covariance matrix. The walk-forward analysis involves calculating portfolio weights from a training data subset and then evaluating performance using a distinct evaluation data subset. This process consists of sequential walk-forwards applied to various data subsets, ensuring non-overlapping evaluation periods.

In order to evaluate portfolio performance and flexibility in response to market fluctuations, two 240-day (12*20) look-back window analyses are performed. These analyses include re-balancing periods of 60 days (3*20) and 120 days (6*20), allowing for more frequent adjustments.

For each day t, portfolio weights are computed using price data from $t - 239$ to t, with trades executed on day $t + 1$. Returns are accumulated over the subsequent 121 days for the 120-day re-balancing period and 61 days for the 60-day period.

The portfolio performance is analyzed using the `portfolioBacktest` library and several performance metrics, including the Sharpe Ratio (SR), Maximum Drawdown (MD), Sortino Ratio, Sterling Ratio, Value at Risk (VaR), and Turnover (TO) Ratio [62].

IV. Result and Discussion

A. Algorithmic Performance Analysis

Figure 2 illustrates the fitness value convergence for the long-only and long-short non-convex RP portfolios using GA on both real and simulated datasets. GA-LO-Real and GA-LO-Sim refer to GA for long-only non-convex RP portfolios on real and simulated datasets, respectively.

The performance of the GA on both real and simulated datasets for long-only and long-short non-convex RP portfolios is illustrated by the convergence of the fitness values in Figure 2. GA-LO-Real-JII and GA-LO-Sim-JII refer to GA for long-only non-convex RP portfolios on real and simulated JII datasets, respectively. Similarly, GA-LS-Real-JII and GA-LS-Sim-JII refer to GA for long-short non-convex RP portfolios on real and simulated JII datasets, respectively. The fitness values fluctuate and improve over time, indicating that the algorithm is effectively exploring the search space and making progress toward better solutions. The convergence curves show that the long-short RP portfolios stabilize more quickly than the long-only portfolios, which require

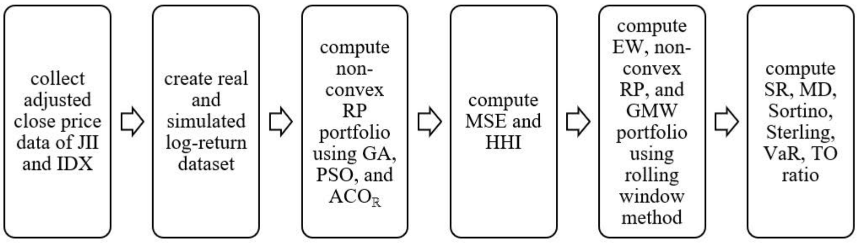


Fig. 1: Flow chart of research steps

more iterations to reach convergence. This difference can be attributed to the more restrictive search space of the long-only portfolios, which permit only positive weights, thereby limiting the ability to adjust risk contributions. In contrast, the long-short portfolios, which allow both positive and negative weights, offer greater flexibility, enabling the GA to explore potential solutions more efficiently and achieve risk parity in fewer generations. The expanded search space of the long-short portfolios accelerates convergence by providing more opportunities to balance risk across assets.

Figure 3 shows the RRC and asset weights for the long-only and long-short non-convex RP portfolios using GA, PSO, and $ACO_{\mathbb{R}}$ on real JII datasets. For real IDX datasets, these results are presented in Figure 4. For simulated JII datasets, see Figure 5, and for simulated IDX datasets, refer to Figure 6. Several conclusions can be drawn from the figures depicting RRC and asset weights. The PSO algorithm produces asset weights that result in nearly equal RRC across all assets, regardless of whether the dataset is small or large. Although the long-short RP portfolio allows for negative weights, the PSO algorithm identifies all optimal asset weights as positive. In contrast, the asset weights generated by the $ACO_{\mathbb{R}}$ and GA algorithms provide varying RRC across different assets. Additionally, the asset weights for the long-only and long-short RP portfolios exhibit significant differences. Unlike the PSO algorithm, both the GA and $ACO_{\mathbb{R}}$ algorithms produce optimal asset weights for the long-short RP portfolio that include negative weights.

Table V shows the MSE and HHI for the non-convex long-only and long-short RP portfolio using GA, PSO and $ACO_{\mathbb{R}}$. The algorithm is deemed effective if it assigns asset weights such that each asset contributes equally to the overall risk. For the non-convex RP portfolio, the target RRC is $1/27 = 0.0370370$ for the small-scale datasets and $1/75 = 0.0133333$ for the large-scale datasets. As shown in Table V, PSO consistently outperforms GA and $ACO_{\mathbb{R}}$ across all datasets. PSO closely matches the target risk parity with minimal deviation, demonstrating a good balance between exploration and exploitation. It achieves the lowest MSE and HHI, with slight deviations from the ideal risk contribution target in both small and large-scale datasets. PSO outperforms other algorithms, especially in long-only portfolios, for both large-scale and small-scale datasets.

In contrast to PSO, the performance of the $ACO_{\mathbb{R}}$ algorithm remains stable across all datasets. It performs marginally better for the long-only RP portfolio than for the long-short RP portfolio in all cases. On the other hand, GA shows inconsistency in its performance for both non-convex long-only and long-short RP portfolios. GA performs slightly better for long-only portfolios in large-scale datasets but underperforms in small-scale datasets when compared to long-short portfolios.

