Received: June 25, 2025; **Accepted:** October 7, 2025; **Published:** January 1, 2026.
DOI. 10.30473/coam.2025.74960.1316
Volume 11, Issue 1, p.p. 1-29: Winter-Spring (2026)
Research Article

Open Access

Control and Optimization in Applied Mathematics - COAM

Designing a New Continuous Quantum Evolutionary Algorithm for Nonlinear Optimization and Efficiency Frontier Evaluation

Tahereh Azizpour, **Majid Yarahmadi** ✉

Department of Mathematics and Computer Science, Lorestan University, Lorestan, 68151-44316, Iran.

✉ **Correspondence:**
Majid Yarahmadi
E-mail:
yarahmadi.m@lu.ac.ir

How to Cite
Azizpour, T., Yarahmadi, M. (2025). "Designing a New Continuous Quantum Evolutionary Algorithm for Nonlinear Optimization and Efficiency Frontier Evaluation", Control and Optimization in Applied Mathematics, 11(1): 1-29, doi: 10.30473/coam.2025.74960.1316.

Abstract. In this paper, we introduce a new continuous quantum evolutionary optimization algorithm designed for optimizing nonlinear convex functions, non-convex functions, and efficiency evaluation problems using quantum computing principles. Traditional quantum evolutionary algorithms have primarily been implemented for discrete and binary decision variables. The proposed method has been designed as a novel continuous quantum evolutionary optimization algorithm tailored to problems with continuous decision variables. To assess the algorithm's performance, several numerical experiments are conducted, and the simulated results are compared with the Grey Wolf Optimizer and Magnet Fish Optimization search algorithm. The simulation results indicate that the proposed algorithm can approximate the optimal solution more accurately than the two compared algorithms.

Keywords. Convex optimization, Non-Convex optimization, Evolution algorithm, Quantum evolution algorithm, Efficient frontier, Markowitz efficient frontier.

MSC. 81Q93; 62P05.

https://mathco.journals.pnu.ac.ir

1 Introduction

An algorithm is a set of instructions and structures that we can use to solve complex problems. Since soft computing methods require specific structures to solve complex problems, designing efficient algorithms is very important [27]. One type of the most popular algorithms in this area is the evolutionary algorithms or EA [3]. The evolutionary algorithms are a kind of computational algorithm in artificial intelligence that it draws inspiration from the principles of evolution and natural selection in organisms to solve the complex problems. These algorithms find optimal solutions to complex problems through continuous and dynamic repetition [19].

Generally, the evolution algorithms are based on exploration and they are used to solve problems that can not be solved in polynomial time. Such as classically hard problems of degree $NP-$Hard and other problems that take too long to complete [6]. However, evolutionary genetic algorithms are often used together with other methods to serve as a quick solution to find an optimal starting point for other algorithms [22].

A compact MLCP-based projection recurrent neural network model to solve shortest path problem is designed [10]. This paper guarantees convergence to the global optimum by focusing on optimization problems with nonlinear, non-convex, and constrained constraints.

There are many interpretations of the risk concept, there is risk or danger in any activity that does not have a 100% chance of success. Today, most researchers associate the risk of an investment with changes in the "rate of return"; that is, the greater the changes in an investment's return, the higher the risk [26]. While there is no complete agreement in this field, most analysts agree that the risk should be calculated according to the return changes and the standard deviation of the return rate [25]. In fact, risk is defined as the possible changes or standard deviation of the portfolio's expected return.

In the real world, increasing the number of stocks in an investment portfolio reduces its risk, but adding a large number of securities does not completely eliminate risk; a certain amount of risk always remains [5].

