Quantum-inspired meta-heuristic approaches for a constrained portfolio optimization problem

Abhishek Gunjan

Christ University

Siddhartha Bhattacharyya

dr.siddhartha.bhattacharyya@gmail.com

Rajnagar Mahavidyalaya

Research Article

Keywords: Portfolio Optimization, Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Quantum-inspired meta-heuristics

Posted Date: September 25th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3366903/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Evolutionary Intelligence on March 25th, 2024. See the published version at https://doi.org/10.1007/s12065-024-00929-4.

Quantum-inspired meta-heuristic approaches for a constrained portfolio optimization problem

Abhishek Gunjan and Siddhartha Bhattacharyya

Abstract Portfolio optimization has always been a challenging proposition and a highly studied problem in finance and management. Portfolio optimization facilitates the selection of the right assets and their distribution according to the set objectives. Often, it has been found that this nonlinear constraint problem cannot be efficiently solved using a traditional approach. In this paper, quantum-inspired incarnations of three evolutionary techniques, viz., (i) genetic algorithm (GA), (ii) differential evolution (DE), and (iii) particle swarm optimization (PSO) are used for the portfolio optimization problem. Experiments have been conducted with more than 10 years of stock price data from NASDAQ, BSE, and Dow Jones. Several enhancements of the evolutionary algorithms have been proposed in this article, viz., (i) enhanced crossover techniques for the portfolio optimization problem, (ii) regularization function to allocate funds efficiently, and (iii) dynamic parameter tuning using sensitivity analysis.

Key words: Portfolio Optimization, Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Quantum-inspired meta-heuristics

1 Introduction

A portfolio is a range of investments held by an individual or an organization in the form of stocks, bonds, mutual funds, commodities, cash, and cash equivalents,

Abhishek Gunjan
Christ (Deemed to be University), Bangalore, India
e-mail: abhishek.gunjan@res.christuniversity.in

Siddhartha Bhattacharyya
Rajnagar Mahavidyalaya, Birbhum, India
Algebra University College, Zagreb, Croatia
e-mail: dr.siddhartha.bhattacharyya@gmail.com

including closed-end funds [1] and exchange-traded funds [2]. The sole purpose of maintaining a portfolio is to achieve a balance between risk and return based on the investor's financial goals. A well-diversified portfolio can help portfolio managers achieve the financial goals set, but excessive diversification should be avoided to reduce transaction cost [3][4].

Managing a portfolio involves complex decision-making and is a frequently studied topic in finance. Due to the non-linear nature of constraints, it requires complex computation and hence cannot be easily solved by traditional techniques. Near-accurate and timely calculation of weights is the key in any decision-making process and hence requires fast and more accurate techniques to be incorporated as a portfolio optimization process. Additionally, with the dawn of quantum computing and quantum-inspired techniques, metaheuristic-based techniques have been observed to outperform traditional machine learning-based techniques and have been found to be suitable for portfolio optimization problems.

In this study, an effort has been made to study different evolutionary techniques and compare them with their quantum-inspired versions. Some of the key highlights of this study include the following.

- Benchmarked datasets: Three benchmark datasets, viz. (i) NASDAQ (from 2012-06-23 to 2022-06-27) [5], (ii) BSE (from 2011-05-13 to 2023-02-07) [6], and (iii) Dow Jones (from 2009-08-06 to 2023-05-05) [7], are selected for this experiment, and the results are presented and discussed as part of this study.
- Genetic algorithm (GA): The genetic algorithm (GA) [8] is one of the first population-based metaheuristic algorithms inspired by the Darwinian theory of evolution [9][10]. As a part of this study, GA is studied in detail, and two popular crossover techniques, viz. (i) arithmetic crossover and (ii) heuristic crossover techniques, are implemented, and the results of application on the mentioned datasets are reported.
- Differential evolution (DE): Differential evolution (DE) [11][12] is another population-based metaheuristic algorithm, similar to GA, and is generally found to be faster and more accurate than GA. As a part of this study, DE is studied in detail with two popular crossover techniques, viz., (i) arithmetic crossover and (ii) heuristic crossover techniques, are implemented, and the results of application on the mentioned datasets are reported.
- Particle swarm optimization (PSO): Particle swarm optimization (PSO) [13][14] is the third evolutionary technique considered for this study. PSO has been found to be efficient in dealing with complex optimization space and volatile market conditions. As a part of this study, PSO is implemented, and the corresponding results are reported.
- Quantum-inspired versions of GA, DE, and PSO: An effort, first of its kind, has been made to design and implement GA, DE, and PSO in the quantum-inspired domain, and the results are compared with the mentioned datasets.

Portfolio managers must ensure that all funds are allocated efficiently and in the right proportion to achieve maximum gain. Additionally, it has been observed that the random selection of weights, as proposed in the evolutionary-based optimization techniques, does not guarantee that all available funds are allocated. Sometimes, the sum of all the weights could be higher or less than 100%. In order to overcome this problem, a regularization function is proposed as a part of this experiment, which is further represented as

$$w_{reg_i} = \frac{w_i}{\sum_{i=1}^{n} w_i} \tag{1}$$

where, w_{reg_i} is the regularized weight of i^{th} stock, w_i is the random weight selected, and $\sum_{i=1}^{n} w_i$ is the total weight that has been randomly selected.

Furthermore, it has been observed that the choice of parameters in the portfolio optimization process could be an iterative and time-consuming process. To overcome this challenge, a sensitivity analysis is performed, and the values of the parameters are chosen accordingly. The performance of the mentioned techniques is measured using the mean error, execution time, and fitness function (minimum risk).

The remainder of the paper is organized as follows. Section 2 briefly lists the motivation and contribution of this work. Section 3 provides an overview of portfolio management and optimization. A brief review of the related works on portfolio optimization is provided in Section 4. Section 5 briefly lists the evolutionary techniques employed in this study. Section 6 elucidates on the relevant quantum computing and quantum-inspired techniques. Section 7 presents the experimental results, and finally, the paper is concluded in Section 8.

The list of commonly used abbreviations in this study can be found in Table 1.

Table 1: Abbreviations used in this study

Abbreviation	Description	Abbreviation	Description
GA	Genetic algorithm	QiGA	Quantum-inspired genetic algorithm
DE	Differential evolution	QiDE	Quantum-inspired differential evolution
PSO	Particle swarm optimization	QiPSO	Quantum-inspired particle swarm optimization
SBPSO	Set-based particle swarm optimization	FA	Firefly algorithm
NSGA2	Non-dominated sorting genetic algorithm 2	SPEA2	Strength pareto evolutionary algorithm 2
HPNSGA2	Harmonic progression NSGA2	RINSGA2	Random immigration NSGA2
GWASFGA	Global weighting achievement scalarizing function genetic algorithm	PESA2	Pareto envelope-based selection algorithm 2
MINLP	Mixed-integer non-linear programming	ABC	Artificial bee colony
ANNs	Artificial neural networks	SVM	Support vector machine
RF	Random forest	SVR	Support vector regression
LSTM	Long-short-term memory	CQR	Chicago quantum ratio
CQNS	Chicago quantum net score	SRI	Social responsibility investment

2 Motivation and Contributions

Portfolio optimization requires a sound understanding of the market dynamics. It has always been considered a challenging problem and therefore holds a special place in the research arena [15]. Investing in the right stock at the right time is the key to any portfolio optimization problem. There are multiple techniques and guidelines proposed in this domain, but it remains a challenging problem to solve. With the recent advancements in evolutionary techniques and a leap in the computational infrastructure, it is now possible to handle large combinatorial optimization problems. Some of the other key factors which make it even more interesting [16] include (i) dynamically changing market parameters (ii) dynamically changing constraints like transaction cost, management fees, legal constraints, regulatory constraints, and similar other constraints (iii) limitation of historical data and their quality (iv) limitations of model assumptions (v) diversification challenges and constraints (vi) behavioural biases (vii) computational challenges and model complexities (viii) lack of knowledge to choose better optimization techniques based on market conditions and imposed constraints.
Additionally, portfolio optimization is highly affected by external factors, making it more challenging for portfolio managers to make decisions. Some of the key external factors [17][18] include (i) macroeconomic indicators like inflation rates, interest rates, GDP growth, and other economic behaviours (ii) geopolitical events like political instability, election results, power changes, war or war-like conditions, and other types of trade disputes (iii) central bank policies like repo rates, cash reserve ratio, and other interest rate policies (iv) market sentiments like investors' behaviour and psychology (v) natural disasters and other calamities, and (vi) global economic behaviours and trends.
This study involves the examination and evaluation of various evolutionary methods in three benchmark datasets. An effort, the first of its kind, is made to implement the same techniques in both the classical and quantum-inspired domains. Based on careful observation of the execution of GA [8][19] with arithmetic and heuristic crossovers, an enhancement is proposed in this paper, and the experimental results are presented. The highlights of the contribution include the following.

- Dynamic selection of crossover parameter values α and β. The selection of the parameters for a genetic algorithm or a differential evolution algorithm is an iterative process that can have a significant effect on the convergence of the overall solution. The performance and efficiency of the algorithms can be improved by dynamically selecting the crossover parameters, allowing them to better adapt to the complexity of the problem and achieve a balance between exploration and exploitation of the feature space. Additionally, this dynamic selection can help to tune and speed up the convergence process automatically.
- Normalization function to avoid negative allocation caused due to incorrect crossover parameter selection. The choice of incorrect crossover parameters, such as α and β, can result in negative weights, which are not desirable. To

ensure a more realistic allocation of stocks, this study proposes a normalization function that takes into account the absolute value of the weights.

- Regularization function for better utilization of allocated funds. The total allocation of a portfolio can often exceed 100% when the sum of the dynamically selected weights is taken into account. This is not desirable. The proposed regularization function ensures that the total investment does not exceed 100%.
- Dynamic selection of optimization parameters based on sensitivity analysis. A thorough sensitivity analysis can be used to assess the influence and connection between the parameters used in the methods discussed. This can then be used to reduce the number of iterations and speed up the convergence of the chosen techniques.

3 Overview of Portfolio Management

Building a successful investment in the ever-changing dynamic market requires a fundamental change in how one thinks about investment. As a portfolio manager, the sole objective is to get high returns based on the objectives set. Additionally, new advances in investments and offers have conditioned us to think beyond the strategies we grew up with. It is also a time that requires us to apply a more advanced mode of investment decisions and to have better control over our strategies. Fortunately, with the improvement of technology, we can now employ new ways of optimization and help portfolio managers in a better decision-making process.
A portfolio, in literal terms, can be described as a collection of financial investments in the form of stocks, bonds, cash, assets, and any other form of commodities that tends to give some form of return in the future. Building an ever-growing portfolio is the key objective of an individual or an organization and can be managed by the individual or organization. It has been observed that companies sometimes open multiple business models at the same time, despite their ability to execute them successfully [20]. These factors can lead to severe losses and must be considered during investment. Furthermore, a portfolio can be mathematically represented as

$$P(t) = \sum_{j=1}^{n} w_j(t) s_j \tag{2}$$

where, at any given time, t, $s_1, s_2, ..., s_n$ are the securities and $w_1, w_2, ..., w_n$ are the weights, percentages or quantities associated with the securities. The weight at time t is represented as

$$w_{tj} = \frac{\text{Amount invested in securities } s_j}{\text{Total amount invested in portfolio } P_t} \tag{3}$$

Let us assume a list of securities as *GOOGL*, *AMZN*, *AAPL* with optimum percentages as 0.3, 0.2, and 0.5, then the corresponding portfolio at time t is represented as

$$P_t = 0.3 \times GOOGL + 0.2 \times AMZN + 0.5 \times AAPL \tag{4}$$

Due to the dynamic nature of the market, portfolio managers need to make the right decision at the right time to either invest more or withdraw from the stock. In addition to that, there is also a transaction cost associated with the allocation and deallocation decisions, making this problem even more challenging and interesting. Hence, as a portfolio manager, one must consider key characteristics like (i) future events and targets been set, (ii) fast and accurate decision-making, (iii) stage in which the investments are made, and (iv) resource limitations. Typically, the job of a portfolio manager includes (i) return maximization, (ii) diversification, (iii) balancing, (iv) setting up the right strategy, and (v) investing in the right number of assets. A brief study of portfolios, portfolio types, and portfolio optimization is presented in [21].

4 Related Works

The analysis of multiple nonlinear constraints has always been a challenging proposition and portfolio optimization is no exception.
Several works have been reported in the literature to address the portfolio optimization problem. Meghwani et al. [22] proposed a portfolio optimization approach under practical constraints using multi-objective evolutionary algorithms (MOEAs). This study proposed a candidate generation procedure and a repair mechanism to optimize a large portfolio. It was found that the methods proposed in [22] could handle a wide range of constraints, such as cardinality, pre-assignment, budget, quantity, and round-lot constraints. Furthermore, techniques such as the non-dominated Sorting Genetic Algorithm 2 (NSGA2) [23], the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [24], the Global Weighting Achievement Scalarizing Function Genetic Algorithm (GWASFGA) [25], and the Pareto envelope-based Selection Algorithm 2 (PESA2) [26] have been found to be effective in optimizing large portfolios.
Generally, portfolio optimization requires addressing complex search spaces that contain both continuous and discrete variables. Utilizing a complex coding scheme (CCS) [27], which formulates the problem as mixed-integer non-linear programming (MINLP), has been demonstrated to be effective when the continuous variables are contingent on discrete variables. CCS has been shown to be efficient and reliable in dealing with multiple objectives and constraints in large portfolio optimization problems. Furthermore, NSGS2 [23], SPEA2 [24], and PESA2 [26] can be used when dealing with a limited number of restrictions [28]. In addition to the capacity to manage multiple constraints, it is also essential to take into account the generation distance when using evolutionary techniques. According to the generation distance, SPEA2 has been found to be superior to PESA2 [29].
Different versions of evolutionary techniques, such as artificial bee colony (ABC) [30], firefly algorithm (FA) [31], genetic algorithm (GA) [32], and particle swarm optimization (PSO) [13], have been found to be effective in managing portfo-

lio optimization problems [33][34]. Furthermore, a set-based particle swarm optimization (SBPSO)[34] redefines the domain of the portfolio optimization problem. Few evolutionary techniques, such as GA, suffer from the problem of balance between exploration and exploitation. To overcome this problem, two balancing methods, viz. (i) harmonic progression NSGA2 (HPNSGA2) and (ii) random immigration NSGA2 (RINSGA2), have been proposed in [35]. The results show that the RINSGA2 algorithm generates a better spread compared to NSGA2 [36].

It has also been observed that a learning component can be beneficial in tackling multiple objectives and constraints in a large portfolio optimization problem. A study on multi-objective evolutionary techniques based on guided learning has been conducted in [37][38][39][40]. In portfolio optimization, it is important to reduce the number of stocks to minimize transaction costs. Studies on this topic can be found in [41]. Furthermore, various versions of evolutionary techniques with multiple objectives and constraints have been explored in [42][43][44].

Recently, it has also been observed that the potential of evolutionary techniques can be improved by combining classical machine learning and deep learning approaches. A comprehensive comparison of various incarnations of particle swarm optimization (PSO) was conducted in [45], which also included the analysis of artificial neural networks (ANNs) [46] based on swarm intelligence. In addition, stock market data was studied using a variety of methods, such as support vector machine (SVM) [47], deep learning [48][49], and clustering [50]. In [51], a strategy was proposed that combines two machine learning models, random forest (RF) [52] and support vector regression (SVR) [53], with three deep learning models [54], long-short-term memory (LSTM) networks [55][56], deep multilayer perceptron, and convolutional neural networks [57]. Furthermore, Zheng et al. [58] suggested a hybrid model that combines XGBoost [59][60] with an improved version of the firefly algorithm.

Recent efforts have been made to use evolutionary techniques in classical and quantum-inspired domains to address non-linear portfolio optimization problems, resulting in a surge of research in this area [61][62][63]. Quantum computing has become a popular topic in fields ranging from portfolio optimization to blockchain and security [64]. A portfolio optimization technique using a DWave quantum annealer was proposed in [65]. The study employed four different measures, including mean-variance, Sharpe ratio, simplified Chicago quantum ratio (CQR), and a new Chicago quantum net score (CQNS) to identify the optimal risk. The results show DWave outperforms the Monte Carlo method and is best suited for largely constrained multi-objective functions. Furthermore, simulated quantum annealing has been shown to be more robust and computationally faster when using the D-Wave 2000Q$^{\text{TM}}$ quantum annealer in a variety of portfolio optimization problems [66][67]. Additionally, it has been observed that quantum-inspired simulated bifurcation is an effective way to tackle NP-hard combinatorial optimization problems, as demonstrated in [68].