B. Portfolio Performance Analysis

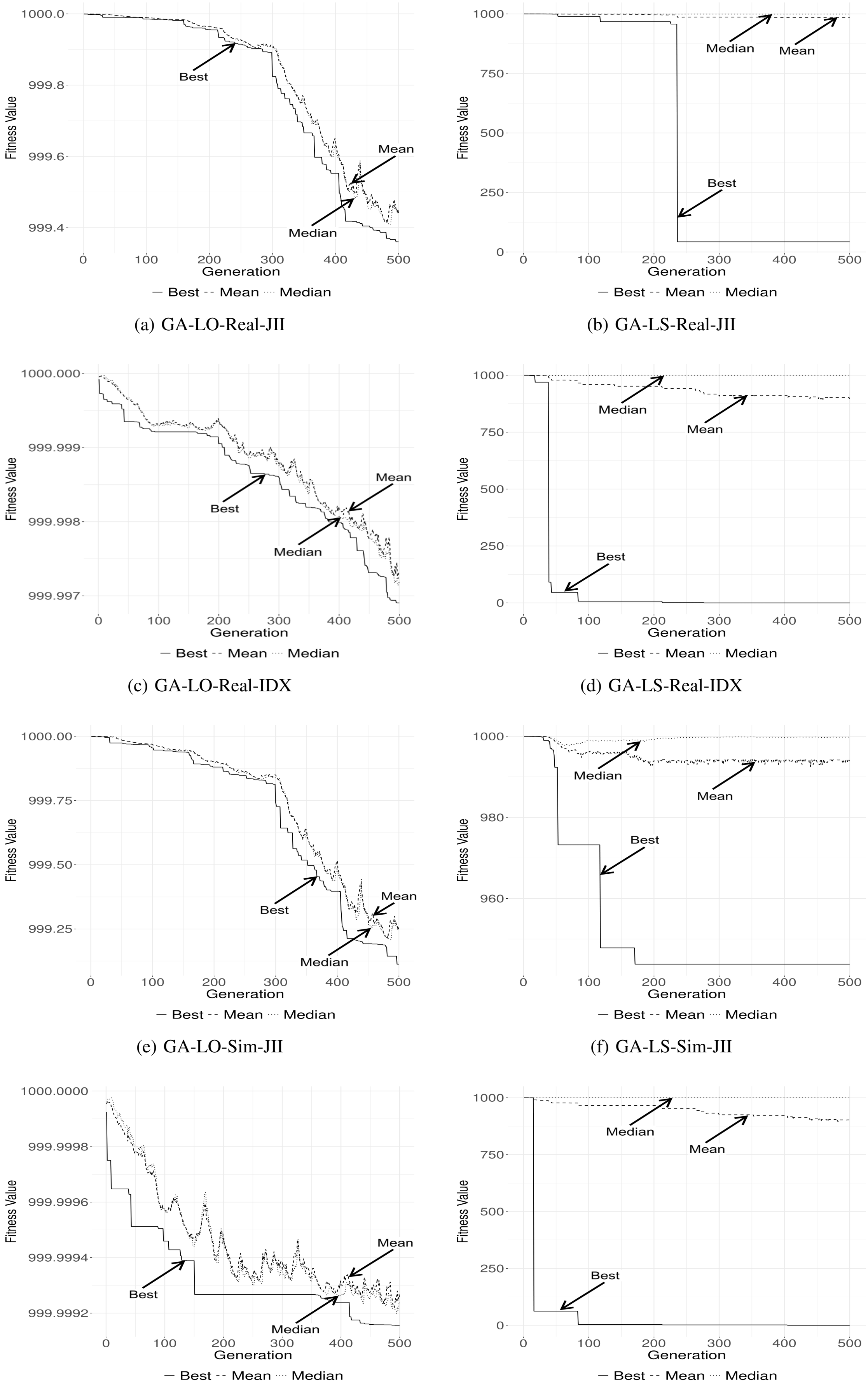
In comparison to the EW and GMV portfolios, the metric values in Tables VI and VII provide a comprehensive assessment of the RP portfolio's resilience in risk management. Although not all criteria support its superiority, the RP portfolio performs better than other portfolio types. The RP portfolio continuously displays the lowest VaR and MD ratios across four datasets, suggesting a lower risk of suffering significant losses. This suggests the RP portfolio is a more conservative investment strategy, particularly resilient to market shocks and volatility. By balancing risk across assets, the RP portfolio limits potential losses, ensuring stable returns, especially in volatile markets, making it ideal for risk-averse investors.

The RP portfolio seeks to evenly allocate risk across all assets, preventing any single asset from being overly exposed to large losses. Its lower MD ratio demonstrates an ability to minimize downturn severity, effectively reducing losses during prolonged market declines. This balanced risk approach ensures more stable returns, particularly in volatile markets, making it an attractive choice for risk-averse investors.

The SR and Sterling Ratio of the RP portfolio are comparable to, and in some cases surpass, the best values observed across other portfolios, particularly in the rolling-window analysis with re-balancing period of 6*20-day period on real IDX datasets. This suggests that the RP portfolio demonstrates superior risk-adjusted performance. Specifically, the SR indicates that the RP portfolio achieves higher returns per unit of risk, while the Sterling ratio underscores its effectiveness in minimizing drawdowns and managing downside risk, which contributes to its overall stability. The RP portfolio consistently outperforms both the Equal-Weighted (EW) and Global Minimum Volatility (GMV) portfolios, which adopt a more balanced approach to profitability and risk management.

V. Conclusion

This study addresses the optimization of the non-convex RP portfolio, which may have multiple local minima. While non-convex problems are often transformed into convex ones, this can introduce challenges such as infeasibility, loss of original structure, or increased computational complexity. The study presents computational results from three meta-heuristic algorithms—GA, PSO, and $ACO_{\mathbb{R}}$—applied to the non-convex RP portfolio. Although PSO shows some inconsistency in performance for both long-only and long-short