The most important concepts in investment decisions (in the portfolio) are return and risk. The relationship between return and risk is a "direct" relationship [12]. This is the reason why different portfolio selection models and risk measurement criteria have been presented by financial experts who seek to find achievable and efficient optimal points according to the portfolio's return and risk. These optimal portfolios are located, as some points, on the efficient frontier [13]. This is where the concept of "risk management" comes from. Risk management is the same process through which an organization or investor reacts to various risks in an optimal approach. From a mathematical point of view, risk management is the process of shaping a loss distribution [23]. When making investment decisions, investors simultaneously consider the risk and return of different options. Risk and return, if not the only influential dimensions in the field of investment decisions, are without a doubt the most important ones [5]. In the financial and economic literature, it is clearly stated that a wise person is someone who seeks to achieve a certain level of return by bearing the minimum possible risk. Also, some investments want to achieve the maximum return at a certain level of risk. Therefore, risk is an inseparable part of return and we cannot talk about investment return without considering the associated risk [5]. In 1952, Markowitz proposed the concept of optimal portfolio and showed that it is possible to form thousands of different portfolios with different levels of risk and return at the same time [20]. It is up to the investor to decide how much risk he can tolerate and diversify his portfolio risk in such a way that he gets the highest possible return. Experts have introduced different metrics to measure risk, each of which refers to an aspect of the category of uncertainty, and sometimes they complement each other. Risk measurement criteria were first determined by studying statistical dispersion indices (such as variance, standard deviation, range). After that, newer measures such as duration, beta coefficient, and value at risk were developed [28].

A convex optimization problem finds the minimum of the convex functions (or the maximum of the concave function) in the convex set [1, 15]. The most important advantage of this type of optimization problem is that every local optimal solution is also a global optimal solution and every optimization algorithm that finds a local optimal solution has actually found a global optimal point. Many important practical problems are non-convex, and most non-convex problems are hard (if not impossible) to solve exactly in a reasonable time [21]. Indeed, most non-convex problems suffer from the ‘‘curse’’ of local minimum, which may trap algorithms into a spurious solution. In [29], Tuy is proposed a mathematical programming problem to solve a non-convex optimization problem. This method separates nonconvex variables into two groups: $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$, such that the objective function and any constraint function is a sum of a convex function of (x, y) jointly and a nonconvex function of x alone. Henrion in [14] shows how GloptiPoly can solve challenging nonconvex optimization problems in robust nonlinear control. GloptiPoly is a general-purpose software with a user-friendly interface.

In the field of non-convex optimization, novel technologies such as quantum computing and quantum-inspired algorithms have provided new possibilities for exploring the solution space. Research on integrating quantum computing with evolutionary algorithms began in the late $1990s$, and recently, Quantum-Inspired Evolutionary Algorithms (QEA) have been introduced, which have a better ability to explore the search space [17]. These algorithms can play an effective role in solving difficult optimization problems, including non-convex ones, and significantly help in finding global optimal solutions [16]. In this paper, the idea is to use of the QEA approach to solve convex and non-convex optimization problems. The QEA is a global optimization method. Therefore, it is a good approach to solve non-convex optimization problems. Firstly, we design the QEA to determine Markowitz's efficient frontier, then by changing this algorithm, it can be used for other convex and non-convex optimization problems. Therefore, we want to use Markowitz's risk management model to form optimal portfolios. For this purpose, the standard deviation is considered as the risk measure criterion. On the other hand, to obtain Markowitz's efficient frontier, a constrained optimality problem must be solved so that the optimal portfolio with given risks has the maximum expected return. To solve the constrained optimality problem, a quantum evolutionary algorithm is introduced, where the proposed approach is inspired by quantum computing. To better exploration in the search space of optimization problems, the proposed approach uses the concepts of quantum computing [18].

After designing the QEA to determine Markowitz's efficient frontier, the observer operation is modified to use this algorithm to solve other examples. In example one (constrained and convex), we consider a portfolio consisting of three stocks. The results show that the proposed algorithm can design an optimal portfolio, well. In Example 2, two non-convex functions are optimized using the QEA. According to the real solutions,the QEA can approximate the optimal solution, as well. Also, in the following, various optimization problems are solved by this algorithm.