Investors and researchers have recently been drawn to social responsibility investment (SRI) [69]. A three-step framework based on environmental, social, and governance criteria (ESG) has been proposed in [70]. Combining ESG scores with financial indicators demonstrates that it is possible to invest in a way that is consistent with social values. As [70] suggests, long-term financial returns can be achieved by reducing potential risks such as litigation, tax, compliance, and honor risks [71]. Additionally, the ESG score can be taken into account when selecting stocks to minimize potential losses. A bi-level approach based on ESG [72] has been found to be effective in reducing the overall impact on the company.

5 Evolutionary Optimization

Evolutionary computing [73] is a branch of artificial intelligence that is often used to solve complex optimization problems. It is based on the concept of biological evolution and is suitable for tackling problems that involve a large number of variables. Just like natural selection, the fitness of an individual in a population is determined by how well it is able to adapt and reach its objectives. In the context of portfolio selection, a candidate is chosen based on their potential to be a successful candidate in the evolutionary process.

The development of the first evolutionary algorithm (EA) [74][75] is attributed to a few pioneers who independently proposed different approaches, viz., (i) Evolutionary programming (EP) [76] as proposed by Fogel et al. [77] aiming at evolving finite automata, later at solving numerical optimization problems, (ii) Genetic algorithm (GA) [78] using binary string aiming to solve combinatorial optimization problems [79], (iii) Evolutionary strategies (ES) as proposed in [80] motivated by engineering problems and aiming to solve selection problems, and (iv) Genetic programming (GP) as proposed in [81] aiming to optimize computer programs. The above-mentioned techniques form the core of evolutionary computing.

Typically, any EA involves four steps, viz., (i) initialization, (ii) selection, (iii) genetic operators, and (iv) termination as mentioned in Fig. 1. A step-by-step understanding of the evolutionary computing algorithm [74] is provided in Algorithm 1.

Generally, evolutionary computing (EC) [73] algorithms include genetic algorithm (GA) [78][8], evolutionary strategies (ES) [82][83], learning classifier systems (LCS) [84], differential evolution (DE) [11][12], and estimation of distribution algorithm (EDA) [85]. Recently, swarm intelligence (SI) [86] algorithms such as ant colony optimization (ACO) [87] and particle swarm optimization (PSO) [14] have gained popularity and are categorized as EC [73] family algorithms.

In this article, three of the evolutionary computing techniques, viz. (i) genetic algorithm (GA) [8], (ii) differential evolution (DE) [11], and (iii) particle swarm optimization (PSO) [14], are implemented for addressing the portfolio optimization problem. For both the genetic algorithm (GA) and differential evolution (DE), advanced crossover techniques are used, viz. (i) arithmetic crossover and (ii) heuristic crossover with proposed enhancements are implemented. An effort has been made to come up with two hybrid approaches on top of the arithmetic and heuristic crossovers. An effort, first of its kind, is also made to implement the mentioned techniques in the quantum-inspired domain.

Algorithm 1 Evolutionary Computing Algorithm

P = initializePopulation (pop_size, no_of_stocks)
P = evaluatePopulation(P)
P_{best} = getBestSolution(P)

while testForTermination == false **do**
  $parents$ = getRandomParents($no\_of\_parents$)
  $offspring = 0$
  for $parent_i \in parents$ **do**
    $offspring_i$ = mutate($parent_i$)
    $offspring = offspring \bigcup offspring_i$
  end for
  evaluatePopulation ($offspring$)
  $bestSolution$ = getBestSolution ($offspring$, $bestSolution$)
  $P = P \bigcup offspring$
  P = environmentalSelection (P)
end while
return $bestSolution$

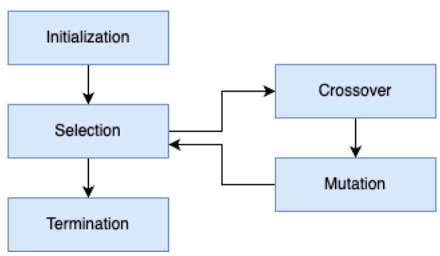


Fig. 1: Evolutionary Computing Steps

5.1 Genetic Algorithm

Genetic algorithms (GAs) [8] are population-based metaheuristic algorithms that are inspired by the Darwinian theory of evolution [9][10]. Like other evolutionary algorithms, GAs use selection, crossover, and mutation as their main operators. When dealing with portfolio optimization problems, it is important to select the right crossover technique and measure. A study of different GA techniques and measures used in portfolio optimization problems can be found in [21]. GAs evaluate randomly selected populations using a fitness function. The pseudocode for GA is provided in Algorithm 2, and the flowchart of the algorithm's operation is shown in Fig. 2.

Algorithm 2 Genetic Algorithm

P = InitializePopulation ($pop_{s}ize$, $no_{o}f_{s}tocks$)
P = Sort(P)
P_{best} = getBestSolution (P)

while testForTermination == false **do**
  Selection(population)
  Crossover(population)
  Mutate(population)
  P_{best_i} = getBestSolution (P)
  P_{best_i} = UpdateGlobalBest(P_{best_i}, P_{best})
end while
return P_{best}

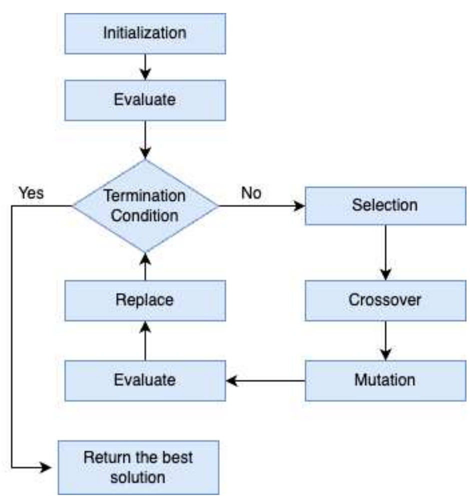


Fig. 2: Genetic Algorithm Flowchart

5.2 Differential Evolution

Differential Evolution (DE) [11][12] is a member of the Evolutionary Algorithm (EA) family, which is known to be more accurate, faster, and more reliable for opti-

mization problems. It is easy to comprehend, similar to the Genetic Algorithm (GA) [8], and can be used for portfolio optimization. DE [88] has a non-uniform population generation process that increases diversity and often produces more optimized results. The crossover used in GA is also used in DE, allowing for a better comparison. This is why DE [88] is chosen to test the modified crossover techniques in this article. Additionally, DE has been observed to perform better with floating-point weights, which is a desired outcome in portfolio optimization.
The algorithm for differential evolution is analogous to that of a genetic algorithm, as shown in Algorithm 2, with an improved population generation process. The equation used for population generation is expressed as

$$New_{wt_i} = X_{min} + wt_i * (X_{max} + X_{min}) \tag{5}$$

where, New_{wt_i} is the new weight generated through the DE process, X_{min} is the minimum weight of the allocation, wt_i is the current weight, and x_{max} is the maximum allocation possible for a portfolio.

5.3 Crossover Techniques Used

The effectiveness of genetic algorithm (GA) [8] and differential evolution (DE) [89][90] depends on three factors, viz. (i) population, (ii) mutation rate, and (iii) crossover parameters chosen. This article proposes improvements to the popular crossover techniques, viz., arithmetic crossover [91] and heuristic crossover [92], for portfolio optimization problems. A comparative study of these crossover operators is made on different datasets to establish the significance of the proposed enhancements.

5.3.1 Arithmetic Crossover

The arithmetic crossover operator [93] combines two parents linearly to generate children. The mathematical technique used is given by

$$\begin{aligned} ch_1 &= \alpha * p_1 + (1-\alpha) * p_2 \\ ch_2 &= (1-\alpha) * p_1 + \alpha * p_2 \end{aligned} \tag{6}$$

where, ch_1 is the first child, ch_2 is the second child, α is the crossover rate, p_1 is the first parent, and p_2 is the second parent selected for the crossover operation.

5.3.2 Heuristic Crossover

The heuristic crossover operator [94] combines two parents to generate children that are between the best and worst selected parents using the following expression.

$$\begin{aligned} ch1 &= p\_best + \beta * (p\_best - p\_worst) \\ ch2 &= p\_worst - \beta * (p\_best - p\_worst) \end{aligned} \tag{7}$$

where, $ch1$ is the first child, $ch2$ is the second child, β is the crossover rate, *p_best* is the best parent, and *p_worst* is the worst parent selected for the crossover operation.

5.3.3 Challenges in the Crossover Methods for Portfolio Optimization

When it comes to portfolio optimization, the mentioned crossover operations exhibit four problems, viz., (i) these are highly dependent on the selection of the appropriate value of α based on the problem statement (ii) the mentioned crossover techniques cannot be generalized for different problems (iii) the child weights generated could be negative, which denotes negative investment and (iv) the total of allocation could be less than or more than 100% of the available funds. The objective of crossover in any portfolio optimization problem is to ensure that it generates non-negative weights and that the total allocation should not be greater than 100%. Furthermore, for the best return, the optimizer should allocate all the available funds.
An effort has been made in this work to overcome the above-mentioned challenges in the arithmetic and heuristic crossovers by introducing two improvements, viz. (i) a circular regularization function to ensure 100% allocation and (ii) by considering absolute weights to avoid non-zero values.

5.4 Particle Swarm Optimization

Particle swarm optimization (PSO) [13][14] is one of the evolutionary techniques that is based on swarm intelligence [14]. PSO intends to simulate social behavior as a representation of the movement of organisms in a flock of birds or a fish school and can also be used in a volatile market condition. PSO considers each member of the population as a particle. PSO [14][95] uses the technique of ”gbest neighborhood topology” as proposed by Kennedy et al. [96]. In multiple research studies, it has been proven that PSO outperforms other evolutionary techniques such as GA [8] and DE [11][12] and therefore has been used in this work. PSO has several advantages over other evolutionary techniques and has been applied in areas involving function optimization, artificial neural network training, fuzzy control systems, and fields related to finance and accounting. The evolution process in PSO can be given by [97]

$$\begin{aligned} V_i^k &= wV_i^k + c_1 r_1 (pbest_i^k - x_i^k) + c_2 r_2 (gbest^k - x_i^k) \\ x_i^{k+1} &= x_i^k + v_i^{k+1} \end{aligned} \tag{8}$$

where, v_i^k is the velocity of the i^{th} particle at the k^{th} iteration, and x_i^k is the current solution (or position) of the i^{th} particle at the k^{th} iteration. c_1, c_2 are positive constants, and r_1, r_2 are two random variables between (0, 1). In Equation 8, w is the inertia weight of the stock. The pseudo-code of PSO is given in Algorithm 3.
A survey on PSO-based financial application [98] in portfolio optimization is presented in [45]. The proposed paper [45] studied various computational intelligence-based approaches and financial trading frameworks employed for forecasting using ANNs [99], SVMs [100], PSO [14], GA [8], ABC [101] and their impacts. It is found that the swarm intelligent techniques outperform the classical techniques in dynamic optimization problems, which are discrete, continuous, constrained, or multi-objective [102] in nature. The survey as proposed in [45], was performed on multiple stocks from different datasets such as BSE, NASDAQ, S&P 500, and the Shanghai stock market. Furthermore, a careful study of the stock market shows that a group of stocks has a certain common property in terms of the ways they move up and down. These kinds of stocks can be handled by applying an improved set-based PSO (SBPSO) as proposed in [103]. The set-based approach uses quadratic programming as a weight optimizer for determining the contribution of the individual assets. The proposed techniques were tested on multiple datasets, such as Hang Seng, DAX 100, FTSE 100, S&P 100, and Nekkei. The experiments conducted in [103] show that although the SBPSO result is comparable to GA, it outperforms GA in terms of speed.
Of late, it has been observed that the traditional PSO may lead to a local optimum solution. The article proposed in [104] used a hybrid approach where PSO is combined with SA [105][106], and ANN [107] to reduce the chances of local optimum solution and can be employed to portfolio optimization problems. PSO can also be extended to multi-objective function optimization as proposed in [108][109].

6 Quantum-inspired Metaheuristics for Portoflio Optimization

In recent years, a lot of effort has been devoted to the use of hybrid approaches to make portfolio optimization more realistic. It has been discovered that quantum and quantum-inspired computing techniques can be beneficial in solving complex optimization problems [62]. Quantum computing [110] uses the principles of quantum mechanics to process information. Unlike classical computing, which represents information as bits, quantum computing uses quantum bits or qubits to represent information. Qubits exist in a superposition of states, which means that the information can be both $|0\rangle$ and $|1\rangle$ simultaneously.
Quantum computing [110] is found to perform better than classical computers, especially in problems that involve searching and finding the optimum solution on

Algorithm 3 Particle Swarm Optimization Algorithm

Input: $N, x_l, X_u, C_1, C_2, i_{max}, f$
Output: A swarm S of size N (N position vectors)

Initialize S, randomly generate the position x of each particle w.r.t. the bounds x_l, x_u of the objective function
Initialize all velocities u to zero
Initialize best positions $x*$ (and respective values) for individual particles and find $g*$
Choose randomly two values in [0,1] for r_1 and r_2
Iteration $i = 0$
Initialize θ_{min}, θ_{max}

while i ¡ i_{max} **do**
  Calculate inertia: $\theta = \theta_{max}$ - $\frac{\theta_{max}-\theta_{min}}{i_{max}} * i$
  for $p_i \in S$ **do**
    1. Update velocity: $u_i = \theta * u_{i-1} + c_1 r_1\ [x^* - x_{i-1}] + c_2 r_2\ [g^* - x(i-1)]$;
    2. Update position: $x = x_{i_1} + u_i$;
    3. Compute the value of the new position according to f;
    4. Check/Update: x^*, g^*
  end for
  (Optional) Check for convergence
  Upgrade iteration: $i = i + 1$
end while
return S;

large datasets. The representation of quantum states is done through a wave function in the Hilbert space $\mathscr{H}$ [111]. This space can be thought of as an extension of a two-dimensional or three-dimensional space to one with a finite or infinite number of dimensions.
Quantum gates [112] are the building blocks of quantum circuits [113][114]. There are several gates used in quantum computing, including Hadamard gates, Pauli gates, phase gates, and CNOT gates, among others. The choice of gates plays a vital role in the performance of quantum circuits. In this article, the Hadamard gate [115], as mentioned in Equation, 9, has been used due to its simplicity and robustness.

$$Hadamard\ (H) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{9}$$

6.1 Qubit

Quantum bit or qubit is the smallest unit of quantum information in quantum computing [110] and is represented as $|0\rangle$ and $|1\rangle$. The qubit vectors are also represented using Dirac notation [116] as

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$$
$$|1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \tag{10}$$

where, $|0\rangle$ implies that the qubit is in the ground state and $|1\rangle$ implies that the qubit is in the excited state.
Qubits possess a remarkable property known as entanglement, which allows the state of one qubit to be linked to the state of another qubit, even when they are separated by a large distance. This property enables parallel computing, making quantum computers much faster than traditional computers.

6.2 Quantum Superposition

The quantum superposition principle is the linear combination of quantum vectors represented in two states as [110]

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{11}$$

where, α and β are complex numbers which must satisfy the condition

$$|\ \alpha\ |^2 + |\ \beta\ |^2 = 1 \tag{12}$$

where, α^2 and β^2 are the probabilities of measuring the basis states $|0\rangle$ and $|1\rangle$, respectively. A qubit can be expressed as a combination of two basis states, $|0\rangle$ and $|1\rangle$, which are created using quantum gates (Q-gates). This combination of the states is maintained until a quantum measurement destroys the superposition. Therefore, for n qubits, the number of states in a quantum machine is 2^n [117][118].