(a) GA-LO-Real-JII

(b) GA-LS-Real-JII

(c) GA-LO-Real-IDX

(d) GA-LS-Real-IDX

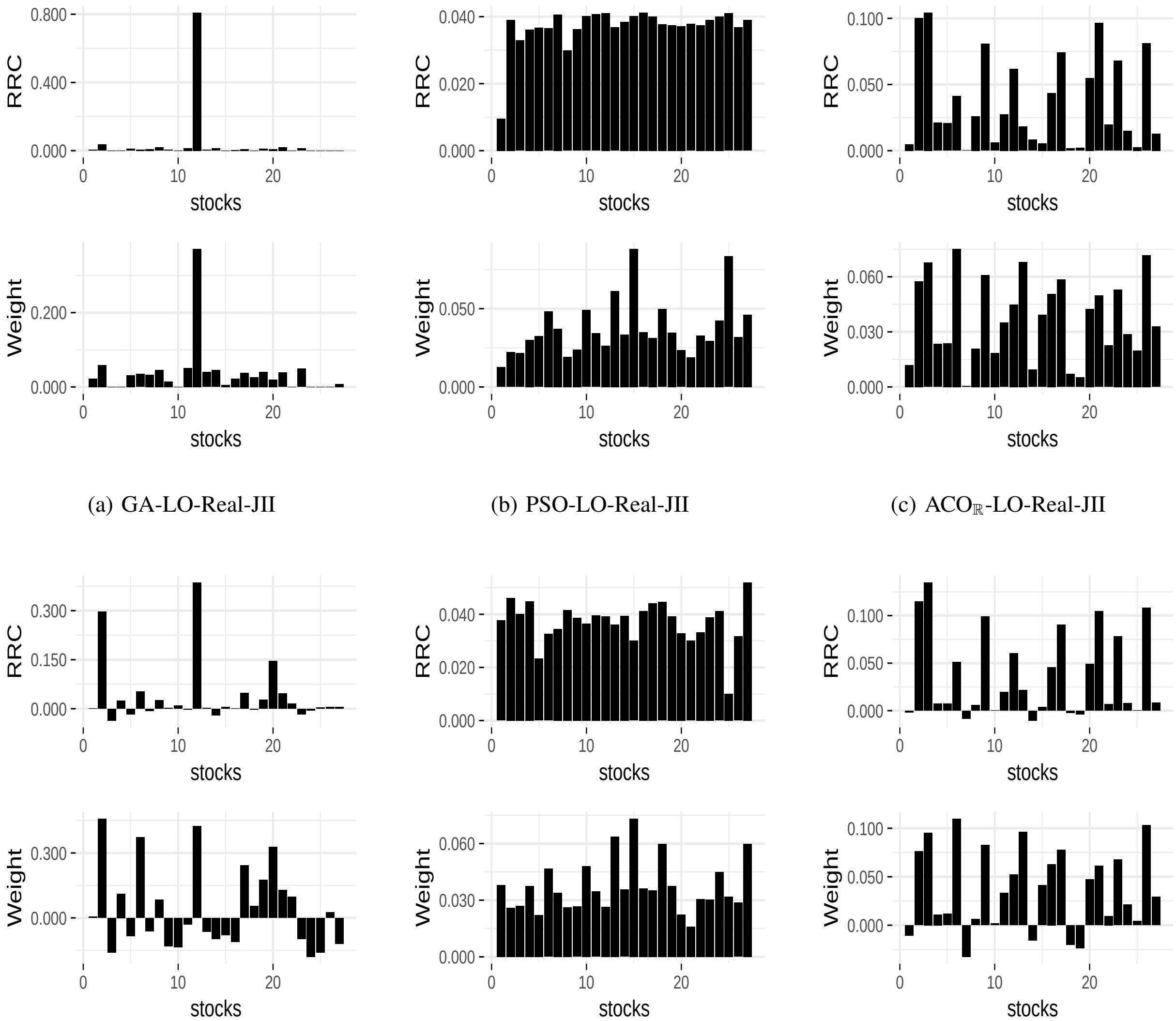
(e) GA-LO-Sim-JII

(f) GA-LS-Sim-JII

(g) GA-LO-Sim-IDX

(h) GA-LS-Sim-IDX

Fig. 2: Convergence plot of fitness value of the RP portfolio with GA



(a) GA-LO-Real-JII (b) PSO-LO-Real-JII (c) $ACO_{\mathbb{R}}$-LO-Real-JII

(d) GA-LS-Real-JII (e) PSO-LS-Real-JII (f) $ACO_{\mathbb{R}}$-LS-Real-JII

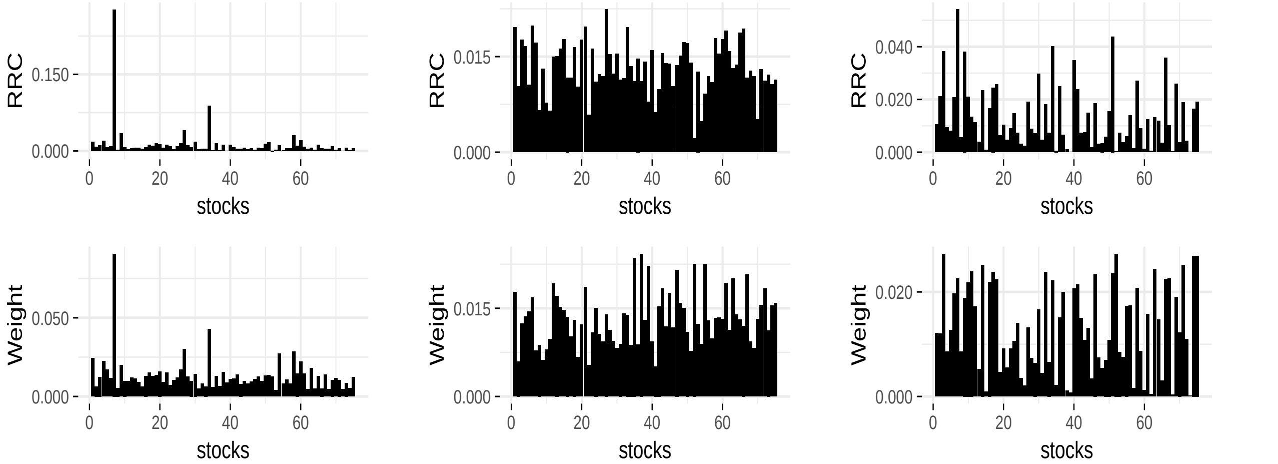
Fig. 3: Relative risk contribution (RRC) and asset weight of the RP portfolio with GA, PSO, and $ACO_{\mathbb{R}}$ on real JII dataset