The paper is organized as follows: The continuous quantum-inspired evolutionary algorithm is described in Section 2. In Section 3, the components of risk management, optimization of Markowitz risk, and the Markowitz efficient frontier are presented. In Section 4, the quantum evolution algorithm to determine the Markowitz efficient frontier is presented. Examples and case studies simulations are also given in Section 5. In Section 6, test optimization functions are presented. Also, the sensitivity analysis of the proposed algorithm is discussed in Section 7. Finally, the paper's conclusion is presented in Section 8.

2 Designing of Quantum Evolutionary Algorithm

The Quantum Evolution Algorithm (QEA) uses a new representation that is based on the concept of a qubit. The smallest unit of stored information in a quantum computer is called a qubit. A qubit can be on the state ‘‘0”, state ‘‘1” or a superposition of them. The state of a qubit can be expressed as follows:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad s.t. \quad |\alpha|^2 + |\beta|^2 = 1. \tag{1}$$

Where α and β are complex numbers such that $|\alpha|^2$ and $|\beta|^2$ represent the probabilities of the qubit being in states $|0\rangle$ and $|1\rangle$, respectively. In the QEA, an individual qubit is the string of m qubits and can be described as follows [11]:

$$\begin{bmatrix} \alpha_1 & | & \alpha_2 & | & ... & | & \alpha_m \\ \beta_1 & | & \beta_2 & | & ... & | & \beta_m \end{bmatrix}, \quad s.t. \quad |\alpha_i|^2 + |\beta_i|^2 = 1, \; i = 1, 2, ..., m. \tag{2}$$

This representation indicates a base vector for linear combination of any states. Suppose that the following matrix represents the quantum state corresponding to $x = (x_1, x_2, x_3)$:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & 1 & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & 0 & \frac{\sqrt{3}}{2} \end{bmatrix}, \tag{3}$$

the first column of the matrix corresponds to the first qubit (corresponding to x_1), the second column corresponds to the second qubit (corresponding to x_2), and the third column corresponds to the third qubit (corresponding to x_3). In other words:

$$\begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \begin{bmatrix} \alpha_2 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha_3 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}. \tag{4}$$

The overall state of the system is expressed in terms of eight basic states:

$$\{|000\rangle, |001\rangle, |010\rangle, |100\rangle, |011\rangle, |101\rangle, |110\rangle, |111\rangle\}.$$

To determine this state, the coefficients $c_{|ijk\rangle}$ of the basic states must be specified:

$$|ijk\rangle = |i\rangle \otimes |j\rangle \otimes |k\rangle,\ c_{|ijk\rangle} = c_{|i\rangle} \times c_{|j\rangle} \times c_{|k\rangle}; \quad i, j, k \in \{0, 1\}. \tag{5}$$

For example:

$$|000\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle,\ c_{|000\rangle} = c_{|0\rangle} \times c_{|0\rangle} \times c_{|0\rangle} = \frac{1}{\sqrt{2}} \times 1 \times \frac{\sqrt{3}}{2} = \frac{\sqrt{3}}{2\sqrt{2}}. \tag{6}$$

Therefore, the state of the system is expressed as follows:

$$\frac{1}{2\sqrt{2}}|000\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|001\rangle + \frac{1}{2\sqrt{2}}|100\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|101\rangle, \tag{7}$$

this demonstrates that the probability observation of states $|000\rangle$, $|001\rangle$, $|100\rangle$ and $|101\rangle$ are $\frac{1}{8}$, $\frac{3}{8}$, $\frac{1}{8}$ and $\frac{3}{8}$, respectively. It is clear, that the $3-$qubit system (3) contains the information of $8-$states. Therefore, it is observed that the quantum evolutionary computations can demonstrate a better diversity of the population than classical version. Let $Q(t)$ be a population of states, where t represents the generation. Then we have:

$$Q(t) = \{q_1^t, q_2^t, ..., q_n^t\},\ q_j^t = \begin{bmatrix} \alpha_{j1}^t & | & \alpha_{j2}^t & | & ... & | & \alpha_{jm}^t \\ \beta_{j1}^t & | & \beta_{j2}^t & | & ... & | & \beta_{jm}^t \end{bmatrix},\ j = 1, ..., n, \tag{8}$$

and, n is the population size. The QEA process for the binary problem is presented in following box [17]:

Algorithm 1 Procedure of QEA

Begin $t \leftarrow 0$.

i. *Initialization:* Initialize $Q(t)$.

ii. *Observation and evaluation:* Form $P(t)$ by observing the states of $Q(t)$.

iii. Evaluate $P(t)$.

iv. *Storage:* Store the best solutions among $P(t)$ into $B(t)$.

v.

while not termination condition **do**

$t \leftarrow t+1$.

vi. Form $P(t)$ by observing the states of $Q(t-1)$.

vii. Evaluate $P(t)$.

viii. Update $Q(t)$ using U-gates.

ix. Store the best solutions among $B(t-1)$ and $P(t)$ into $B(t)$.

x. **IF** (Migration condition) **THEN** Migrate b or b_j^t to $B(t)$ globally or locally.

end while

xi. **return** the best found solution(s) in $B(t)$.

In step (i), $q_j^0 = q_j^t|_{t=0}, j = 1,2,...,n$ is initialized with $\frac{1}{\sqrt{2}}$. That is, the individual qubit q_j^0 is the linear superposition of all states with same probability.
Step (ii) creates binary solutions in generation $t = 0$, i.e., $P(0) = \{x_1^0, x_2^0, ..., x_n^0\}$, by observing $Q(0)$ states. A binary solution $x_j^0, j = 1,2,...,n$ will be obtained by a binary observing 0 or 1 for each bit according to $|\alpha_i|^2$ or $|\beta_i|^2$.
In step (iii), any binary solution x_j^0 will be evaluated.
In step (iv), the best solutions among the binary solutions are stored in $B(0) = \{b_1^0, b_2^0, ..., b_n^0\}$. Also, $b_j^0 = (b_j^t|_{t=0})$ is the same as x_j^0 in the initial generation.
In step (v), run QEA in the $while$ loop until the terminal condition be satisfied.
In steps (vi) and (vii) for steps $while$ loop, the binary solutions are generated by observing $Q(t-1)$ same as the step (ii), and any binary solution is evaluated.
In step (viii) The individual qubits are updated by applying the rotation gate on them as follows:

$$U(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}, \tag{9}$$

where θ is the rotation angle that is designed to designer view point and tuned based on a given tuning law. The details of this step will be explained later.
In steps (ix) and (x), the best solutions are selected based on $B(t-1)$ and $P(t)$ and stored in $B(t)$. If the best solution contained in $B(t)$ is superior to the currently stored solution b, then b is updated by replacing it with this new solution.
In step (xi), If the termination condition is satisfied, the best solution b is assigned to $B(t)$.

2.1 Updating of individual qubit

In QEA, the individual qubit must be updated in a new generation to evolve to an optimal individual. This can be achieved by quantum gate operation. In the theory of quantum mechanics, the common gates are: Hadamard gate, rotation gate, NOT gate and controlled NOT gate. The most commonly used quantum gate in QEA is the rotation gate. Quantum rotation gates can be designed according to the practical problems and usually can be defined as Equation (9), where θ is rotation gate angle. For each of its state vectors $[\alpha_i, \beta_i]^\top$, its update can be obtained as follows:

$$\begin{bmatrix}\alpha_i' \\ \beta_i'\end{bmatrix} = U(\theta)\begin{bmatrix}\alpha_i \\ \beta_i\end{bmatrix} = \begin{bmatrix}cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta)\end{bmatrix}\begin{bmatrix}\alpha_i \\ \beta_i\end{bmatrix}. \tag{10}$$

The updating of a qubit state vector is illustrated in Figure 1.