6.3 Quantum-inspired techniques for portfolio optimization

Computer scientists often face a challenge in creating advanced, reliable, and faster algorithms for dynamic and complex optimization problems. Evolutionary techniques have been found to be effective in any given problem space. About two decades ago, new optimization techniques based on quantum computing were proposed [118][119] to address search optimization issues. Quantum-inspired metaheuristic algorithms imitate the principles of quantum mechanics and have been demonstrated to outperform classical techniques in solving combinatorial optimization problems [120][121]. Due to their capacity to solve complex and large

computational problems, these quantum-inspired metaheuristics are widely used in constrained and unconstrained problems [122][62]. A survey of quantum-inspired metaheuristics-based techniques is presented in [123].
In recent times, various versions of quantum-inspired metaheuristics have been developed to create quantum-inspired versions of classical metaheuristics, such as the genetic algorithm [124], tabu search [125], differential evolution [11][12], particle swarm optimization [14], and ant colony optimization [126] to be used in quantum space. Examples of these include the quantum-inspired genetic algorithm [127][128][129], quantum-inspired tabu search [130][131][132], quantum-inspired differential evolution [133], quantum-inspired particle swarm optimization [134][135], and quantum-inspired ant colony optimization [136][137]. A comprehensive examination of the progress made in support vector machines, neural networks, and evolutionary computing can be found in [138][139].

6.4 Quantum-inspired techniques used in this study

Quantum-inspired methods have been found to be more effective than their classical equivalents when tackling issues of large search space and high dimensionality. This approach takes advantage of quantum parallelism and superposition to explore the search space more thoroughly, leading to a more optimal solution. The main features of the implementation include the following.

- Designing a quantum-inspired population: Quantum-inspired techniques use qubits to represent an individual in the population. Each population consists of two weights for each stock taken into account to form the portfolio. The individual weights have been encoded using qubits $|0\rangle$ and $|1\rangle$. Additionally, quantum-inspired crossover and mutation take advantage of quantum mechanical principles to generate entanglement and introduce randomness.
- Implementation of Hadamard gates: The selection of weights in the population, crossover, and mutation has been performed using the Hadamard gate, as mentioned in Equation 9.
- Implementation of the fitness function: Finally, the fitness function sorts the population considering both states, viz., (i) α, and (ii) β.

7 Experimental Results and Analysis

This study uses the genetic algorithm, differential evolution, particle swarm optimization, and their quantum-inspired counterparts for comparison. The mentioned techniques have been implemented (coded) in Python on a MacBook Pro with an Apple M2 chip with 8 GB of RAM and a macOS Ventura 13.4.1. Experiments have been conducted on three benchmark datasets, viz. (i) NASDAQ [5] with 2515 data points from 2012-06-28 to 2022-06-27, (ii) Bombay Stock Exchange (BSE) [6] with 2285 data points from 2013-11-05 to 2023-02-13, and (iii) Dow Jones [7] with 3461 data points from 2009-08-06 to 2023-05-05. An effort has been made to consider a better mix of stocks that had positive and negative growths in the selected period. The average and highest percentage changes of stock are calculated using the following equations:

$$APC_i = \frac{S_i^n - S_i^1}{S_i^1} * 100 \tag{13}$$

$$HPC_i = \frac{max(S_i) - min(S_i)}{min(S_i} * 100 \tag{14}$$

where, APC_i and HPC_i are the average and highest percentage changes of stock S_i, respectively. S_i^n and S_i^1 are the n^{th}, and the 1^{st} observations of the stock, respectively, and $max(S_i)$ and $min(S_i)$ are the corresponding maximum and minimum observations of the stock in the given period.

Table 2 presents the NASDAQ sample data used in this study. The graphical representation of the data, as shown in Fig. 3, shows a mix of stocks that have changed in different proportions over the last 10 years. Table 3 presents the average and highest percentage changes of individual stocks according to Equations 13 and 14, respectively. The Average Percentage Change (APC) and the Highest Price Change (HPC) of the chosen stocks vary from 80.86% to 4546.66% and 180.05% to 11585.16%, respectively, indicating a good diversification of the portfolio.

Table 4 presents the BSE sample data used in this study. The graphical representation of the data, as shown in Fig. 4, shows a mix of stocks that have changed in different proportions over the last 10 years. Table 5 presents the average and highest percentage changes of individual stocks according to Equations 13 and 14, respectively. The average percentage change (APC) of the chosen stocks ranges from -66.88% to 361.31%, while the highest price change (HPC) varies from 447.28% to 1496.01%.

Table 6 presents the sample data from Dow Jones used in this study. The graphical representation of the data, as shown in Fig. 5, shows a mix of stocks that have changed in different proportions over the last 10 years. Table 7 presents the average and highest percentage changes of individual stocks according to Equations 13 and 14, respectively. The average percentage change (APC) of the chosen stocks ranges from 66.24% to 1182.61%, while the highest price change (HPC) varies from 345.47% to 5933%.

An attempt is therefore made to pick those stocks that have experienced significant price fluctuations, with a high and a low in the chosen time frame.

Table 2: Sample NASDAQ [5] Dataset

Date	AAPL	MSFT	CSCO	SBUX	NVDA	JPM	PFE
6/28/2012	20.35	29.735	16.425	25.96	3.27875	35.39	21.97355
6/29/2012	20.68	30.415	16.9475	26.56	3.42375	36.025	22.3401
7/2/2012	21.02	30.415	17.06	26.3975	3.405	35.975	22.4525
7/3/2012	21.32	30.465	17.12	26.08	3.40875	35.8375	22.3645
7/5/2012	21.68	30.5775	16.955	26.1925	3.4025	34.94	22.2179
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
6/21/2022	135.19	252.13	43.99	72.6	167.075	116.035	47.697
6/22/2022	135.84	253.77	43.6425	73.02	164.21	115.29495	48.6749
6/23/2022	137.11	256.5	43.0875	74.435	162.19	113.33	49.575
6/24/2022	140.84	264.85	43.7275	76.85	167.25	115.325	50.97
6/27/2022	142.23	265.79	43.9575	77.8525	169.7775	117.07085	51.7425

** AAPL = Apple Inc, MSFT = Microsoft Corp, CSCO = Cisco Systems, SBUX = Starbucks Corp*
** NVDA = NVIDIA Corp, JPM = JP Morgan Chase, PFE = Pfizer Inc*

Table 3: Average and Highest Percentage Change of NASDAQ [5] Dataset

Parameters	AAPL	MSFT	CSCO	SBUX	NVDA	JPM	PFE
Min Price	14.01	26.485	15.26	21.805	2.8475	33.8	21.6999
Max Price	181.03	344.61	63.9175	125.485	332.735	171.72	60.77
HPC(%)	1192.15	1201.15	318.86	475.49	11585.16	408.05	180.05
APC(%)	778.51	658.64	80.86	144.94	4546.66	112.05	90.76

** AAPL = Apple Inc, MSFT = Microsoft Corp, CSCO = Cisco Systems, SBUX = Starbucks Corp*
** NVDA = NVIDIA Corp, JPM = JP Morgan Chase, PFE = Pfizer Inc*

Table 4: Sample BSE [6] Dataset

Date	ADANIPORTS	DLF	PRESTIGE	TATASTEEL	UPL
11/5/13	151.85	160.8	143.5	338.95	166.95
11/6/13	150.9	155.9	144.95	333.3	171.95
11/7/13	148.3	148.85	147.5	345.8	167.75
11/8/13	148.4	152.6	138.45	355.75	167.9
11/11/13	149.9	148.2	130.75	360.8	165.55
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
2/13/23	553.7	357.9	400.95	108.75	733.8
2/14/23	565.1	353.95	395.9	109.25	761.6
2/15/23	569.05	361.0	405.8	110.3	769.35
2/16/23	577.2	371.7	419.95	112.0	770.35
2/17/23	578.65	364.55	433.25	112.25	770.15

** ADANIPORTS = Adani Ports, DLF = Delhi Land & Finance, PRESTIGE = Prestige Group*
** TATASTEEL = Tata Steel Ltd, UPL = United Phosphorus Limited*

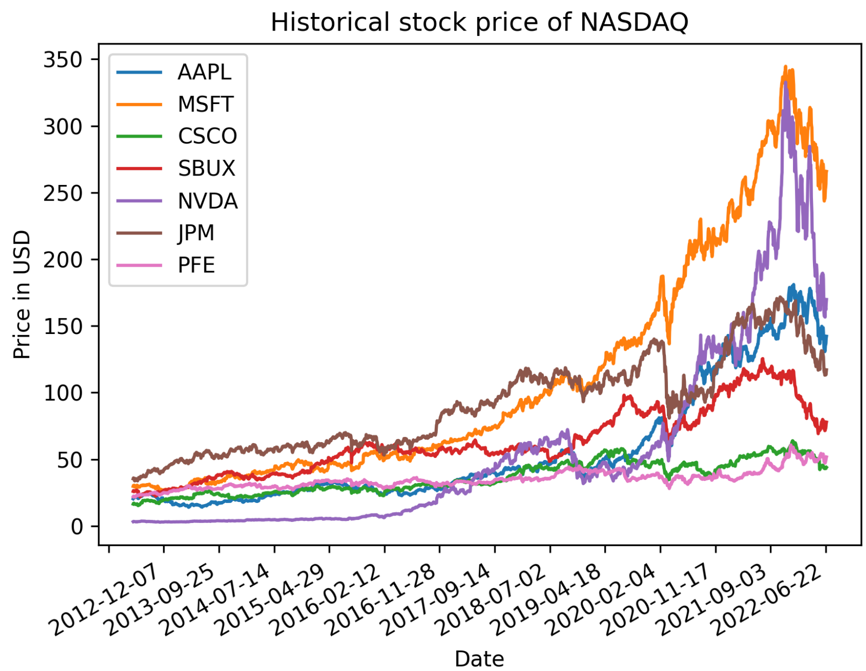


Fig. 3: NASDAQ [5] Historical Data

Table 5: Average and Highest Percentage Change of BSE [6] Dataset

Parameters	ADANIPORTS	DLF	PRESTIGE	TATASTEEL	UPL
Min Price	141.65	80.8	124.65	95.2	158.05
Max Price	970.25	442.2	526.65	1519.4	1034.5
HPC(%)	584.97	447.28	322.50	1496.01	554.54
APC(%)	281.07	126.71	201.92	-66.88	361.31

* *ADANIPORTS = Adani Ports, DLF = Delhi Land & Finance, PRESTIGE = Prestige Group*
* *TATASTEEL = Tata Steel Ltd, UPL = United Phosphorus Limited*

7.1 Enhancement Analysis

This study proposes three enhancements, viz., (i) a model to select positive weights, (ii) a regularization function to ensure full utilization of the available funds, as presented in Equation 1, and (iii) random selection of the parameters of the metaheuristics under consideration. A comparative analysis with and without the mentioned enhancements is presented in Tables 8 - 11. The results demonstrate that the cal-

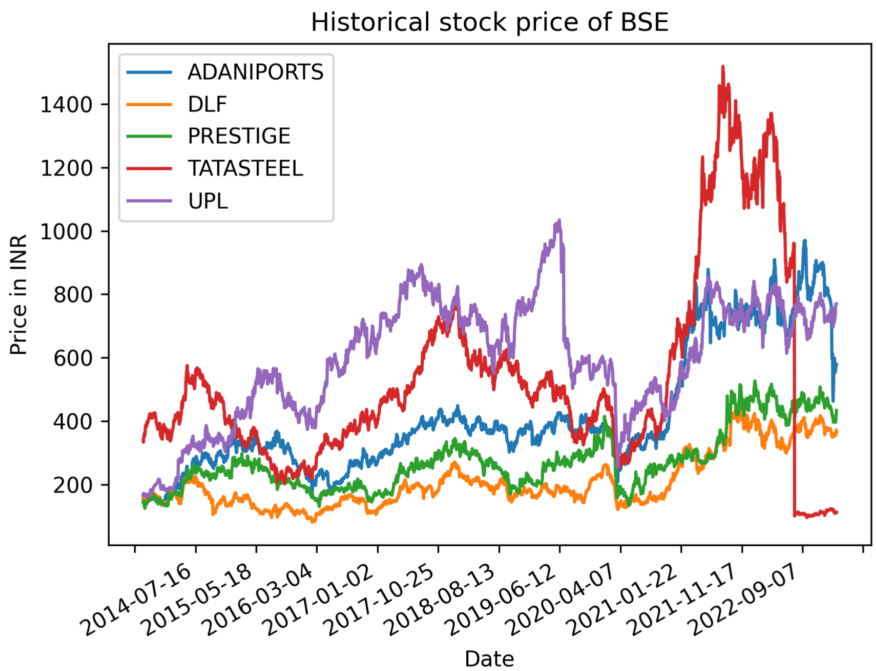


Fig. 4: BSE [6] Historical Data

Table 6: Sample Dow Jones [7] Dataset

Date	AVGO	BA	GOOGL	JPM	UNH	XOM
06/08/09	12.0039	35.1112	11.2939	28.7499	21.3532	41.4645
07/08/09	12.1893	36.0137	11.4629	29.8858	21.7888	41.3099
10/08/09	11.8481	35.304	11.4506	30.1186	22.2491	41.1374
11/08/09	11.6255	35.0109	11.3835	29.0956	22.8655	40.7628
12/08/09	11.8703	35.7437	11.5	29.78	22.8408	41.3492
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
01/05/23	637.95	203.87	107.2	141.2	495.7	114.67
02/05/23	612.34	203.25	105.32	138.92	493.39	110.1
03/05/23	613.2	200.93	105.41	135.98	489.44	107.93
04/05/23	610.16	197.05	104.69	134.12	487.28	106.04
05/05/23	630.12	198.34	105.57	136.74	494.28	108.68

** AVGO = Broadcom Inc, BA = Boeing Co, GOOGL = Alphabet Inc*
** JPM = JPMorgan Chase & Co, UNH = UnitedHealth Group Inc, XOM = Exxon Mobil Corp*

Table 7: Average and Highest Percentage Change of Dow Jones [7] Dataset

Parameters	AVGO	BA	GOOGL	JPM	UNH	XOM
Min Price	10.75	33.57	10.94	20.57	19.76	26.57
Max Price	648.55	430.35	149.84	163.96	551.47	118.34
HPC(%)	5933.00	1182.00	1270.20	697.17	2691.02	345.47
APC(%)	1182.61	81.98	272.35	196.92	596.70	66.24

** AVGO = Broadcom Inc, BA = Boeing Co, GOOGL = Alphabet Inc*
** JPM = JPMorgan Chase & Co, UNH = UnitedHealth Group Inc, XOM = Exxon Mobil Corp*

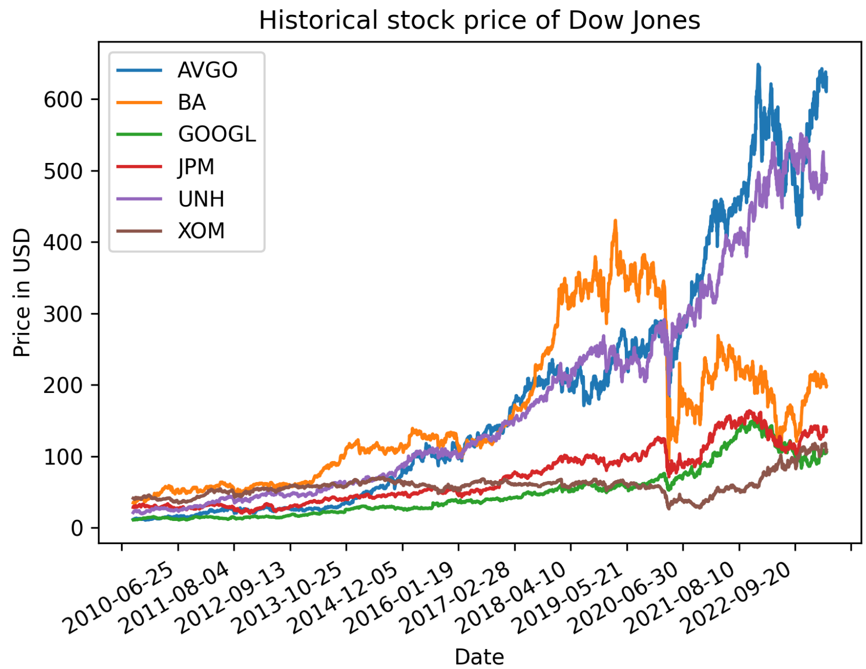


Fig. 5: Dow Jones [7] Historical Data

culated risk and return without the mentioned enhancements are not ideal, and the total allocation exceeds the available funds.