TABLE V: Mean squared error (MSE) and herfindahl-hirschman index (HHI) of the relative risk contribution (RRC)

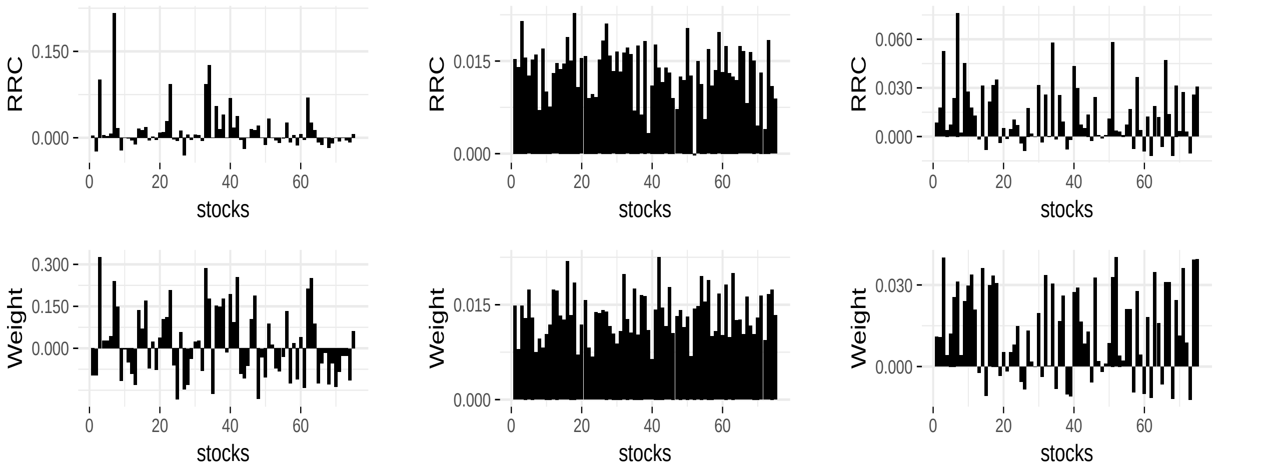
	Real-JII		Real-IDX		Sim-JII		Sim-IDX	
	MSE	HHI	MSE	HHI	MSE	HHI	MSE	HHI
GA-LO	0.0230228	0.6586534	0.0010770	0.0941099	0.0239136	0.6827047	0.0006084	0.0589633
GA-LS	0.0086636	0.2709551	0.0013805	0.1168685	0.0081813	0.2579323	0.0004003	0.0433555
PSO-LO	**0.0000353**	**0.0379901**	**0.0000162**	**0.0145483**	**0.0000607**	**0.0386755**	**0.0000121**	**0.0142423**
PSO-LS	0.0000620	0.0387121	0.0000200	0.0148303	0.0000685	0.0388861	0.0000175	0.0146484
$ACO_{\mathbb{R}}$-LO	0.0011311	0.0675754	0.0001399	0.0238283	0.0012084	0.0696649	0.0001383	0.0237083
$ACO_{\mathbb{R}}$-LS	0.0019742	0.0903416	0.0003404	0.0388655	0.0019310	0.0891751	0.0003405	0.0388720

TABLE VI: Portfolio performance with re-balancing period of 3*20 days

	Real-JII			Real-IDX			Sim-JII			Sim-IDX		
	EW	RP	GMV	EW	RP	GMV	EW	RP	GMV	EW	RP	GMV
SR	**0.4960**	0.3973	-0.0577	**0.4537**	0.4520	-0.1039	**0.8768**	0.8723	-0.3340	**1.3088**	1.2858	1.0936
MD	0.1419	**0.1253**	0.3198	0.1180	**0.1175**	0.2355	0.1239	**0.1047**	0.5983	**0.0823**	0.0858	0.1355
Sortino Ratio	**0.6768**	0.5452	-0.0837	**0.6305**	0.6237	-0.1466	**1.3132**	1.2953	-0.4585	**1.9146**	1.8759	1.5812
Sterling Ratio	**0.4988**	0.4036	-0.0659	**0.4981**	0.4890	-0.0977	1.1376	**1.1742**	-0.3600	**2.0451**	1.8236	1.5770
VaR (0.95)	0.0156	**0.0122**	0.0344	0.0133	**0.0132**	0.0222	0.0152	**0.0141**	0.0663	0.0132	**0.0125**	0.0190
TO	**0.0016**	0.0026	0.0105	**0.0017**	0.0050	0.0123	**0.0019**	0.0025	0.0082	**0.0019**	0.0040	0.0112



(a) GA-LO-Real-IDX (b) PSO-LO-Real-IDX (c) $ACO_{\mathbb{R}}$-LO-Real-IDX



(d) GA-LS-Real-IDX (e) PSO-LS-Real-IDX (f) $ACO_{\mathbb{R}}$-LS-Real-IDX

Fig. 4: Relative risk contribution (RRC) and asset weight of the RP portfolio with GA, PSO, and $ACO_{\mathbb{R}}$ on real IDX dataset