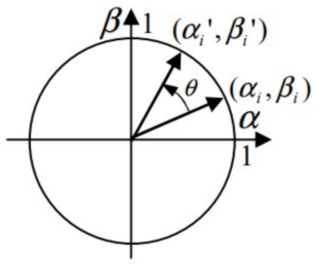

Figure 1: Polar plot of qubit state vector updating in QEA.

Updating all the qubit state vectors, generates a new individual qubit q' expressed as:

$$q' = \begin{bmatrix}\alpha_1' & | & \alpha_2' & | & \dots & | & \alpha_m' \\ \beta_1' & | & \beta_2' & | & \dots & | & \beta_m'\end{bmatrix}. \tag{11}$$

Assume q_j^g denotes $j-$th individual qubit in the $g-$th generation which can be represented as

$$q_j^g = [\alpha_{ji}^g, \beta_{ji}^g]^t, \tag{12}$$

the rotation parameter corresponding to q_j^g is denoted by θ_{ji}^g, which is calculated according to following relation [30].

$$\theta_{ji}^g = s_{ji}^g \Delta\theta_{ji}^g, \tag{13}$$

where s_{ji}^g is the direction indicator of the rotation. In the quantum evolution algorithm, the value and sign of $\Delta\theta$ is determined by the regulation strategy, and the rotation of the qubits is done in a direction that moves the current state to the optimal state, which is presented in Table 1.

Table 1: Adjustment strategy of rotating angle [2, 30].

			$s(\alpha_{ji}^g, \beta_{ji}^g)$				
x_i^g	$best_i^*$	$f(x) \geq f(best)$	$\alpha_{ji}^g \beta_{ji}^g > 0$	$\alpha_{ji}^g \beta_{ji}^g < 0$	$\alpha_{ji}^g = 0$	$\beta_{ji}^g = 0$	$\Delta\theta_{ji}^g$
0	0	false	0	0	0	0	0
0	0	true	0	0	0	0	0
0	1	false	0	0	0	0	0
0	1	true	-1	$+1$	±1	0	0.05π
1	0	false	-1	$+1$	±1	0	0.01π
1	0	true	$+1$	-1	0	±1	0.025π
1	1	false	$+1$	-1	0	±1	0.005π
1	1	true	$+1$	-1	0	±1	0.0025π

x_i^g is the $i-$th bit of the current individual (g-th generation); best i is the $i-$th element of the current optimal individual. Also, $f(x)$ is the fitness function; $s(\alpha_{ji}^g, \beta_{ji}^g)$ is the direction of the rotating angle; $\Delta\theta_{ji}^g$ is the value of the rotating angle.

Conventional quantum evolutionary algorithms address only discrete and binary decision variable problems, and they are unable to solve optimization problems involving continuous decision variables. To address this challenge, this paper presents a new quantum evolutionary optimization algorithm. Table 2 outlines the primary approaches and innovative ideas proposed in this paper to overcome these issues and challenges.

3 Measurement of Return and Risk

The two basic indices of portfolio management are return and risk, and a trade off between them is an optimal objective. Consider n shares with their prices at any time, as follows [25]:

$$P_i(t), \ \ i = 1, 2, ..., n, \ \ t = 1, 2, ..., T, \tag{14}$$

the return of i stock is defined as follows:

$$r_i(t) = \frac{P_i(t) - P_i(t-1)}{P_i(t-1)}. \tag{15}$$

Let X be the total wealth and let w_i be the portion of the total wealth for buying or holding i stock. Then

$$\sum_{i=1}^{n} w_i = X, \quad or \quad \sum_{i=1}^{n} w_i \leq X. \tag{16}$$

We want to assign the total wealth between n stocks. The amount of exact wealth isn't important for portfolio design. The decision key variable is what proportion of this budget is invested in

Table 2: Issues, ideas, and related sections of the proposed CQEA.