Table 8: Arithmetic crossover results without proposed enhancements

Risk	Return	Total Allocation	Allocation Distribution
0.43725948	0.49238143	2.49799162	[0.91392113, 0.75009807, 0.50433553, 0.32963688]
0.42476089	0.49083262	2.36742172	[0.87472188, 0.88431377, 0.10388881, 0.50449726]
0.32943594	0.38400195	1.8772583	[0.51240134, 0.84583051, 0.40922836, 0.10979809]
0.39335253	0.44944748	2.23307304	[0.78943504, 0.75417708, 0.24070072, 0.4487602]
0.53382178	0.58226318	3.12733067	[0.86295175, 0.8789972, 0.85539376, 0.52998795]
0.49046968	0.53759106	2.86778851	[0.82048647, 0.81216284, 0.7524143, 0.4827249]
0.42946743	0.4861936	2.40896752	[0.97191603, 0.72469758, 0.25047287, 0.46188103]
0.4022885	0.46315369	2.2572317	[0.62911085, 0.96077614, 0.04096606, 0.62637865]
0.51614329	0.56336109	3.00832468	[0.825731, 0.89577398, 0.94290855, 0.34391116]
0.54348558	0.60302327	3.17919337	[0.8731615, 0.98017644, 0.69998464, 0.62587079]
0.54334289	0.60284858	3.17846592	[0.87229227, 0.98015662, 0.69977728, 0.62623975]
0.44158135	0.50057985	2.56729824	[0.67861343, 0.92388471, 0.42016579, 0.54463432]
0.38787203	0.44786412	2.24110515	[0.58530536, 0.92776234, 0.45919353, 0.26884392]
0.50496791	0.55090699	2.95312069	[0.83487292, 0.80118576, 0.63636414, 0.68069788]
0.48737856	0.52344968	2.80362232	[0.88260513, 0.66265358, 0.3980223, 0.86034131]
0.48298803	0.51582622	2.79231058	[0.85166369, 0.63895669, 0.46072106, 0.84096914]
0.43614161	0.44664829	2.56609622	[0.18641471, 0.88725884, 0.89863014, 0.59379252]
0.39925686	0.42571745	2.25683827	[0.86791265, 0.4389099, 0.28688973, 0.66312599]
0.27709581	0.3015809	1.61839942	[0.44922116, 0.46147304, 0.49121167, 0.21649355]

Table 9: Arithmetic crossover results with proposed enhancements

Risk	Return	Total Allocation	Allocation Distribution
0.18816554	0.2274488	1.0	[0.3773142, 0.55804829, 0.04523255, 0.01940496]
0.18816974	0.22738958	1.0	[0.37896231, 0.55619314, 0.04586476, 0.01897978]
0.18863448	0.225746	1.0	[0.42470353, 0.5047058, 0.06341113, 0.00717954]
0.18866373	0.22568678	1.0	[0.42635164, 0.50285065, 0.06404334, 0.00675437]
0.18508445	0.22075708	1.0	[0.40043325, 0.48155361, 0.09335309, 0.02466004]
0.18050009	0.21662506	1.0	[0.29910234, 0.52402136, 0.07090727, 0.10596902]
0.17733953	0.20972876	1.0	[0.19702565, 0.55786266, 0.17837042, 0.06674126]
0.17796369	0.2111444	1.0	[0.31235524, 0.46608254, 0.11007477, 0.11148744]
0.18343888	0.21824601	1.0	[0.39350049, 0.45850616, 0.05497639, 0.09301696]
0.17945013	0.21392784	1.0	[0.32285206, 0.4835512, 0.10875281, 0.08484394]
0.18401832	0.21719516	1.0	[0.39883742, 0.43768039, 0.00598494, 0.15749725]
0.19262115	0.22393166	1.0	[0.52841532, 0.40113899, 0.06518147, 0.00526422]
0.17493263	0.20287877	1.0	[0.28283704, 0.42698923, 0.22238122, 0.0677925]
0.18882082	0.21838108	1.0	[0.01642904, 0.78548234, 0.13829853, 0.0597901]
0.18054424	0.21166618	1.0	[0.39513283, 0.3995517, 0.09192361, 0.11339186]
0.1702984	0.19086392	1.0	[0.17479474, 0.40371664, 0.22623568, 0.19525293]
0.17073331	0.19078806	1.0	[0.17316756, 0.39874034, 0.18326016, 0.24483194]
0.17160148	0.19128208	1.0	[0.29125646, 0.30769298, 0.20511686, 0.1959337]
0.17220267	0.18922145	1.0	[0.32176341, 0.26625287, 0.23249608, 0.17948765]
0.17087154	0.19057586	1.0	[0.24175251, 0.35116623, 0.27279687, 0.13428439]

7.2 Sensitivity Analysis

The implementation of evolutionary techniques requires the selection of multiple parameters, which can be a difficult task. Sensitivity analysis [140] can be used to assess the influence of the input parameters on the result. In recent years, sensitivity analysis has been used in several fields, such as finance, engineering, operations research, and risk management, to name a few, to evaluate the reliability and robustness of the selected model. In this study, SALib [141][142] in Python has been used to study the impact of the input parameters, such as population size, α and β val-

Table 10: Heuristic crossover results without proposed enhancements

Risk	Return	Total Allocation	Allocation Distribution
12.78774304	15.52159253	68.14638814	[15.63910484, 46.76028321, 3.95945057, 1.78754952]
80.50210392	95.96807896	410.61290795	[65.55466873, 332.25330684, 49.7201455, -36.91521311]
0.45690216	0.51731405	2.37234613	[1.27899754, 0.76578453, -0.09409481, 0.42165887]
0.45690216	0.51731405	2.37234613	[1.27899754, 0.76578453, -0.09409481, 0.42165887]
80.50210392	95.96807896	410.61290795	[65.55466873, 332.25330684, 49.7201455, -36.91521311]
184.94947197	220.1189818	938.82372655	[145.83274446, 769.81540759, 114.82411594, -91.64854144]
125.1402649	148.5916856	625.73068819	[94.72720247, 532.35335504, 71.61588591, -72.96575523]
125.1402649	148.5916856	625.73068819	[94.72720247, 532.35335504, 71.61588591, -72.96575523]
29.52603975	35.0836751	148.84219306	[23.0731789, 123.86913584, 18.35763402, -16.45775571]
29.52603975	35.0836751	148.84219306	[23.0731789, 123.86913584, 18.35763402, -16.45775571]
3.21806463	3.63307101	12.63174527	[1.87356301, 16.40048854, -1.51976405, -4.12254223]
3.21806463	3.63307101	12.63174527	[1.87356301, 16.40048854, -1.51976405, -4.12254223]
193.54928735	230.01028598	984.78757617	[149.67536653, 802.98899807, 129.10691337, -96.98370181]
193.54928735	230.01028598	984.78757617	[149.67536653, 802.98899807, 129.10691337, -96.98370181]
32.23241822	37.90186445	149.96188587	[22.66311619, 148.09539659, 7.59941849, -28.3960454]
32.23241822	37.90186445	149.96188587	[22.66311619, 148.09539659, 7.59941849, -28.3960454]
0.54453191	0.61388072	2.97756193	[1.17561256, 0.99326772, -0.04706333, 0.85574498]
0.54453191	0.61388072	2.97756193	[1.17561256, 0.99326772, -0.04706333, 0.85574498]
215.76963744	255.92805254	1144.87305041	[174.87786301, 830.22726443, 198.04197708, -58.27405412]
215.76963744	255.92805254	1144.87305041	[174.87786301, 830.22726443, 198.04197708, -58.27405412]

Table 11: Heuristic crossover results with proposed enhancements

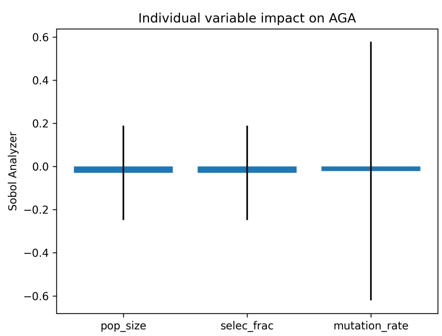
Risk	Return	Total Allocation	Allocation Distribution
0.1932686	0.23587147	1.0	[0.27512947, 0.71742572, 0.00057264, 0.00687217]
0.19497197	0.23733354	1.0	[0.23236919, 0.76697335, 0.00040629, 0.00025117]
0.19317858	0.23576149	1.0	[0.2800743, 0.71283359, 0.00500797, 0.00208414]
0.19282985	0.23507993	1.0	[0.3445041, 0.6522532, 0.00252911, 0.00071359]
0.19195767	0.23429285	1.0	[0.30104034, 0.68109812, 0.00271741, 0.01514412]
0.19200532	0.23436324	1.0	[0.29534295, 0.68631949, 0.00116388, 0.01717368]
0.19573441	0.23747934	1.0	[0.20122682, 0.79461686, 0.002399, 0.00175732]
0.19448238	0.23640207	1.0	[0.21795384, 0.77127206, 0.00882512, 0.00194898]
0.19448238	0.23640207	1.0	[0.21795384, 0.77127206, 0.00882512, 0.00194898]
0.19199174	0.23434572	1.0	[0.29544642, 0.68607097, 0.00119508, 0.01728753]
0.19444843	0.23589372	1.0	[0.19878253, 0.78241116, 0.00848771, 0.01031861]
0.19097213	0.23283073	1.0	[0.32729071, 0.64511006, 0.00117222, 0.02642701]
0.19019824	0.23176323	1.0	[0.27892182, 0.67982912, 0.03533553, 0.00591353]
0.19019824	0.23176323	1.0	[0.27892182, 0.67982912, 0.03533553, 0.00591353]
0.19536023	0.23642932	1.0	[0.18248008, 0.80162552, 0.01248911, 0.00340529]
0.19452411	0.23564206	1.0	[0.18604442, 0.79002642, 0.00364933, 0.02027982]
0.19030738	0.23189694	1.0	[0.26633194, 0.69090992, 0.02998883, 0.01276931]
0.19037710	0.23194001	1.0	[0.26631337, 0.6918704, 0.03409392, 0.00772231]
0.19184019	0.23291573	1.0	[0.21327059, 0.74570628, 0.03635607, 0.00466707]
0.19184019	0.23291573	1.0	[0.21327059, 0.74570628, 0.03635607, 0.00466707]

ues, and cognitive parameters on the objective function in the selected evolutionary techniques. The input parameters are systematically varied, and the impact on the objective function is studied. Some of the popular methods proposed for conducting sensitivity analysis include:

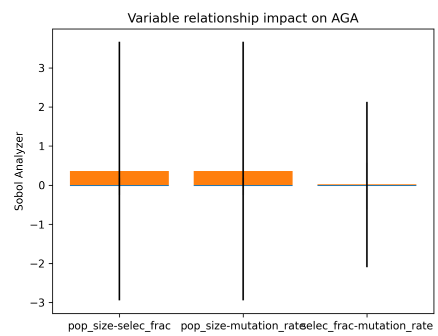
- One at a time [143]: This is one of the simplest forms of sensitivity analysis in which one variable is modified while keeping others constant. The change in the output variable is observed against each change in the input parameter, and the impact of individual parameters is studied.

- Tornado diagram [144]: Also referred to as a tornado chart, it graphically represents the sensitivity of the output parameter to the change in the input parameters. Typically, it ranks the parameters according to their influence on the output parameter.
- Monte Carlo simulation [145]: This technique involves a two-step process viz., (i) generation of random values for each input parameter in the defined ranges and (ii) running multiple simulations of the generated input parameter. It provides a probabilistic-based understanding of the impact of the input parameters on the output parameter.
- Design of experiment (DOE) [140]: This technique involves varying the input parameters based on a predefined experiment. The impact on the output parameter is studied with the change in the input parameters.

For this study, Monte Carlo simulation [145] and DOE [140] have been used. The impact of individual parameters and second-order correlation is studied and has been presented for the selected evolutionary techniques in both the classical and quantum-inspired domains. A sensitivity analysis has been performed for each of the selected methods, and the best parameter values have been identified for each technique. Figs. 6 - 15 present the best results of the sensitivity analysis of the techniques chosen from multiple executions.



(a) Independent variable impact

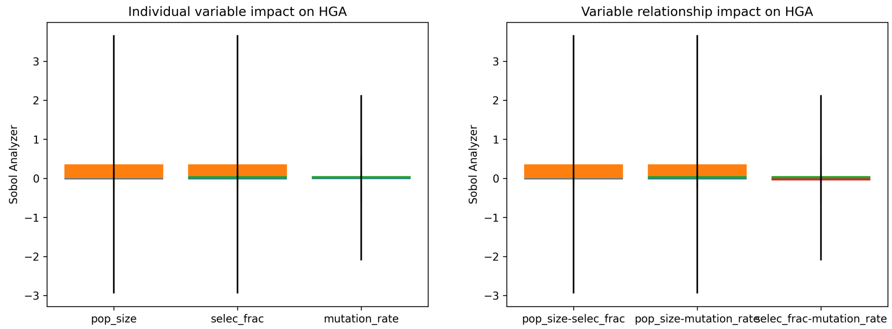


(b) Second order variable relationship

Fig. 6: Sensitivity analysis and variable impact in GA with arithmetic crossover

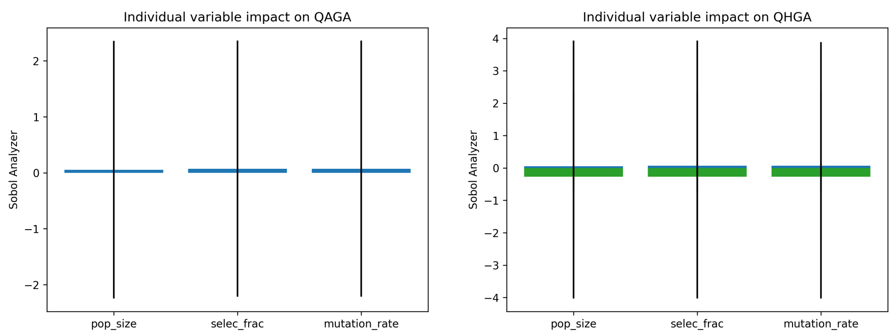
7.3 Analysis of Experimental Results

The results of the experiments carried out with GA, DE, and PSO in both the classical and quantum-inspired domains are presented in Tables 13 - 18. The optimizers have been executed multiple times with different iteration values of K, where K is



(a) Independent variable impact (b) Second order variable relationship

Fig. 7: Sensitivity analysis and variable impact in GA with heuristic crossover



(a) Independent variable impact (b) Second order variable relationship

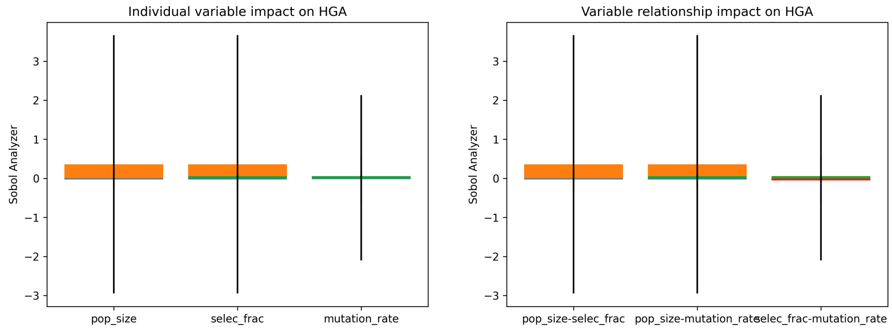
Fig. 8: Sensitivity analysis and variable impact in QiGA with arithmetic crossover

the number of times the optimizers were executed. The performance of the evolutionary techniques is evaluated using statistical measures such as mean square error (MSE), mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) [146] given by

$$Mean\ Square\ Error\ (MSE) = \frac{1}{N} * \sum_{i=1}^{n} (X_i - \bar{X})^2 \tag{15}$$

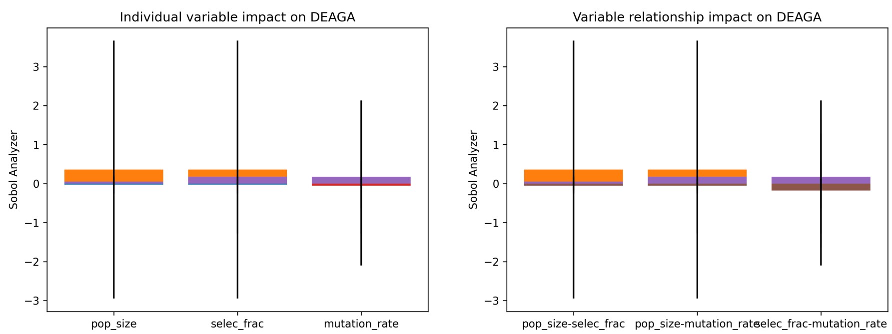
$$Mean\ Absolute\ Error\ (MAE) = \frac{1}{N} * \sum_{i=1}^{n} \mid X_i - \bar{X} \mid \tag{16}$$

$$Root\ Mean\ Square\ Error\ (RMSE) = \sqrt{\frac{1}{N} * \sum_{i=1}^{n} (X_i - \bar{X})^2} \tag{17}$$



(a) Independent variable impact (b) Second order variable relationship

Fig. 9: Sensitivity analysis and variable impact in QiGA with heuristic crossover



(a) Independent variable impact (b) Second order variable relationship

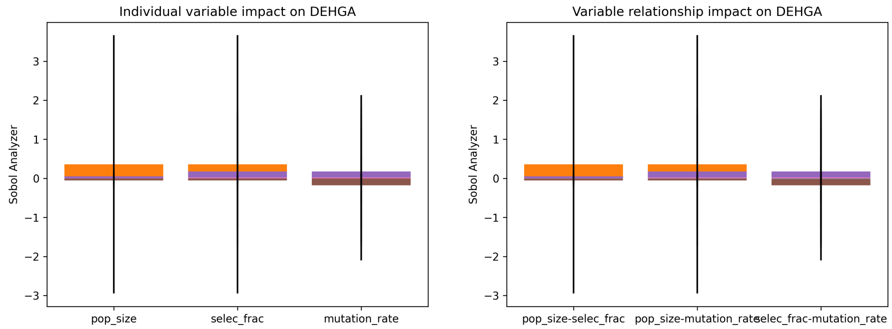
Fig. 10: Sensitivity analysis and variable impact in DE with heuristic crossover

$$Mean\ Absolute\ Percentage\ Error\ (MAPE) = \frac{1}{N} * \sum_{i=1}^{n} \frac{\mid X_i - \bar{X} \mid}{X_i} * 100 \qquad (18)$$

where, X_i is the i^{th} observation, $\bar{X}$ is the mean observation, and N is the number of executions.