TABLE VII: Portfolio performance with re-balancing period of 6*20 days

	Real-JII			Real-IDX			Sim-JII			Sim-IDX		
	EW	RP	GMV	EW	RP	GMV	EW	RP	GMV	EW	RP	GMV
SR	**0.4563**	0.3138	0.4639	0.3589	**0.5268**	-0.0671	**0.8538**	0.8047	-0.5021	**1.2157**	1.1940	0.8035
MD	**0.1350**	0.1371	0.2988	0.1153	**0.1139**	0.2284	0.1227	**0.1010**	0.5735	**0.0863**	0.0893	0.1447
Sortino Ratio	0.6241	0.4297	**0.6718**	0.4953	**0.7291**	-0.0950	**1.2806**	1.1960	-0.6865	**1.7730**	1.7444	1.1400
Sterling Ratio	**0.6718**	0.2953	0.5632	0.4050	**0.5906**	-0.0636	1.1115	**1.1287**	-0.5219	**1.8156**	1.6575	1.0492
VaR (0.95)	0.0149	**0.0119**	0.0340	0.0138	**0.0134**	0.0217	0.0153	**0.0140**	0.0625	0.0132	**0.0126**	0.0189
TO	**0.0007**	0.0012	0.0046	**0.0007**	0.0018	0.0072	**0.0009**	0.0012	0.0060	**0.0009**	0.0020	0.0054

portfolios, it consistently outperforms GA and $ACO_{\mathbb{R}}$ across all datasets. A rolling-window analysis demonstrates that the RP portfolio outperforms both the EW and GMV portfolios, achieving the lowest VaR and MD ratios, though not all metrics support this. These findings indicate that the proposed method is effective and offers a promising alternative for solving non-convex RP portfolio optimization. However, reliance on historical data may not fully reflect dynamic market conditions, and performance could vary in different scenarios. Future research should address these limitations by testing the algorithm on more complex problems with diverse objectives and real-world constraints, and comparing it with other hybrid or meta-heuristic methods.

REFERENCES

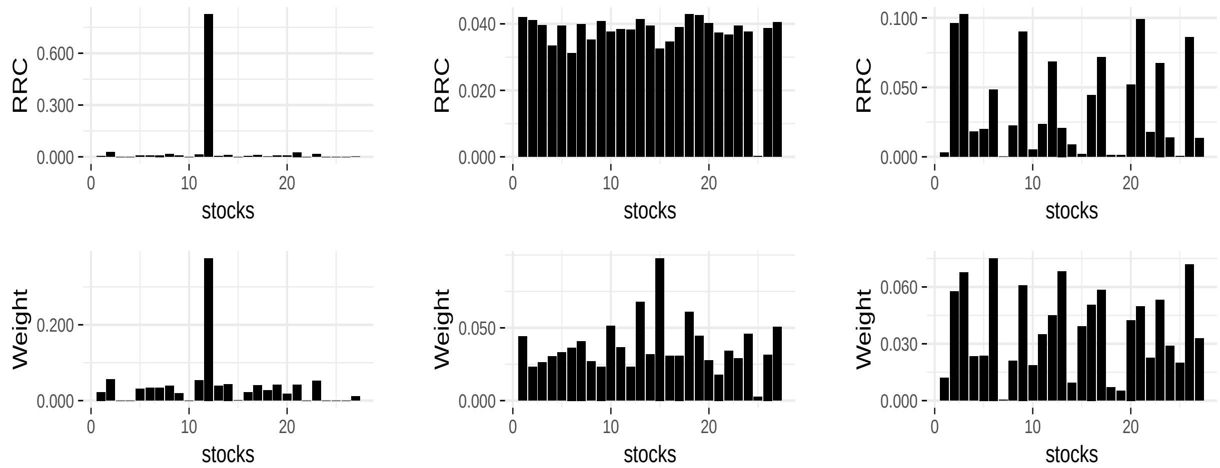
[1] W. H. Wagner, and S. C. Lau, "The Effect of Diversification on Risk," *Financial Analysts Journal*, vol. 27, no. 66, pp. 48–53. 1971.

[2] H. Markowitz, "Portfolio Selection." *The Journal of Finance*, vol. 7, pp. 77–91. 1952.

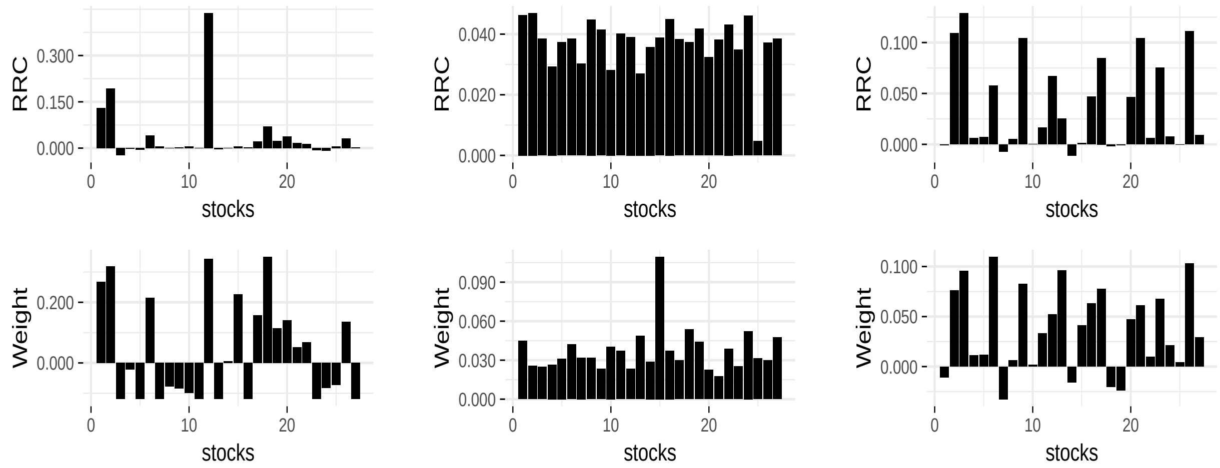
[3] J. L. Campbell, "The US financial crisis: Lessons for Theories of Institutional Complementarity," *Socio-Economic Review*, vol. 9, no. 2, pp. 211-234. 2011.

[4] Y. Choueifaty, and Y. Coignard, "Toward Maximum Diversification," *Journal of Portfolio Management*, vol. 35, no. 1, pp. 40-51. 2008.