Issues and Challenges	Ideas and Novelties	Related Sections
Design a quantum evolutionary algorithm to determine the Markowitz efficient frontier.	A new Continuous Quantum Evolutionary Algorithm (CQEA) is designed to determine the Markowitz efficient frontier.	Sections 3, 2, 4 and Algorithm 2.
Disability of the conventional quantum evolutionary algorithm for solving continuous optimization problems.	In CQEA, the observer operation is modified to extend the algorithm to other continuous optimization problems.	Sections 4 and 5.
Evaluate the technical characteristics of the proposed optimization algorithm.	The proposed algorithm is tested using well-known benchmark functions, including unimodal and multimodal functions.	Section 6.
Examining the performance of the proposed CQEA.	Two performance indices are computed to demonstrate the convergence behavior of the algorithm.	Subsection 5.1.
Application and implementation.	Six examples (convex, non-convex, and efficient evaluation optimization problems) and two high-dimensional test functions are simulated.	Sections 5 and 6.
Sensitivity analysis.	Sensitivity analysis is conducted on the key parameters of the CQEA to evaluate their impact on convergence and solution quality.	Section 7.

each stock. This approach action removes X from the equations. For this purpose, one can use weight in calculations (total wealth is considered as one unit):

$$x_i = \frac{w_i}{X} = \frac{w_i}{\sum_{i=1}^{n} w_i} \Rightarrow \begin{cases} \sum_{i=1}^{n} x_i = 1, \\ 0 \leq x_i \leq 1, \end{cases} \tag{17}$$

where x_i is the invested weight, in other words, what proportion of the wealth is invested in i stocks. The return of stock at any time is obtained as follows:

$$R_P(t) = \sum_{i=1}^{n} x_i r_i(t), \tag{18}$$

that $r_i(t)$ is the return of stock i at time t. Also, the expected return of the portfolio is calculated as follows:

$$\mu_p = \mathbb{E}(R_P(t)) = \sum_{i=1}^{n} \mathbb{E}(r_i(t)) x_i = \sum_{i=1}^{n} \mu_i x_i, \tag{19}$$

where μ_i is the expected return on stock i. Here, the risk of the portfolio is measured by variance or standard deviation, in other words:

$$\sigma_P^2 = var(R_P(t)) = \sum_{i=1}^{n} \sigma_i^2 x_i^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j, \quad \rightarrow \sigma_P^2 = x^T \sum x \geq 0, \tag{20}$$

σ_i^2 is the variance of the return of stock i, σ_{ij} is the covariance between stocks i and j, $\sum$ is the variance -covariance matrix, and σ_P^2 is the variance of the portfolio return.

3.1 Optimization of Markowitz Risk

According to the Markowitz model, risk is related to return fluctuations, and these fluctuations are measured by return variance. The rate of return of a portfolio, consisting of different assets, is obtained from the weighted average of the returns of the individual assets that make up that portfolio. To optimize the portfolio based on the Markowitz risk management model, the following non-linear programming model must be solved [20]:

$$Mean - variance \text{ model:} \begin{cases} \max \quad \mu_P \\ \min \quad \sigma_P^2 \\ s.t. \quad \sum_{i=1}^{n} x_i = 1, \quad x_i \geq 0. \end{cases} \tag{21}$$

Optimization problem (21) is a bi-objective problem. This problem can be reformulated in two simpler forms:

$$\begin{aligned} &\max \quad \mu_P = \sum_{i=1}^{n} \mu_i x_i \\ &s.t. \\ &\sigma_P^2 = \sum_{i=1}^{n} \sigma_i^2 x_i^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \leq \sigma_{P_0}^2 = v \\ &\sum_{i=1}^{n} x_i = 1, \quad x_i \geq 0. \end{aligned} \tag{22}$$

$$\begin{aligned} &\min \quad \sigma_P^2 = \sum_{i=1}^{n} \sigma_i^2 x_i^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \\ &s.t. \\ &\mu_P = \sum_{i=1}^{n} \mu_i x_i \geq \mu_{P_0}^2 = R, \\ &\sum_{i=1}^{n} x_i = 1, \quad x_i \geq 0. \end{aligned} \tag{23}$$

3.2 Efficient Frontier

Definition 1. A stock portfolio x is called efficient if it has the highest expected return among all portfolios with the same variance or it has the lowest variance among all portfolios with a certain expected return.