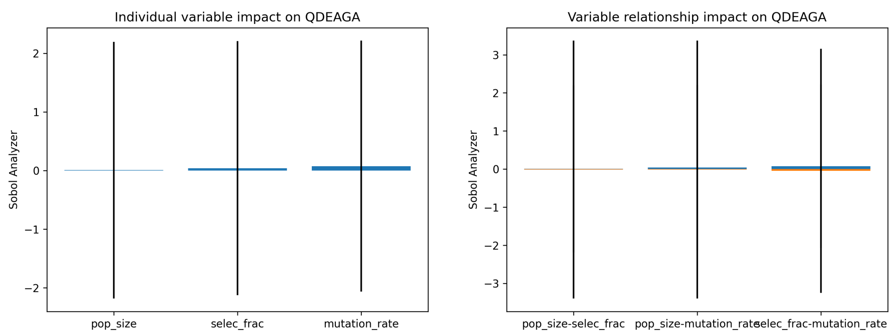
The experiments carried out reveal that quantum-inspired particle swarm optimization (QiPSO) is more effective and faster than the other techniques chosen in both the classical and quantum-inspired domains. Furthermore, sensitivity analysis has been conducted to select the appropriate parameter values, which enabled the techniques to converge faster in fewer iterations.

A comprehensive observation based on the experiments conducted is presented in Table 12. It has been observed that QiPSO is a more effective technique than others when it comes to portfolio optimization. Additionally, the quantum-inspired version



(a) Independent variable impact (b) Second order variable relationship

Fig. 11: Sensitivity analysis and variable impact in DE with heuristic crossover



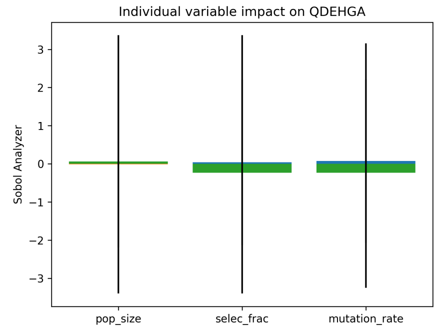
(a) Independent variable impact (b) Second order variable relationship

Fig. 12: Sensitivity analysis and variable impact in QiDE with arithmetic crossover

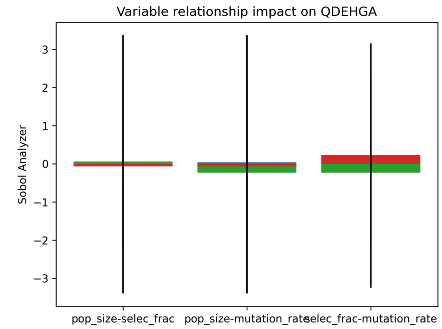
is dependable and can reach an optimal solution quicker than its classical counterpart without compromising the desired result (minimizing risk).

7.4 Statistical Tests

The Friedman chi-square test has been used to compare the mean risk of the evolutionary techniques chosen using the ”friedmanchisquare” module from the ”scipy.stats” package in Python. The test reveals that the significant level, p-value, is less than **0.0001**, allowing us to reject the null hypothesis. Consequently, it can be

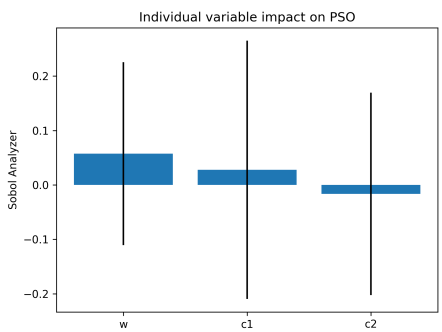


(a) Independent variable impact

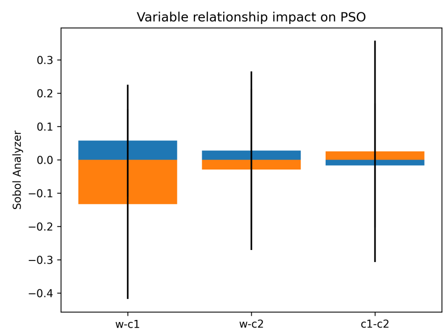


(b) Second order variable relationship

Fig. 13: Sensitivity analysis and variable impact in QiDE with heuristic crossover



(a) Independent variable impact



(b) Second order variable relationship

Fig. 14: Sensitivity analysis and variable impact in PSO

Table 12: Analysis of Results

Parameter	Observation
Risk	A quantum-inspired approach produces a more optimal risk than its classical counterpart.
Risk	QiPSO has the potential to generate the most optimal risk compared to other selected methods.
Error	Particle Swarm Optimization (PSO) is found to have the lowest mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). Therefore, it can be used for portfolio optimization problems.
Execution Time	Quantum-inspired versions can reach convergence faster than their classical counterparts, often with better results.
Convergence	Sensitivity analysis can be employed to reduce the amount of time it takes for convergence.

Table 13: Experimental Results with NASDAQ [5] data

Parameters	PSO	QiPSO	GA		QiGA		DEGA		QiDEGA	
			arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic
K = 2										
Risk	0.16993661	0.16969060	0.17705350	0.17237750	0.17532711	0.16992065	0.17594422	0.17139626	0.18342174	0.17652409
Return	0.18192847	0.18479456	0.20808422	0.16631283	0.19904282	0.17855352	0.20448473	0.16653309	0.21181152	0.15534185
MSE	0.00000939	0.00005573	0.00013482	0.00012547	0.00010463	0.00003413	0.00001398	0.00009188	0.00010546	0.00000938
MAE	0.00306447	0.00746518	0.01161108	0.01120147	0.01022910	0.00584200	0.00373904	0.00958541	0.01026938	0.00306314
RMSE	0.00433382	0.01055735	0.01642054	0.01584127	0.01446613	0.00826184	0.00528780	0.01355582	0.01452309	0.00433194
MAPE	0.01656991	0.04217254	0.05299886	0.07259424	0.04899653	0.03386394	0.01796263	0.06130259	0.04634070	0.02012350
MET (s)	31.933	4.999	30.795	15.278	30.135	9.950	30.156	15.323	30.354	15.263
TET (s)	63.866	9.997	61.59	30.555	60.270	19.901	60.312	30.647	60.707	30.527
K = 3										
Risk	0.16992334	0.16995519	0.17119414	0.17087693	0.18403938	0.17032572	0.18425073	0.17011897	0.17252434	0.17556135
Return	0.18100438	0.17995760	0.18746639	0.17414382	0.22161790	0.16399788	0.21933735	0.17566186	0.18510521	0.16621698
MSE	0.00004206	0.00000382	0.00039795	0.00002207	0.00004110	0.00008761	0.00003098	0.00000949	0.00043632	0.00013154
MAE	0.00611154	0.00180278	0.01875541	0.00441499	0.00603321	0.00879823	0.00469045	0.00281044	0.01969292	0.00968760
RMSE	0.01123309	0.00338495	0.03455212	0.00813706	0.01110364	0.01621211	0.00964080	0.00533702	0.03617960	0.01986499
MAPE	0.03253085	0.00993949	0.09081590	0.02618631	0.02643523	0.05764703	0.02074091	0.01631934	0.09611325	0.06436268
MET (s)	30.137	9.998	30.774	15.240	30.392	15.342	333.049	16.327	200.484	15.231
TET (s)	90.410	29.995	92.322	45.720	91.175	46.027	999.148	48.981	601.451	45.693
K = 4										
Risk	0.16995836	0.16982314	0.17627839	0.17114969	0.18359181	0.17009134	0.18667375	0.17117551	0.17950037	0.17134613
Return	0.18096377	0.18050074	0.20633540	0.17434984	0.21955100	0.17869638	0.22587454	0.17109051	0.21423827	0.16913689
MSE	0.00000958	0.00000394	0.00007213	0.00003121	0.00002574	0.00047627	0.00000775	0.00001802	0.00004911	0.00047317
MAE	0.00254276	0.00162109	0.00711309	0.00456201	0.00467241	0.02138047	0.00223214	0.00356867	0.00604265	0.01803456
RMSE	0.00618871	0.00397049	0.01698618	0.0111729	0.01014753	0.04364712	0.00556695	0.00849067	0.01401622	0.04350498
MAPE	0.01373867	0.00890953	0.03293238	0.02747092	0.0206670	0.13338176	0.00971364	0.02141996	0.02721078	0.10102173
MET (s)	30.175	10.989	30.196	12.651	24.051	12.192	30.009	15.219	265.688	15.679
TET (s)	120.702	43.957	120.783	50.604	96.204	48.768	120.038	60.878	1062.751	62.715
K = 5										
Risk	0.16976253	0.16914772	0.18410736	0.17049340	0.17452429	0.16922859	0.17435691	0.17021266	0.18927980	0.17417318
Return	0.18333495	0.17700309	0.22268964	0.18268224	0.19277681	0.17949698	0.17376899	0.17292770	0.23061484	0.19205464
MSE	0.00000827	0.00008718	0.00001237	0.00015114	0.00103984	0.00016033	0.00042309	0.00001409	0.00002304	0.00045396
MAE	0.00232903	0.00733088	0.00300757	0.01074279	0.02648564	0.00994461	0.01717683	0.00311604	0.00444242	0.01845545
RMSE	0.0064295	0.02087766	0.00786390	0.02749021	0.07210550	0.02831309	0.04599381	0.00839224	0.01073223	0.04764254
MAPE	0.01250981	0.04429320	0.01323483	0.06554203	0.15089645	0.06155401	0.08560473	0.01863193	0.01957884	0.10559275
MET (s)	598.165	8.786	30.845	13.279	32.517	16.292	32.424	16.45	32.981	16.879
TET (s)	2990.825	43.930	154.224	66.394	162.587	81.458	162.12	82.248	164.907	84.394

Table 14: Experimental Results with NASDAQ [5] data

Parameters	PSO	QiPSO	GA		QiGA		DEGA		QiDEGA	
			arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic
K = 6										
Risk	0.16991471	0.16948417	0.18330889	0.17057467	0.18873052	0.17235245	0.17392753	0.16939330	0.18629067	0.17308318
Return	0.18283825	0.17834630	0.22122277	0.16683851	0.22835212	0.15736639	0.18312229	0.17187760	0.22638376	0.15406684
MSE	0.00000718	0.00004037	0.00000974	0.00003332	0.00000321	0.00120167	0.00032398	0.00000804	0.00001422	0.00015219
MAE	0.00213137	0.00562028	0.0027089	0.00508367	0.00149663	0.03222602	0.01333486	0.00242933	0.00329960	0.01091220
RMSE	0.00656494	0.01556435	0.00764348	0.01413942	0.00438723	0.08491185	0.04408943	0.00694342	0.00923675	0.03021779
MAPE	0.01151816	0.03247763	0.01209981	0.03041851	0.00648008	0.17438457	0.06518694	0.01444670	0.01433773	0.06723716
MET (s)	63.670	9.743	32.178	12.553	31.712	15.889	31.826	14.991	31.951	16.81
TET (s)	382.017	58.459	193.071	75.317	190.271	95.333	190.958	89.945	191.703	100.857
K = 7										
Risk	0.16996138	0.16964126	0.17360405	0.17003687	0.17718959	0.16950878	0.17439554	0.16931103	0.18021038	0.1712571
Return	0.18139689	0.18017094	0.19628734	0.17584127	0.20640099	0.17533103	0.20233942	0.17084308	0.18451309	0.17230551
MSE	0.00002342	0.00001087	0.00019007	0.00004653	0.00008397	0.00003899	0.00029444	0.00016997	0.00027190	0.00033367
MAE	0.00361044	0.00288312	0.01254938	0.00638662	0.00652425	0.00502626	0.01409131	0.00917104	0.01206228	0.01370321
RMSE	0.01280417	0.00872233	0.03647584	0.01804664	0.02424445	0.01652129	0.04539879	0.03449316	0.04362694	0.04832918
MAPE	0.01918727	0.01596876	0.05903936	0.03831897	0.02970901	0.03047852	0.06764414	0.06133234	0.05814921	0.08439615
MET (s)	64.520	10.795	32.391	14.773	28.400	14.841	32.318	16.194	33.259	16.194
TET (s)	451.637	75.568	226.738	103.410	198.803	103.889	226.228	113.355	232.812	113.36
K = 8										
Risk	0.16975521	0.16908122	0.17126649	0.16925918	0.17362484	0.17140028	0.17768848	0.17055679	0.16958768	0.16966068
Return	0.18214731	0.17655875	0.19143338	0.17550782	0.15296998	0.17564184	0.19000911	0.17737303	0.18132045	0.17431752
MSE	0.00006124	0.00003176	0.00015085	0.00004933	0.00071351	0.00067460	0.00018003	0.00003884	0.00031053	0.00019024
MAE	0.0049918	0.00459684	0.00812461	0.00618130	0.01750553	0.02125720	0.01045553	0.00520109	0.01340355	0.01043391
RMSE	0.02213358	0.01594109	0.03473907	0.01986517	0.0755518	0.07346286	0.03795068	0.01762655	0.04984254	0.03901182
MAPE	0.02559813	0.02660622	0.03874855	0.03685256	0.09444352	0.12104584	0.04961305	0.03119509	0.06450095	0.06688881
MET (s)	64.572	8.940	31.767	14.598	31.865	16.055	29.145	16.045	32.581	14.704
TET (s)	516.573	71.520	254.140	116.781	254.918	128.442	233.159	128.363	260.648	117.634
K = 9										
Risk	0.16990022	0.16903544	0.17595410	0.16983567	0.18491184	0.17027057	0.17525003	0.16922948	0.16957274	0.17091427
Return	0.18250675	0.17380322	0.20816186	0.17853944	0.22380783	0.18950829	0.20159583	0.17409425	0.18673983	0.17068453
MSE	0.00007142	0.00000846	0.00012677	0.00009104	0.00001294	0.00048902	0.00007630	0.00007817	0.00047023	0.00009336
MAE	0.00517215	0.00260224	0.00920149	0.00701500	0.00298684	0.01913371	0.00558147	0.00723927	0.01959283	0.00765507
RMSE	0.02535376	0.00872745	0.03377771	0.02862388	0.01079290	0.06634157	0.02620436	0.02652415	0.06505450	0.02898613
MAPE	0.02639878	0.01474015	0.04291744	0.0436875	0.01301785	0.11549976	0.02582574	0.04494142	0.09644354	0.05020096
MET (s)	65.299	8.051	32.017	11.130	32.421	15.834	31.949	13.693	32.665	15.970
TET (s)	587.688	72.462	288.151	100.174	291.792	142.505	287.541	123.236	293.983	143.731