[5] S. Maillard, T. Roncalli, and J. Teiletche, "On the Properties of Equally-Weighted Risk Contributions Portfolios," *The Journal of Portfolio Management*, vol. 36, pp. 60–70. 2010.



(a) GA-LO-Sim-JII (b) PSO-LO-Sim-JII (c) ACO$_{\mathbb{R}}$-LO-Sim-JII



(d) GA-LS-Sim-JII (e) PSO-LS-Sim-JII (f) ACO$_{\mathbb{R}}$-LS-Sim-JII

Fig. 5: Relative risk contribution (RRC) and asset weight of the RP portfolio with GA, PSO, and ACO$_{\mathbb{R}}$ on simulated JII dataset

[6] T. Roncalli, *Introduction to Risk Parity and Budgeting*, Chapman & Hall / CRC Financial Mathematics Series, 2016.

[7] E. Qian, "On the Financial Interpretation of Risk Contribution: Risk Budgets Do Add Up," *SSRN Electronic Journal*, vol. 4, no. 4. 2006.

[8] B. Bruder, N. Kostyuchyk, and T. Roncalli, "Risk Parity Portfolios with Skewness Risk: An Application to Factor Investing and Alternative Risk Premia," *SSRN Electronic Journal*, pp. 1–48. 2016.

[9] J. Nocedal, S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[10] H. Mausser, O. Romanko, "Computing equal risk contribution portfolios," *IBM Journal of Research and Development*, vol. 58, no. 4. 2014.

[11] X. Bai, K. Scheinberg, and R. Tutuncu, "Least-Squares Approach to Risk Parity in Portfolio Selection." *SSRN Electronic Journal*, pp. 1–27. 2013.

[12] Y. Feng, D. P. Palomar, "Successive Convex Optimization Methods for Risk Parity Portfolio Design," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5285–5300. 2015.

[13] Y. Chen, C. Li, Y. Hu, Z. Lu, "Parallel Non-Convex Approximation Framework For Risk Parity Portfolio Design," *Parallel Computing*, vol. 116. 2023.

[14] R. Hochreiter, "An Evolutionary Optimization Approach to Risk Parity Portfolio Selection," *Lecture Notes in Computer Science*, vol. 9028, pp. 279–288. 2015.

[15] G. Filograsso, G. di. Tollo, "Adaptive Evolutionary Algorithms for Portfolio Selection Problems," *Computational Management Science*, vol. 20, no. 1, pp. 1-38. 2023.

[16] S. M. E. Mirmohammadi, M. Madanchi zaj, H. Panahian, H. Jabbary, "Stock Portfolio Optimization Using a Combined Approach of Relative Robust Risk Parity," *Iranian Journal of Finance*, vol. 5, no. 4, pp. 87–106. 2021.

[17] V. G. A. Pai, "Risk Budgeted Portfolio Optimization Using an Extended Ant Colony Optimization Metaheuristic," *International Journal of Applied Metaheuristic Computing*, vol. 3, no. 4, pp. 25–42. 2010.

[18] E. H. Abdelhameed, S. Abdelraheem, Y. S. Mohamed, A. A. Z. Diab, "Effective hybrid search technique based constraint mixed-integer programming for smart home residential load scheduling," *Scientific Reports*, vol. 13, no. 1, pp. 1–23. 2023.

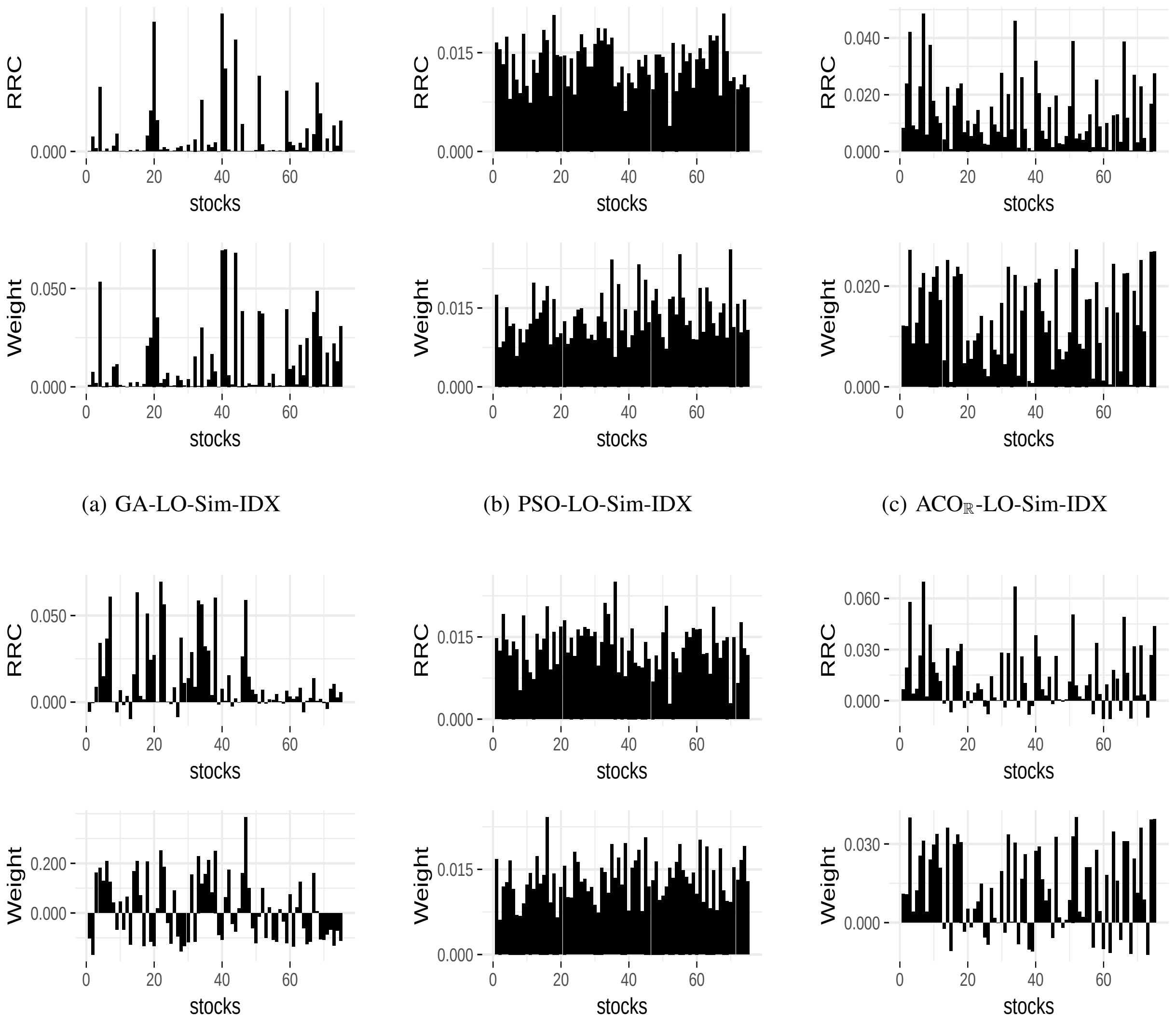
[19] H. T. Anis, G. Costa, G., R. H. Kwon, "Risk-allocation-based index tracking," *Computers and Operations Research*, vol. 154, pp. 1–20. 2023.