The set of efficient portfolios makes the efficient frontier. The efficient frontier is displayed as a curve in a two-dimensional graph, where the coordinates of a point correspond to the expected return and return deviation of the efficient portfolio. In Figure 2, the curve AB shows the efficient frontier. This efficient set on the curve AB has priority on all of the curve internal

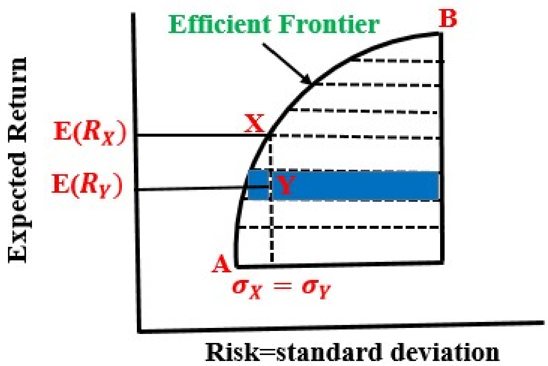

Figure 2: Efficient frontier.

portfolios. Due to the specific risk, they have a higher expected return, or their risk is the least for a given expected return. For example, consider portfolios X and Y on the AB frontier curve. Although both portfolios have the same risk, the expected return of X is greater than that of Y, therefore X will be preferred.

4 The CQEA to Determine Markowitz's Efficient Frontier

In Section 2, the quantum evolutionary algorithm was presented for the binary optimization problems. The application of this algorithm is limited to optimization problems whose variables are in the set $\{0, 1\}$, e.g., the knapsack problem. But there are many optimization problems that have continuous variables. In this section, we will present modifications to the QEA algorithm for the optimization of continuous problem, and evaluate the Markowitz efficient frontier, especially.

Suppose we have a portfolio that contains m stocks. We want to determine the percentage of investment on each stock to reached the maximum expected return while bearing the minimum risk v. In other words, let μ_i and σ_i be the expected return and risk of stock i, respectively. Additionally, the amount of investment on this stock is indicated with x_i. Mathematical modeling of the Markowitz efficient frontier problem, is considered as nonlinear optimization problem as follows:

$$\begin{aligned}
&\max \quad \sum_{i=1}^{m} \mu_i x_i \\
&s.t. \\
&x^T R x \leq v, \quad x^T = [x_1, ..., x_m], \\
&\sum_{i=1}^{m} x_i = 1, \;\; x_1, ..., x_m \in [0,1].
\end{aligned} \tag{24}$$

In optimization problem (24), the matrix R represents the covariance-variance matrix. As shown, each x_i belongs to the interval $[0, 1]$; hence, the decision variables are continuous. Consequently, we need to modify the previous quantum evolution algorithm. First, to enhance diversity in the population, we employ the chaotic approach [11] instead of the approach based on the uniform selection strategy.

4.1 Population Initialization Based on the Chaotic Approach

Consider a chaos state variable generated by the following mathematical model:

$$\phi_{k+1} = \mu\phi_k(1-\phi_k), \quad \phi_k \in [0,1], \quad \mu = 4. \tag{25}$$

The chaos variable ϕ_k is obtained by solving Equation (25). The quantum population initialization is as follows [11]:

1. $i = 1$.