Table 15: Experimental Result with BSE [6] data

Parameters	PSO	QiPSO	GA		QiGA		DEGA		QiDEGA	
			arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic
K = 2										
Risk	0.28437094	0.28214914	0.28502824	0.28007045	0.28031957	0.28175554	0.27652596	0.28455587	0.28337837	0.28341821
Return	0.21842813	0.20962188	0.22669033	0.21338897	0.22601575	0.22833074	0.21605035	0.22813861	0.22638731	0.22857929
MSE	0.00008068	0.00002227	0.0000001	0.00005144	0.0000091	0.00000095	0.00003833	0.00000341	0.00000097	0.00000122
MAE	0.00898212	0.00471895	0.00032034	0.00717249	0.00301653	0.00097679	0.00619107	0.00184566	0.00098742	0.00110482
RMSE	0.01270263	0.00667361	0.00045303	0.01014343	0.00426602	0.00138139	0.0087555	0.00261015	0.00139642	0.00156246
MAPE	0.04296413	0.02304242	0.00141513	0.03255366	0.01317305	0.00429643	0.02787906	0.00815659	0.00434278	0.0048103
MET(s)	34.367	9.959	22.166	15.988	31.757	16.157	20.514	10.500	22.195	16.040
TET(s)	68.734	19.917	44.331	31.975	63.515	32.313	41.027	20.999	44.390	32.080
K = 3										
Risk	0.28294459	0.28856965	0.28031129	0.27950949	0.28031764	0.2815683	0.28300703	0.27978056	0.28515237	0.28178408
Return	0.20641171	0.20058753	0.22558222	0.2205529	0.22369842	0.22820138	0.22483935	0.22446383	0.22607401	0.22771045
MSE	0.0000081	0.00000017	0.00000194	0.00001133	0.00000871	0.00000072	0.00007754	0.00013349	0.00000028	0.00001752
MAE	0.00267817	0.00034037	0.00131182	0.00294205	0.00261448	0.00078261	0.00815847	0.01089099	0.00044767	0.00351825
RMSE	0.00493055	0.00071931	0.00241128	0.00583026	0.00511217	0.00146879	0.01525158	0.02001176	0.00092143	0.00724983
MAPE	0.01317084	0.00170117	0.0057763	0.0131145	0.01151857	0.003439	0.03749967	0.051559	0.00197436	0.01549997
MET(s)	31.670	16.783	31.855	16.155	24.411	16.266	26.921	15.554	27.754	15.613
TET(s)	95.010	50.350	95.566	48.464	73.233	48.798	80.764	46.661	83.262	46.839
K = 4										
Risk	0.2863223	0.2843913	0.27754847	0.27890665	0.28235542	0.28231772	0.2830106	0.27950385	0.28026566	0.28161895
Return	0.20375011	0.20736651	0.22045822	0.22393577	0.22784993	0.22977416	0.22675308	0.22360924	0.22669827	0.2238272
MSE	0.0000021	0.00001006	0.00003307	0.00010542	0.00037633	0.00000037	0.00001387	0.00000605	0.00002698	0.00001445
MAE	0.00125441	0.00311505	0.00537919	0.00877245	0.0191956	0.00052203	0.00315154	0.00243964	0.00446862	0.00363467
RMSE	0.00289753	0.00634369	0.01150197	0.02053486	0.03879838	0.00121098	0.00744757	0.00492041	0.01038754	0.00760175
MAPE	0.00620802	0.01530975	0.02419373	0.041126	0.09310855	0.00227848	0.01421814	0.01077447	0.02019375	0.01609477
MET(s)	30.686	12.030	30.512	15.233	30.633	15.562	26.065	11.998	23.112	13.512
TET(s)	122.743	48.121	122.046	60.931	122.533	62.247	104.260	47.993	92.448	54.047
K = 5										
Risk	0.28726834	0.28500637	0.27842196	0.28214208	0.28395524	0.28076799	0.28189607	0.27727102	0.28113982	0.28130639
Return	0.20121946	0.20476489	0.22348196	0.22114626	0.22865163	0.226003	0.2269111	0.21859512	0.22524474	0.22799192
MSE	0.0000011	0.0000023	0.00003643	0.00023328	0.00026869	0.00000396	0.00027392	0.00007394	0.0000072	0.00001776
MAE	0.00082143	0.00115672	0.00423554	0.01203782	0.01440459	0.00139788	0.01295917	0.0069089	0.00204974	0.00344998
RMSE	0.00234425	0.00339179	0.01349705	0.03415289	0.0366534	0.00444977	0.03700786	0.01922795	0.0059986	0.00942295
MAPE	0.00407438	0.00570775	0.0191558	0.05905266	0.06830938	0.00617019	0.06348708	0.03168621	0.00893239	0.01546503
MET(s)	30.738	11.026	30.542	15.694	29.097	220.874	30.878	15.488	28.247	12.198
TET(s)	153.691	55.131	152.71	78.471	145.484	1104.371	154.391	77.439	141.235	60.991

Table 16: Experimental Result with BSE [6] data

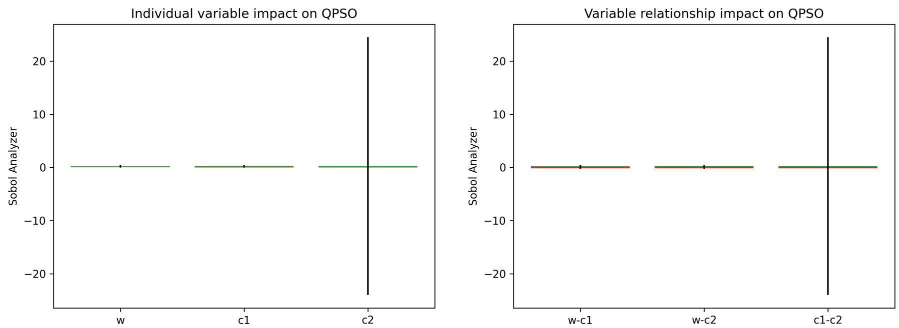
Parameters	PSO	QiPSO	GA		QiGA		DEGA		QiDEGA	
			arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic
K = 6										
Risk	0.28858718	0.28687543	0.27820896	0.27867707	0.28133062	0.28278287	0.28254121	0.28153882	0.27990647	0.28195452
Return	0.20031739	0.19895496	0.22187481	0.22411144	0.22771339	0.22945470	0.22776984	0.22197536	0.20318911	0.22439380
MSE	0.00000067	0.00000750	0.00000841	0.00000358	0.00000358	0.00000119	0.00005878	0.00006453	0.00010104	0.00016046
MAE	0.00073857	0.00228548	0.00236132	0.00177540	0.00172173	0.00094885	0.00593202	0.00599245	0.00888858	0.01182036
RMSE	0.00200392	0.00670673	0.00710148	0.00463368	0.00463680	0.00267637	0.01878012	0.01967763	0.02462251	0.03102850
MAPE	0.00366590	0.01125287	0.01042308	0.00783898	0.00749611	0.00412217	0.02700775	0.02761415	0.04097222	0.05532416
MET(s)	66.482	10.394	28.819	14.514	23.946	15.79	32.796	14.196	32.969	16.125
TET(s)	398.89	62.365	172.917	87.084	143.678	94.739	196.774	85.175	197.816	96.751
K = 7										
Risk	0.28858221	0.28412028	0.27762965	0.28034130	0.28003706	0.28288114	0.28018045	0.27745717	0.28216079	0.28051595
Return	0.20047752	0.19847003	0.22126969	0.22529056	0.22341762	0.22956759	0.21427785	0.22146709	0.22573199	0.22468565
MSE	0.00000405	0.00000525	0.00019632	0.00006983	0.00020046	0.00013191	0.00010532	0.00008572	0.00003483	0.00005798
MAE	0.00139795	0.00163364	0.01028021	0.00677093	0.00962692	0.00988118	0.00859009	0.00677250	0.00422604	0.00586555
RMSE	0.00532307	0.00606337	0.03707120	0.02210885	0.03745985	0.03038696	0.02715221	0.02449590	0.01561539	0.02014609
MAPE	0.00687927	0.00821589	0.05020690	0.03116995	0.04668386	0.04606596	0.04007327	0.03190185	0.01907450	0.02691879
MET(s)	65.85	7.895	30.131	15.118	30.824	16.301	31.281	14.192	29.853	14.413
TET(s)	460.95	55.265	210.919	105.828	215.767	114.109	218.97	99.345	208.968	100.891
K = 8										
Risk	0.28880686	0.28593086	0.27987376	0.28072145	0.27732614	0.28265701	0.27665700	0.28115766	0.27813660	0.28337148
Return	0.20046338	0.19859496	0.22306477	0.22618225	0.20903531	0.22995301	0.21065786	0.22801528	0.21957902	0.22996660
MSE	0.00002962	0.00000060	0.00001371	0.00002838	0.00014306	0.00001421	0.00017934	0.00002880	0.00010157	0.00012104
MAE	0.00416720	0.00064223	0.00278399	0.00340542	0.00948469	0.00295694	0.01018498	0.00350710	0.00778856	0.00715444
RMSE	0.01539310	0.00219716	0.01047466	0.0150682	0.03383026	0.01066314	0.03787818	0.01517774	0.02850579	0.03111728
MAPE	0.02005783	0.00321272	0.01249494	0.01547315	0.04507733	0.01307327	0.04976932	0.01600913	0.03625387	0.03399871
MET(s)	65.874	12.654	29.756	15.017	28.538	15.647	30.728	15.007	31.971	16.055
TET(s)	526.988	101.229	238.049	120.137	228.304	125.175	245.827	120.052	255.767	128.441
K = 9										
Risk	0.28831141	0.28642296	0.27834260	0.27720490	0.28090881	0.27766933	0.27880045	0.28082889	0.28041073	0.28111413
Return	0.20022604	0.20360647	0.22331308	0.21278182	0.22663396	0.22155397	0.22359693	0.22673572	0.22661164	0.22750071
MSE	0.00000347	0.00000283	0.00001180	0.00004199	0.00001095	0.00002063	0.00018131	0.00003290	0.00009006	0.00018945
MAE	0.00165371	0.00145753	0.00247509	0.00560189	0.00220016	0.00340838	0.00993773	0.00407589	0.00668846	0.01136856
RMSE	0.00558551	0.00504769	0.01030353	0.01943878	0.00992723	0.01362572	0.04039521	0.01720786	0.02847027	0.04129226
MAPE	0.00816825	0.00725027	0.01106082	0.02544708	0.00969532	0.0151024	0.04838399	0.01858024	0.03116014	0.05369207
MET(s)	64.275	11.633	23.926	13.734	29.133	15.424	25.03	14.186	31.682	14.982
TET(s)	578.477	104.693	215.333	123.608	262.196	138.814	225.27	127.677	285.142	134.839

Table 17: Experimental Result with Dow Jones [7] data

Parameters	PSO	QiPSO	GA		QiGA		DEGA		QiDEGA	
			arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic
K = 2										
Risk	0.21580509	0.21414730	0.21731237	0.21154595	0.21470904	0.22433182	0.20656942	0.21653964	0.24366822	0.21467436
Return	0.21067360	0.17490340	0.24675284	0.22640051	0.19608109	0.26248624	0.22974629	0.18104858	0.25240986	0.19043817
MSE	0.00005582	0.00042652	0.00013663	0.00017045	0.00138087	0.00179190	0.00000031	0.00001836	0.00069396	0.00005637
MAE	0.00747106	0.02065233	0.01168871	0.01305548	0.03716012	0.04233082	0.00056047	0.00428445	0.02634307	0.00750816
RMSE	0.01056567	0.02920680	0.01653033	0.01846324	0.05255234	0.05986482	0.00079263	0.00605913	0.03725473	0.01061814
MAPE	0.03428840	0.10679956	0.04532037	0.06142424	0.16346991	0.19965844	0.00244551	0.02425249	0.0953549	0.04111314
MET(s)	34.338	4.31	32.871	16.657	32.803	16.636	32.815	16.572	32.92	13.331
TET(s)	68.677	8.62	65.742	33.315	65.605	33.272	65.63	33.145	65.84	26.661
K = 3										
Risk	0.21584443	0.21446527	0.22436098	0.22122898	0.21924966	0.21101059	0.22127129	0.21751045	0.21345436	0.21843109
Return	0.21021667	0.21329588	0.26776076	0.25505281	0.25623291	0.23783886	0.26025268	0.20125380	0.23982554	0.23674723
MSE	0.00000349	0.00000348	0.00054342	0.00169714	0.00017848	0.00104977	0.00020546	0.00007802	0.00099907	0.00015861
MAE	0.00161561	0.00158926	0.02197548	0.03883484	0.01153428	0.02699837	0.01342831	0.00831418	0.02976754	0.01187002
RMSE	0.00323650	0.00322905	0.04037635	0.07135424	0.02313969	0.05611861	0.02482693	0.01529920	0.05474671	0.02181375
MAPE	0.00760804	0.00746625	0.09119471	0.19277143	0.04259392	0.11840726	0.05450664	0.04343664	0.11137801	0.04752320
MET(s)	32.842	10.896	32.646	16.595	33.257	12.725	32.845	16.685	32.838	16.837
TET(s)	98.526	32.688	97.939	49.786	99.771	38.175	98.534	50.054	98.514	50.51
K = 4										
Risk	0.21516017	0.21475539	0.21082758	0.21576844	0.23147512	0.22103216	0.21962149	0.21495296	0.23330602	0.21626876
Return	0.21109928	0.21264762	0.19912595	0.23274579	0.24333074	0.18406972	0.24883344	0.18391218	0.26976960	0.19027302
MSE	0.00000035	0.00010200	0.00061828	0.00120562	0.00025140	0.00220862	0.00003814	0.00008618	0.00007842	0.00188712
MAE	0.00048191	0.00873370	0.01956604	0.03346106	0.01360662	0.04681217	0.00500881	0.00781047	0.00780286	0.03734141
RMSE	0.00118536	0.02019879	0.04973030	0.06944403	0.03171129	0.09399186	0.01235148	0.01856618	0.01771058	0.08688208
MAPE	0.00228102	0.03920638	0.08553277	0.16190318	0.05223350	0.21660520	0.01952909	0.04054741	0.02774528	0.16973217
MET(s)	32.879	8.606	32.803	16.625	32.911	16.617	72.551	16.877	33.339	17.02
TET(s)	131.515	34.425	131.213	66.499	131.643	66.466	290.204	67.509	133.354	68.08
K = 5										
Risk	0.21334289	0.21427521	0.21902340	0.21683257	0.21365846	0.21691667	0.22182740	0.21256873	0.22095311	0.21619809
Return	0.21294028	0.20064236	0.23218183	0.19860714	0.25110920	0.24410681	0.26172948	0.22045975	0.23028033	0.17892020
MSE	0.00000284	0.00009188	0.00109713	0.00001091	0.00062965	0.00182453	0.00028249	0.00024397	0.00079182	0.00262586
MAE	0.00120587	0.00718080	0.02836220	0.00240921	0.02201402	0.03595928	0.01246015	0.01221302	0.02719447	0.04995252
RMSE	0.00377063	0.02143369	0.07406501	0.00738413	0.05610941	0.09551269	0.03758226	0.03492654	0.06292121	0.11458321
MAPE	0.00564146	0.03325830	0.12620089	0.01244983	0.08364845	0.16838211	0.0498459	0.06153381	0.10853418	0.23680290
MET(s)	33.467	8.347	33.615	15.062	29.323	16.844	33.204	16.849	32.935	16.676
TET(s)	167.336	41.736	168.074	75.309	146.617	84.222	166.018	84.244	164.673	83.38