[20] M. Cavus, A. Allahham, "Enhanced Microgrid Control through Genetic Predictive Control: Integrating Genetic Algorithms with Model Predictive Control for Improved Non-Linearity and Non-Convexity Handling," *Energies*, vol. 17, no. 17. 2024.

[21] N. Nalini, C. Kumar, P. Vijayarajan, N. Chidambararaj, "Integrated approach for dynamic economic/emission dispatch problem: multi-objective moth flame optimizer with transmission loss prediction using cascaded forward neural network," *Electrical Engineering*, vol. 106, no.3, pp. 3495–3528. 2024.

[22] Y. Qiao, Y. Niu, Z. Han, L. Xiong, N. Wang, T. Q. S. Quek, B. Ai, "Resource Allocation for ISAC and HRLLC in UAV-Assisted HSR System With a Hybrid PSO-Genetic Algorithm," *IEEE Internet of Things Journal*, pp. 1–16. 2024.

[23] M. Stefanoni, P. Sarcevic, J. Sárosi, A. Odry, "Optimization Techniques



(a) GA-LO-Sim-IDX (b) PSO-LO-Sim-IDX (c) $ACO_{\mathbb{R}}$-LO-Sim-IDX

(d) GA-LS-Sim-IDX (e) PSO-LS-Sim-IDX (f) $ACO_{\mathbb{R}}$-LS-Sim-IDX

Fig. 6: Relative risk contribution (RRC) and asset weight of the RP portfolio with GA, PSO, and $ACO_{\mathbb{R}}$ on simulated IDX dataset

in the Localization Problem: A Survey on Recent Advances,“ *Machines*, vol. 12, no. 8, pp. 1–51. 2024.

[24] F. Wang, J. Chen, H. Huang, S. Xu, “A particle dynamical system algorithm to find the sparse linear complementary solutions,“ *Applied Soft Computing*, vol. 166. 2024.

[25] J. Zhang, T. Sun, F. Liu, Y. Liu, L. Ye, “Dynamic output feedback control for UAVs with limited network bandwidth: A GA-assisted design approach,“ *Journal of the Franklin Institute*, vol. 361, no. 2, pp. 598–611. 2024.

[26] A. B. Mohammed, A. F. Laila, and K. S. Arun, “Chapter 10 Metaheuristic Algorithms: A Comprehensive Review,“ *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, pp. 185-231. Elsevier Inc. 2018.

[27] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. The MIT Press. 1992.

[28] S. Arnone, A. Loraschi, and A. Tettamanzi, “A Genetic Approach to Portfolio Selection,“ *Neural Network World*, vol. 3, no. 6, pp. 597–604. 1992.

[29] T. J. Chang, S. C. Yang, and K. J. Chang, “Portfolio Optimization Problems in Different Risk Measures using Genetic Algorithm,“ *Expert Systems with Applications*, vol. 36, no. 7, pp. 10529–10537. 2009.

[30] H. Soleimani, H. R. Golmakani, and M. H. Salimi, “Markowitz-based Portfolio Selection with Minimum Transaction Lots, Cardinality Constraints and Regarding Sector Capitalization using Genetic Algorithm,“ *Expert Systems with Applications*, vol. 36, no. 3, pp. 5058–5063. 2009.

[31] R. Fatoni, R. Kusumawati, “Optimal Portfolio Analysis of Fuzzy Mean Absolute Deviation using Genetics Algorithm,“ *Jurnal Sains Dasar*, vol. 11, no. 1, pp. 23–29. 2022.

[32] M. B. Kar, S. Kar, S. Guo, X. Li, and S. Majumder, “A New Bi-Objective Fuzzy Portfolio Selection Model and Its Solution through Evolutionary Algorithms,“ *Soft Computing*, vol. 23, no. 12, pp. 4367–4381. 2019.

[33] S. K. Mishra, G. Panda, S. Meher, and R. Majhi, “Constraint Robust Portfolio Selection By Multiobjective Evolutionary Genetic Algorithm,“ *International Journal of Electronics Signals and Systems*, vol. 2, no. 1, pp. 57–62. 2012.

[34] D. Rosadi, E. P. Setiawan, M. Templ, P. Filzmoser, “Robust Covariance Estimators for Mean-Variance Portfolio Optimization with Transaction Lots,“ *Operations Research Perspectives*, vol. 7, pp. 1-11. 2020.