Table 18: Experimental Result with Dow Jones [7] data

Parameters	PSO	QiPSO	GA		QiGA		DEGA		QiDEGA	
			arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic	arithmetic	heuristic
K = 6										
Risk	0.21553352	0.20676807	0.21337842	0.20522564	0.21853725	0.21472337	0.21961358	0.21460723	0.20701537	0.20359223
Return	0.21195819	0.18475193	0.24149855	0.19707729	0.25791628	0.16092131	0.25647989	0.23312889	0.20155022	0.18956903
MSE	0.00000047	0.00011790	0.00008856	0.00090566	0.00013113	0.00208214	0.00062051	0.00071866	0.00111208	0.00220567
MAE	0.00060160	0.00743801	0.00852630	0.02689840	0.01023778	0.04328281	0.01792030	0.02416442	0.02374258	0.04351820
RMSE	0.00168197	0.02659688	0.02305158	0.07371539	0.02804928	0.11177147	0.06101678	0.06566572	0.08168540	0.11503917
MAPE	0.00285122	0.0373212	0.033594	0.12244377	0.03855296	0.21127028	0.07782068	0.11515180	0.09914482	0.20689379
MET(s)	69.273	11.861	35.041	17.129	34.652	17.201	35.398	17.238	34.569	16.303
TET(s)	415.637	71.167	210.245	102.772	207.913	103.207	212.385	103.428	207.414	97.818
K = 7										
Risk	0.21483613	0.20937639	0.21989691	0.20882962	0.21600184	0.20621107	0.21855169	0.21161935	0.21656244	0.21347471
Return	0.21398046	0.20685696	0.25986377	0.22272233	0.21806962	0.21965715	0.25246505	0.19085884	0.25383838	0.21850531
MSE	0.00000243	0.00002930	0.00026684	0.00137714	0.00086205	0.00119673	0.00025677	0.00064112	0.00101550	0.00140685
MAE	0.00137774	0.00372467	0.01453451	0.03483443	0.02765230	0.03084700	0.01153887	0.02420530	0.02785099	0.03338679
RMSE	0.00412373	0.01432221	0.04321893	0.09818356	0.07768121	0.09152649	0.04239593	0.06699119	0.08431200	0.09923693
MAPE	0.00646964	0.01723462	0.05766779	0.16930816	0.11075308	0.14911858	0.04793963	0.11539668	0.10851429	0.16011522
MET(s)	68.861	13.322	35.169	17.405	34.74	17.285	34.782	17.153	34.76	17.25
TET(s)	482.025	93.253	246.186	121.836	243.177	120.997	243.477	120.071	243.317	120.751
K = 8										
Risk	0.21563058	0.21445588	0.21830920	0.20873599	0.21452374	0.20861943	0.21767365	0.21606888	0.20921626	0.21615418
Return	0.21097903	0.20691119	0.21472663	0.19247558	0.20159066	0.21375502	0.23918568	0.20006919	0.23707271	0.17214498
MSE	0.00000199	0.00000704	0.00045107	0.00034778	0.00113008	0.00154352	0.00053527	0.00009416	0.00044521	0.00101309
MAE	0.00103760	0.00198954	0.01972889	0.01231037	0.02353037	0.03645390	0.01910961	0.00814117	0.01911326	0.02056911
RMSE	0.00399430	0.00750705	0.06007153	0.05274674	0.09508211	0.11112227	0.06543844	0.02744647	0.05967948	0.09002625
MAPE	0.00491222	0.00952996	0.08342115	0.06134460	0.09374677	0.16345183	0.07182127	0.04350093	0.07188841	0.09731884
MET(s)	70.452	12.097	34.022	16.767	35.734	15.272	35.855	17.902	35.243	17.464
TET(s)	563.617	96.778	272.175	134.135	285.87	122.176	286.842	143.216	281.946	139.714
K = 9										
Risk	0.21593190	0.21341804	0.20955251	0.21034961	0.21934443	0.20660075	0.21409116	0.21161358	0.21736852	0.20891785
Return	0.21146489	0.22008195	0.18474028	0.18319306	0.25941766	0.17733199	0.23027939	0.24461605	0.24529067	0.20281033
MSE	0.00000542	0.00002736	0.00095265	0.00094119	0.00016439	0.00070151	0.00051979	0.00083597	0.00031684	0.00169038
MAE	0.00186734	0.00354855	0.02706253	0.02776421	0.01101301	0.01974947	0.02063436	0.02685351	0.01406657	0.03638248
RMSE	0.00698623	0.01569273	0.09259499	0.09203635	0.03846417	0.07945822	0.06839682	0.08673936	0.05340024	0.12334267
MAPE	0.0087419	0.01690759	0.12503001	0.13099201	0.03968041	0.09435329	0.08336806	0.12890845	0.05496648	0.16431488
MET(s)	70.133	12.608	32.284	17.872	33.17	15.982	33.947	16.276	33.155	16.332
TET(s)	631.197	113.474	290.555	160.846	298.534	143.842	305.522	146.486	298.393	146.989



(a) Independent variable impact (b) Second order variable relationship

Fig. 15: Sensitivity analysis and variable impact in QiPSO

concluded that there are significant differences between the groups.

7.5 Time Complexity

The time complexity of the genetic algorithm (GA) and differential evolution (DE) depends on factors such as population size (N), number of stocks (S), number of generations (G), and the cost of the fitness function. The overall execution can be broken down into steps, viz.,

- Initialization: The amount of time needed for initialization depends on the size of the population (N) and the number of stocks (S) taken into account. Thus, the time complexity of initialization can be given as $O(N \times S)$.
- Fitness Function: This experiment uses a fitness function that sorts the population according to the Sharpe ratio. Thus, the time complexity of the fitness function can be given as $O(N \times logN)$.
- Selection: The selection process in this experiment performs two operations, viz., (i) sorting the population, and (ii) generating a sample population. The time complexity of the sorting function and the generation of the sample population are given as $O(N \times logN)$ and $O(N_s \times S)$, respectively, where N_s is the size of the sample population to be generated and S is the number of stocks considered.
- Crossover: The time required to complete this process depends on the size of the subset of the population (N_s) and the number of stocks (S) taken into account. Therefore, the time complexity of the crossover can be expressed as $O(N_s \times S)$.
- Mutation: The mutation perturbs the generation of new children based on the mutation rate. This is done to introduce more diversity into the population. Therefore, the time complexity of mutation can be expressed as $O(N_m \times S)$,

where N_m is the size of the sample to be generated and S is the number of stocks considered. Generally, the mutation rates chosen are very small and so is the time complexity. In a practical sense, the time complexity of mutation can be ignored.

The time complexity of a particle swarm optimization (PSO) depends on factors such as population size (N), number of stocks (S), and number of generations (G). The overall execution can be broken down into steps, viz.,

- Initialization: The initialization process in PSO is the same as that of GA and DE. Therefore, the time complexity of initialization is given as $O(N \times S)$.
- Fitness function: The fitness function used in PSO is the same as that of GA and DE. Therefore, the time complexity of the fitness function is given as $O(N \times logN)$.
- Particle movements: In PSO, two types of updates are performed, viz. (i) position update and (ii) velocity update. The time complexity to update both the population and velocity is $O(N \times S)$, where N is the number of individuals in the population and S is the number of stocks taken into account.

Based on the time complexity analysis, it is evident that PSO is superior in terms of execution time compared to both GA and DE. PSO is also found to converge faster than both GA and DE. Between GA and DE, it has been observed that DE is more likely to converge faster than GA due to its superior exploration capabilities.

8 Discussions and Conclusion

This work presents GA, DE, and PSO-based techniques for portfolio optimization problems. The article presents four different enhancements to these techniques, such as (i) dynamic selection of crossover parameters α and β (ii) normalization function to avoid negative allocation caused due to incorrect crossover parameter selection (iii) regularization function for better utilization of allocated funds and (iv) dynamic selection of optimization parameters based on sensitivity analysis and regularization function. Efforts have been made to improve the classical techniques by introducing quantum-inspired versions. Experiments have demonstrated that these quantum-inspired versions are faster, and the results are comparable or even better than their classical counterparts. In particular, quantum-inspired PSO outperforms all the techniques chosen. Generally, quantum-inspired techniques have been observed to be faster and often converge to an optimal result. The experiments have been carried out multiple times on three datasets, viz. (i) NASDAQ, (ii) BSE, and (iii) Dow Jones. To further optimize the selected techniques, the population size and other optimization parameters were chosen based on sensitivity analysis.

In the future, the mentioned techniques can be extended to the multiple objective domains to optimize both risk and return on the selected benchmarked datasets. The current experiment has considered the Sharpe ratio as the optimization measure, and it would be interesting to see the impact of other measures on the portfolio optimization problem. The authors are currently exploring the use of group-based evolutionary techniques to identify the most cost-effective number of stocks to purchase and sell in situations that demand the management of a large number of stocks. This could potentially reduce the transaction costs associated with stock trading.

9 Declarations

9.1 Funding and/or Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

9.2 Research involving Human Participants and/or Animals

No human participants and/or animals were involved in this study.

9.3 Informed consent

All authors consent to submit the article for consideration.

References

1. Martin Cherkes. Closed-end funds: A survey. *Annu. Rev. Financ. Econ.*, 4(1):431–445, 2012.
2. Itzhak Ben-David, Francesco Franzoni, and Rabih Moussawi. Exchange-traded funds. *Annual Review of Financial Economics*, 9:169–189, 2017.
3. Colin Atkinson, Stanley R Pliska, and Paul Wilmott. Portfolio management with transaction costs. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 453(1958):551–562, 1997.
4. Renata Mansini, Włodzimierz Ogryczak, and M Grazia Speranza. Portfolio optimization with transaction costs. In *Linear and Mixed Integer Programming for Portfolio Optimization*, pages 47–62. Springer, 2015.
5. NASDAQ. Historical data from nasdaq, 2012.
6. BSE. Historical data from bombay stock exchange, 2013.
7. Dow Jones. Historical data from dow jones, 2009.
8. Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.
9. DE Goldberg and JH Holland. Genetic algorithms and machine learning. 3 (2): 95-99, 1988.
10. John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

11. Kit Po Wong and Zhao Yang Dong. Differential evolution, an alternative approach to evolutionary algorithm. In *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, pages 73–83. IEEE, 2005.
12. Karol R Opara and Jarosław Arabas. Differential evolution: A survey of theoretical analyses. *Swarm and evolutionary computation*, 44:546–558, 2019.
13. James Kennedy and Russell Eberhart. Particle swarm optimization in: Neural networks. In *Proceedings IEEE International Conference on 1995*, pages 1942–1948, 1942.
14. James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.
15. Martin B Haugh and Andrew W Lo. Computational challenges in portfolio management. *Computing in Science & Engineering*, 3(3):54–59, 2001.
16. Margaret Peteraf and Randal Reed. Managerial discretion and internal alignment under regulatory constraints and change. *Strategic Management Journal*, 28(11):1089–112, 2007.
17. Ilya RP Cuypers, Jean-François Hennart, Brian S Silverman, and Gokhan Ertug. Transaction cost theory: Past progress, current challenges, and suggestions for the future. *Academy of Management Annals*, 15(1):111–150, 2021.
18. J-P Onnela, Anirban Chakraborti, Kimmo Kaski, Janos Kertesz, and Antti Kanto. Dynamics of market correlations: Taxonomy and portfolio analysis. *Physical Review E*, 68(5):056110, 2003.
19. SN Sivanandam, SN Deepa, SN Sivanandam, and SN Deepa. *Genetic algorithms*. Springer, 2008.
20. Paolo Aversa, Stefan Haefliger, and Danielle Giuliana Reza. Building a winning business model portfolio. *MIT Sloan Management Review*, 58(4):49–54, 2017.
21. Abhishek Gunjan and Siddhartha Bhattacharyya. A brief review of portfolio optimization techniques. *Artificial Intelligence Review*, pages 1–40, 2022.
22. Suraj S Meghwani and Manoj Thakur. Multi-criteria algorithms for portfolio optimization under practical constraints. *Swarm and evolutionary computation*, 37:104–125, 2017.
23. Ankita Golchha and Shahana Gajala Qureshi. Non-dominated sorting genetic algorithm-ii–a succinct survey. *International Journal of Computer Science and Information Technologies*, 6(1):252–255, 2015.
24. Eckart Zitzler and Lothar Thiele. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. *TIK report*, 43, 1998.
25. Rubén Saborido, Ana B Ruiz, and Mariano Luque. Global wasf-ga: An evolutionary algorithm in multiobjective optimization to approximate the whole pareto optimal front. *Evolutionary computation*, 25(2):309–349, 2017.
26. David W Corne, Joshua D Knowles, and Martin J Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In *International conference on parallel problem solving from nature*, pages 839–848. Springer, 2000.
27. Yi Chen, Aimin Zhou, and Swagatam Das. Utilizing dependence among variables in evolutionary algorithms for mixed-integer programming: A case study on multi-objective constrained portfolio optimization. *Swarm and Evolutionary Computation*, 66:100928, 2021.
28. Laura Diosan. A multi-objective evolutionary approach to the portfolio optimization problem. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 183–187. IEEE, 2005.
29. Prisadarng Skolpadungket, Keshav Dahal, and Napat Harnpornchai. Portfolio optimization using multi-obj ective genetic algorithms. In *2007 IEEE Congress on Evolutionary Computation*, pages 516–523. IEEE, 2007.
30. Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Citeseer, 2005.
31. Xin-She Yang. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pages 169–178. Springer, 2009.
32. Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

33. Kyle Erwin and Andries Engelbrecht. Meta–heuristics for portfolio optimization: Part i—review of meta–heuristics. In *International Conference on Swarm Intelligence*, pages 441–452. Springer, 2023.
34. Kyle Erwin and Andries Engelbrecht. Meta–heuristics for portfolio optimization: Part i—review of meta–heuristics. In *International Conference on Swarm Intelligence*, pages 441–452. Springer, 2023.
35. Biplab Mahapatra. Investigation of portfolio optimization with realistic constraints in indian equity market using random immigration genetic algorithm. 2023.
36. Massimiliano Kaucic, Mojtaba Moradi, and Mohmmad Mirzazadeh. Portfolio optimization by improved nsga-ii and spea 2 based on different risk measures. *Financial Innovation*, 5:1–28, 2019.
37. Khin Lwin, Rong Qu, and Graham Kendall. A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, 24:757–772, 2014.
38. Yan Chen, Shingo Mabu, and Kotaro Hirasawa. Genetic relation algorithm with guided mutation for the large-scale portfolio optimization. *Expert Systems with Applications*, 38(4):3353–3363, 2011.
39. Khin T Lwin, Rong Qu, and Bart L MacCarthy. Mean-var portfolio optimization: A nonparametric approach. *European Journal of operational research*, 260(2):751–766, 2017.
40. Khin Thein Lwin. *Evolutionary approaches for portfolio optimization*. PhD thesis, University of Nottingham, 2015.
41. Konstantinos P Anagnostopoulos and Georgios Mamanis. A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research*, 37(7):1285–1297, 2010.
42. Swee Chiang Chiam, Kay Chen Tan, and A Al Mamum. Evolutionary multi-objective portfolio optimization in practical context. *International Journal of Automation and Computing*, 5:67–80, 2008.
43. Iryna Yevseyeva, Andreia P Guerreiro, Michael TM Emmerich, and Carlos M Fonseca. A portfolio optimization approach to selection in multiobjective evolutionary algorithms. In *Parallel Problem Solving from Nature–PPSN XIII: 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings 13*, pages 672–681. Springer, 2014.
44. Konstantinos P Anagnostopoulos and Georgios Mamanis. The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications*, 38(11):14208–14217, 2011.
45. Ankit Thakkar and Kinjal Chaudhari. A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization. *Archives of Computational Methods in Engineering*, 28:2133–2164, 2021.
46. Pankaj K Bharne and Sameer S Prabhune. Survey on combined swarm intelligence and ann for optimized daily stock market price. In *2017 International Conference on Soft Computing and its Engineering Applications (icSoftComp)*, pages 1–6. IEEE, 2017.
47. Jaime G Carbonell, Ryszard S Michalski, and Tom M Mitchell. An overview of machine learning. *Machine learning*, pages 3–23, 1983.
48. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
49. Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep learning for portfolio optimization. *The Journal of Financial Data Science*, 2020.
50. Peter Harrington. *Machine learning in action*. Simon and Schuster, 2012.
51. Yilin Ma, Ruizhu Han, and Weizhong Wang. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165:113973, 2021.
52. Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25:197–227, 2016.
53. Shindi Shella May Wara, Dedy Dwi Prastyo, and Heri Kuswanto. Value at risk estimation with hybrid-svr-garch-kde model for lq45 portfolio optimization. In *AIP Conference Proceedings*, volume 2540. AIP Publishing, 2023.