[35] K. P. Anagnostopoulos, G. Mamanis, “A Portfolio Optimization Model with Three Objectives and Discrete Variables,“ *Computers and Operations Research*, vol. 37, no. 7, pp. 1285–1297. 2010.

[36] K. T. Lwin, R. Qu, and B. L. MacCarthy, “Mean-VaR Portfolio Optimization: A Nonparametric Approach,“ *European Journal of Operational Research*, vol. 260, no. 2, pp. 751–766. 2017.

[37] M. K. Mehlawat, P. Gupta, and A. Z. Khan, “Portfolio Optimization using Higher Moments in An Uncertain Random Environment,“ *Information Sciences*, vol. 567, pp. 348–374. 2021.

[38] W. Chen, Y. Wang, P. Gupta, M. K. Mehlawat, “A Novel Hybrid Heuristic Algorithm for A New Uncertain Mean-Variance-Skewness Portfolio Selection Model with Real Constraints,“ *Applied Intelligence*, vol. 48, no. 9, pp. 2996–3018. 2018.

[39] A. Rangel-Merino, J. L. López-Bonilla, and R. Linares Y Miranda,

"Optimization Method based on Genetic Algorithms," *Apeiron*, vol. 12, no. 4, pp. 393–408. 2005.
[40] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. Springer-Verlag, 1994.
[41] Z. Michalewicz, and C. Z. Janikow, "Handling Linear Constraints in Genetic Algorithms," *Proceedings of the Fourth International Conference on Genetic Algorithms*. 1992.
[42] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization," *Proceedings of the International Conference on Neural Networks*, pp. 1942–1948. 1995.
[43] Y. Lu, M. Liang, Z. Ye, and L. Cao, "Improved particle swarm optimization algorithm and its application in text feature selection," *Applied Soft Computing Journal*, vol. 35, pp. 629–636. 2015.
[44] Abdurakhman, M. N. H. T. Putra, and E. Purnaningrum, "Clustering Performance Analysis in Portfolio Optimization based on the Swarm Intelligence," *International Journal of Computer Science*, vol. 51, no. 12, pp. 1943–1949. 2024.
[45] T. Cura, "Particle swarm optimization approach to portfolio optimization," *Nonlinear Analysis: Real World Applications*, vol. 10, no. 4, pp. 2396–2406. 2009.
[46] H. Zhu, Y. Wang, K. Wang, Y. Chen, "Particle Swarm Optimization (PSO) for The Constrained Portfolio Optimization Problem," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10161–10169. 2011.
[47] S. K. Mishra, G. Panda, R. Majhi, "A Comparative Performance Assessment of A Set of Multiobjective Algorithms for Constrained Portfolio Assets Selection," *Swarm and Evolutionary Computation*, vol. 16, pp. 38–51. 2014
[48] Y. L. T. V. Silva, A. B. Herthel, A. Subramanian, "A Multi-Objective Evolutionary Algorithm for A Class of Mean-Variance Portfolio Selection Problems," *Expert Systems with Applications*, vol. 133, pp. 225–241. 2019
[49] M. Kaucic, "Equity Portfolio Management with Cardinality Constraints and Risk Parity Control using Multi-Objective Particle Swarm Optimization," *Computers and Operations Research*, vol. 109, pp. 300–316. 2019.
[50] N. S. K. Gandikota, M. H. Hasan, J. Jaafar, "An Adaptive Metaheuristic Approach for Risk-Budgeted Portfolio Optimization," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 1, pp. 305–314. 2023.
[51] M. Dorigo, T. Stützle, *Ant Colony Optimization*, The MIT Press Cambridge, 2004.
[52] K. Socha, and M. Dorigo, "Ant Colony Optimization for Continuous Domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173. 2008.
[53] G. F. Deng, W. T. Lin, "Ant Colony Optimization for Markowitz Mean-Variance Portfolio Model," *Lecture Notes in Computer Science*, pp. 238–245. 2010.
[54] M. Khalidji, M. Zeiaee, A. Taei, M. R. Jahed-Motlagh, H. Khaloozadeh, "Dynamically Weighted Continuous Ant Colony Optimization for Biobjective Portfolio Selection using Value-at-Risk," *Proc. 3rd Asia Int. Conf. Model. Simulation*, pp. 230–235. 2009.
[55] W. Ting, Y. Xia, "The Study of Model for Portfolio Investment based on Ant Colony Algorithm," *International Conference on Future Computer and Communication* , pp. 238–240. 2009.
[56] K. F. Haqiqi, T. Kazemi, "Ant Colony Optimization Approach to Portfolio Optimization – A Lingo Companion," *International Journal of Trade, Economics and Finance*, vol. 3, no. 2, pp. 148–153. 2012.
[57] J. J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms," *IEEE Transaction on System Man, and Cybernetics*, vol. SMC-16, no. 1, pp. 122–128. 1986.
[58] V. P. Patil, D. D. Pawar, "The Optimal Crossover or Mutation Rates in Genetic Algorithm: A Review," *International Journal of Applied Engineering and Technology*, vol. 5, no. 3, pp. 38–41. 2015.
[59] J. H. Kim, W. C. Kim, F. J. Fabozzi, "Portfolio Selection with Conservative Short-Selling," *Finance Research Letters*, vol. 18, pp. 363–369. 2016.
[60] M. N. Alam, B. Das, and V. Pant, "A Comparative Study of Metaheuristic Optimization Approaches for Directional Overcurrent Relays Coordination," *Electric Power Systems Research*, vol. 128, pp. 39–52. 2015.
[61] J. A. Ryan, J. M. Ulrich, E. B. Smith, W. Thielen, P. Teetor, S. Bronder, "quantmod: Quantitative Financial Modelling Framework," *R package version 0.4.24.*. 2023.
[62] D. P. Palomar and R. Zhou,"portfolioBacktest: Automated Backtesting of Portfolios over Multiple Datasets," *R package version 0.4.1.*. 2022.