54. Hieu K Cao, Han K Cao, and Binh T Nguyen. Delafo: An efficient portfolio optimization using deep neural networks. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I 24*, pages 623–635. Springer, 2020.
55. Jaydip Sen, Abhishek Dutta, and Sidra Mehtab. Stock portfolio optimization using a deep learning lstm model. In *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, pages 263–271. IEEE, 2021.
56. Van-Dai Ta, CHUAN-MING Liu, and Direselign Addis Tadesse. Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences*, 10(2):437, 2020.
57. Yilin Ma, Ruizhu Han, and Weizhong Wang. Prediction-based portfolio optimization models using deep neural networks. *Ieee Access*, 8:115393–115405, 2020.
58. Zeng Wei Zheng, Yuan Yi Chen, Xiao Wei Zhou, Mei Mei Huo, Bo Zhao, and Min Yi Guo. Short-term prediction model for a grid-connected photovoltaic system using emd and gabpnn. *Applied Mechanics and Materials*, 291:74–82, 2013.
59. KIM Hongjoong. Mean-variance portfolio optimization with stock return prediction using xgboost. *Economic Computation & Economic Cybernetics Studies & Research*, 55(4), 2021.
60. Liangyu Min, Jiawei Dong, Jiangwei Liu, and Xiaomin Gong. Robust mean-risk portfolio optimization using machine learning-based trade-off parameter. *Applied Soft Computing*, 113:107948, 2021.
61. NR Rohith, MD Yashaswini, VK Rohini, and HG Sinchana. Quantum computing: Redefining the future of finance in portfolio optimization. *Eur. Chem. Bull.*, 12(6):1125–1129, 2023.
62. Roman Orus, Samuel Mugel, and Enrique Lizaso. Quantum computing for finance: Overview and prospects. *Reviews in Physics*, 4:100028, 2019.
63. Dylan Herman, Cody Googin, Xiaoyuan Liu, Alexey Galda, Ilya Safro, Yue Sun, Marco Pistoia, and Yuri Alexeev. A survey of quantum computing for finance. *arXiv preprint arXiv:2201.02773*, 2022.
64. Abha Naik, Esra Yeniaras, Gerhard Hellstern, Grishma Prasad, and Sanjay Kumar Lalta Prasad Vishwakarma. From portfolio optimization to quantum blockchain and security: A systematic review of quantum computing in finance. *arXiv preprint arXiv:2307.01155*, 2023.
65. Jeffrey Cohen, Alex Khan, and Clark Alexander. Portfolio optimization of 40 stocks using the dwave quantum annealer. *arXiv preprint arXiv:2007.01430*, 2020.
66. Erica Grant, Travis S Humble, and Benjamin Stump. Benchmarking quantum annealing controls with portfolio optimization. *Physical Review Applied*, 15(1):014012, 2021.
67. Davide Venturelli and Alexei Kondratyev. Reverse quantum annealing approach to portfolio optimization problems. *Quantum Machine Intelligence*, 1(1-2):17–30, 2019.
68. Kosuke Tatsumura, Ryo Hidaka, Jun Nakayama, Tomoya Kashimata, and Masaya Yamasaki. Real-time trading system based on selections of potentially profitable, uncorrelated, and balanced stocks by np-hard combinatorial optimization. *arXiv preprint arXiv:2307.06339*, 2023.
69. Lammertjan Dam and Bert Scholtens. Toward a theory of responsible investing: On the economic foundations of corporate social responsibility. *Resource and Energy Economics*, 41:103–121, 2015.
70. Li Chen, Lipei Zhang, Jun Huang, Helu Xiao, and Zhongbao Zhou. Social responsibility portfolio optimization incorporating esg criteria. *Journal of Management Science and Engineering*, 6(1):75–85, 2021.
71. Russell Sparkes. *Socially responsible investment: A global revolution*. John Wiley & Sons, 2003.
72. Francesco Cesarone, Lorenzo Lampariello, Davide Merolla, Jacopo Maria Ricci, Simone Sagratella, and Valerio Giuseppe Sasso. A bilevel approach to esg multi-portfolio selection. *Computational Management Science*, 20(1):24, 2023.
73. Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003.

74. Thomas Bartz-Beielstein, Jürgen Branke, Jörn Mehnen, and Olaf Mersmann. Evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(3):178–195, 2014.
75. Thomas Bäck and Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, 1(1):1–23, 1993.
76. Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation*, 3(2):82–102, 1999.
77. David B Fogel and Lawrence J Fogel. An introduction to evolutionary programming. In *Artificial Evolution: European Conference, AE 95 Brest, France, September 4–6, 1995 Selected Papers*, pages 21–33. Springer, 2005.
78. Melanie Mitchell, JH Holland, and S Forrest. The royal road for genetic algorithms: Fitness landscapes and ga performance. Technical report, Los Alamos National Lab., NM (United States), 1991.
79. Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media, 2012.
80. Hans-Georg Beyer. *The theory of evolution strategies*. Springer Science & Business Media, 2001.
81. John R Koza. *Genetic programming II: automatic discovery of reusable programs*. MIT press, 1994.
82. Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.
83. Maryam Alavi and John C Henderson. An evolutionary strategy for implementing a decision support system. *Management science*, 27(11):1309–1323, 1981.
84. Ryan J Urbanowicz and Jason H Moore. Learning classifier systems: a complete introduction, review, and roadmap. *Journal of Artificial Evolution and Applications*, 2009, 2009.
85. Mark Hauschild and Martin Pelikan. An introduction and survey of estimation of distribution algorithms. *Swarm and evolutionary computation*, 1(3):111–128, 2011.
86. James Kennedy. *Swarm intelligence*. Springer, 2006.
87. Marco Dorigo and Thomas Stützle. *Ant colony optimization: overview and recent advances*. Springer, 2019.
88. David Ardia, Kris Boudt, Peter Carl, Katharine Mullen, and Brian G Peterson. Differential evolution with deoptim: an application to non-convex portfolio optimization. *The R Journal*, 3(1):27–34, 2011.
89. Yingjie Song, Gaoyang Zhao, Bin Zhang, Huayue Chen, Wuquan Deng, and Wu Deng. An enhanced distributed differential evolution algorithm for portfolio optimization problems. *Engineering Applications of Artificial Intelligence*, 121:106004, 2023.
90. Sheldon Hui and Ponnuthurai N Suganthan. Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization. *IEEE transactions on cybernetics*, 46(1):64–74, 2015.
91. Slimane Sefiane and Mohamed Benbouziane. Portfolio selection using genetic algorithm. 2012.
92. Joseph Ackora-Prah, Samuel Asante Gyamerah, and Perpetual Saah Andam. A heuristic crossover for portfolio selection. 2014.
93. Padmavathi Kora and Priyanka Yadlapalli. Crossover operators in genetic algorithms: A review. *International Journal of Computer Applications*, 162(10), 2017.
94. Anant J Umbarkar and Pranali D Sheth. Crossover operators in genetic algorithms: a review. *ICTACT journal on soft computing*, 6(1), 2015.
95. Mohammed A El-Shorbagy and Aboul Ella Hassanien. Particle swarm optimization from theory to applications. *International Journal of Rough Sets and Data Analysis (IJRSDA)*, 5(2):1–24, 2018.
96. Tunchan Cura. Particle swarm optimization approach to portfolio optimization. *Nonlinear analysis: Real world applications*, 10(4):2396–2406, 2009.

97. Majid Jaberipour, Esmaile Khorram, and Behrooz Karimi. Particle swarm algorithm for solving systems of nonlinear equations. *Computers & Mathematics with Applications*, 62(2):566–576, 2011.
98. Rodolfo C Cavalcante, Rodrigo C Brasileiro, Victor LF Souza, Jarley P Nobrega, and Adriano LI Oliveira. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 55:194–211, 2016.
99. Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.
100. Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.
101. Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39:459–471, 2007.
102. Michalis Mavrovouniotis, Changhe Li, and Shengxiang Yang. A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm and Evolutionary Computation*, 33:1–17, 2017.
103. Kyle Erwin and Andries Engelbrecht. Improved set-based particle swarm optimization for portfolio optimization. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1573–1580. IEEE, 2020.
104. Yi Da and Ge Xiurun. An improved pso-based ann with simulated annealing technique. *Neurocomputing*, 63:527–533, 2005.
105. Yves Crama and Michaël Schyns. Simulated annealing for complex portfolio selection problems. *European Journal of operational research*, 150(3):546–571, 2003.
106. Saeid Qodsi, Reza Tehrani, and Mahdi Bashiri. Portfolio optimization with simulated annealing algorithm. *Financial Research Journal*, 17(1):141–158, 2015.
107. Mohammad Maholi Solin, Andry Alamsyah, Brady Rikumahu, and Muhammad Apriandito Arya Saputra. Forecasting portfolio optimization using artificial neural network and genetic algorithm. In *2019 7th International Conference on Information and Communication Technology (ICoICT)*, pages 1–7. IEEE, 2019.
108. Yong Wang and Zixing Cai. A hybrid multi-swarm particle swarm optimization to solve constrained optimization problems. *Frontiers of Computer Science in China*, 3:38–52, 2009.
109. Aamir Nawaz, Ehtasham Mustafa, Nasir Saleem, Muhammad Irfan Khattak, Muhammad Shafi, and Abdul Malik. Solving convex and non-convex static and dynamic economic dispatch problems using hybrid particle multi-swarm optimization. *Tehnički vjesnik*, 24(4):1095–1102, 2017.
110. David McMahon. *Quantum computing explained*. John Wiley & Sons, 2007.
111. Nicholas Young. *An introduction to Hilbert space*. Cambridge university press, 1988.
112. Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.
113. Dorit Aharonov, Alexei Kitaev, and Noam Nisan. Quantum circuits with mixed states. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 20–30, 1998.
114. Matthew PA Fisher, Vedika Khemani, Adam Nahum, and Sagar Vijay. Random quantum circuits. *Annual Review of Condensed Matter Physics*, 14:335–379, 2023.
115. Dan J Shepherd. On the role of hadamard gates in quantum circuits. *Quantum Information Processing*, 5:161–177, 2006.
116. Roderich Tumulka. Dirac notation. In *Compendium of Quantum Physics*, pages 172–174. Springer, 2009.
117. David P DiVincenzo. Quantum gates and circuits. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):261–276, 1998.
118. Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE transactions on evolutionary computation*, 6(6):580–593, 2002.

119. Swagata Karmakar, Ashmita Dey, and Indrajit Saha. Use of quantum-inspired metaheuristics during last two decades. In *2017 7th International Conference on Communication Systems and Network Technologies (CSNT)*, pages 272–278. IEEE, 2017.
120. Christos Papalitsas, Panayiotis Karakostas, and Kalliopi Kastampolidou. A quantum inspired gvns: Some preliminary results. In *GeNeDis 2016: Computational Biology and Bioinformatics*, pages 281–289. Springer, 2017.
121. Farhad Soleimanian Gharehchopogh. Quantum-inspired metaheuristic algorithms: Comprehensive survey and classification. *Artificial Intelligence Review*, 56(6):5479–5543, 2023.
122. Patrick Rebentrost and Seth Lloyd. Quantum computational finance: quantum algorithm for portfolio optimization. *arXiv preprint arXiv:1811.03975*, 2018.
123. Shahin Hakemi, Mahboobeh Houshmand, Esmaeil KheirKhah, and Seyyed Abed Hosseini. A review of recent advances in quantum-inspired metaheuristics. *Evolutionary Intelligence*, pages 1–16, 2022.
124. Lawrence Davis. *Handbook of genetic algorithms*. CumInCAD, 1991.
125. Fred Glover and Manuel Laguna. Tabu search. In *Handbook of combinatorial optimization*, pages 2093–2229. Springer, 1998.
126. Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.
127. Ajit Narayanan and Mark Moore. Quantum-inspired genetic algorithms. In *Proceedings of IEEE international conference on evolutionary computation*, pages 61–66. IEEE, 1996.
128. Kuk-Hyun Han, Kui-Hong Park, Ci-Ho Lee, and Jong-Hwan Kim. Parallel quantum-inspired genetic algorithm for combinatorial optimization problem. In *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, volume 2, pages 1422–1429. IEEE, 2001.
129. Hatem MH Saad, Ripon K Chakrabortty, Saber Elsayed, and Michael J Ryan. Quantum-inspired genetic algorithm for resource-constrained project-scheduling. *IEEE Access*, 9:38488–38502, 2021.
130. Hua-Pei Chiang, Yao-Hsin Chou, Chia-Hui Chiu, Shu-Yu Kuo, and Yueh-Min Huang. A quantum-inspired tabu search algorithm for solving combinatorial optimization problems. *Soft Computing*, 18(9):1771–1781, 2014.
131. Yao-Hsin Chou, Shu-Yu Kuo, Chi-Yuan Chen, and Han-Chieh Chao. A rule-based dynamic decision-making stock trading system based on quantum-inspired tabu search algorithm. *IEEE Access*, 2:883–896, 2014.
132. Shu-Yu Kuo and Yao-Hsin Chou. Entanglement-enhanced quantum-inspired tabu search algorithm for function optimization. *IEEE Access*, 5:13236–13252, 2017.
133. Wu Deng, Hailong Liu, Junjie Xu, Huimin Zhao, and Yingjie Song. An improved quantum-inspired differential evolution algorithm for deep belief network. *IEEE Transactions on Instrumentation and Measurement*, 69(10):7319–7327, 2020.
134. Manuel S Alvarez-Alvarado, Francisco E Alban-Chacón, Erick A Lamilla-Rubio, Carlos D Rodríguez-Gallegos, and Washington Velásquez. Three novel quantum-inspired swarm optimization algorithms using different bounded potential fields. *Scientific Reports*, 11(1):1–22, 2021.
135. RK Agrawal, Baljeet Kaur, and Parul Agarwal. Quantum inspired particle swarm optimization with guided exploration for function optimization. *Applied Soft Computing*, 102:107122, 2021.
136. Sandip Dey, Siddhartha Bhattacharyya, and Ujjwal Maulik. Quantum-inspired automatic clustering technique using ant colony optimization algorithm. In *Quantum-Inspired Intelligent Systems for Multimedia Data Analysis*, pages 27–54. IGI Global, 2018.
137. Sayed A Mohsin, Saad Mohamed Darwish, and Ahmed Younes. Qiaco: A quantum dynamic cost ant system for query optimization in distributed database. *IEEE Access*, 9:15833–15846, 2021.
138. A Manju and Madhav J Nigam. Applications of quantum inspired computational intelligence: a survey. *Artificial Intelligence Review*, 42:79–156, 2014.
139. Siddhartha Bhattacharyya, Ujjwal Maulik, and Paramartha Dutta. *Quantum Inspired Computational Intelligence: Research and Applications*. Morgan Kaufmann, 2016.

140. Andrea Saltelli, Stefano Tarantola, Francesca Campolongo, Marco Ratto, et al. *Sensitivity analysis in practice: a guide to assessing scientific models*, volume 1. Wiley Online Library, 2004.
141. Jon Herman and Will Usher. Salib: An open-source python library for sensitivity analysis. *Journal of Open Source Software*, 2(9):97, 2017.
142. Takuya Iwanaga, William Usher, and Jonathan Herman. Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses. *Socio-Environmental Systems Modelling*, 4:18155, May 2022.
143. DM Hamby. A comparison of sensitivity analysis techniques. *Health physics*, 68(2):195–204, 1995.
144. Emanuele Borgonovo and Elmar Plischke. Sensitivity analysis: A review of recent advances. *European Journal of Operational Research*, 248(3):869–887, 2016.
145. Peter Doubilet, Colin B Begg, Milton C Weinstein, Peter Braun, and Barbara J McNeil. Probabilistic sensitivity analysis using monte carlo simulation: a practical approach. *Medical decision making*, 5(2):157–177, 1985.
146. Alexei Botchkarev. Evaluating performance of regression machine learning models using multiple error metrics in azure machine learning studio. *Available at SSRN 3177507*, 2018.