2. Generate m random numbers $\phi_j^0, \;\; (j = 1, 2, \ldots, m)$ in the interval $(0, 1)$.

3. Calculate $\phi_j^i, \;\; (j = 1, 2, \ldots, m)$ using iteration Equation (25):
$$\phi_j^i = \mu\phi_j^i(1-\phi_j^i). \tag{26}$$

4. Create a population member $\begin{bmatrix}\alpha_j^i & \beta_j^i\end{bmatrix}^T$, $j = 1, \ldots, m$ where $\alpha_j^i = cos(2\phi_j^i\pi)$ and $\beta_j^i = sin(2\phi_j^i\pi)$.

5. Provide a $m-$ qubit cell $\begin{bmatrix}\alpha_1^i & | & \alpha_2^i & | & \ldots & | & \alpha_m^i \\ \beta_1^i & | & \beta_2^i & | & \ldots & | & \beta_m^i\end{bmatrix}$.

6. $i = i + 1$.

7. If $i > N_{pop}$ then N_{pop} has been obtained, otherwise go to Step 3.

4.2 Observability

Every $P(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$ where $x_j^t = [x_{j1}^t, x_{j2}^t, \dots, x_{jm}^t]$, and $j = 1, 2, \dots, n$, is a classical solution by using an observation of $Q(t)$. Now, we need to define the appropriate observable operator.

Let Γ_1, Γ_2 and Γ_3 operators are as follows:

$$\Gamma_1 : \mathbb{R}^{2\times m} \to \mathbb{R}^{1\times m}$$
$$\Gamma_1(q_j^t) := \left[\langle O\rangle_{|\psi_{j1}^t\rangle}, ..., \langle O\rangle_{|\psi_{jm}^t\rangle}\right] = \left[\langle\psi_{j1}^t|O|\psi_{j1}^t\rangle, ..., \langle\psi_{jm}^t|O|\psi_{jm}^t\rangle\right], \tag{27}$$

where $O = diag([a, b]), 1 < a < b$ and $|\psi_{ji}^t\rangle = \begin{bmatrix}\alpha_{ji}^t & \beta_{ji}^t\end{bmatrix}^\top$. Note that O is a diagonal Hermitian operator, therefore, one can obtain:

$$a \le \langle\psi_{ji}^t|O|\psi_{ji}^t\rangle \le b. \tag{28}$$

Now, by considering $z_j^t = \Gamma_1(q_j^t)$ where $z_j^t = \begin{bmatrix}z_{j1}^t, ..., z_{jm}^t\end{bmatrix}$, Γ_2 is defined as follows:

$$\Gamma_2 : \mathbb{R}^{1\times m} \to \mathbb{R}^{1\times m}$$
$$\Gamma_2(z_j^t) := \frac{1}{b-a} z_j^t + \frac{a}{a-b}. \tag{29}$$

Considering $y_j^t = \Gamma_2(z_j^t)$, we define Γ_3 as follows:

$$\Gamma_3 : \mathbb{R}^{1\times m} \to \mathbb{R}^{1\times m}$$
$$\Gamma_3(y_j^t) := \frac{1}{\sum_{i=1}^m y_{ji}^t} y_j^t. \tag{30}$$

Finally, Γ is defined as the following relations:

$$\Gamma : \mathbb{R}^{1\times m} \to \mathbb{R}^{1\times m}$$
$$\Gamma(q_j^t) := \Gamma_3\left(\Gamma_2(\Gamma_1(q_j^t))\right). \tag{31}$$

Now, the population at generation t in real space is calculated as follows:

$$P(t) = \left\{x_1^t, ..., x_j^t\right\} = \left\{\Gamma(q_1^t), ..., \Gamma(q_n^t)\right\}. \tag{32}$$

According to the operator Γ, the constraints $x_i \in [0, 1]$ and $\sum_{i=1}^m x_i = 1$ can be removed from the optimization problem (24). In next, the proposed Continuous Quantum Evolutionary Algorithm (CQEA) is designed to determine the Markowitz efficient frontier.

Remark 1. The time complexity of this algorithm is

$$O\big(T \cdot N_{pop} \cdot m\big).$$

If the algorithm optimizes a more complex objective function $f(x)$, its time complexity becomes

$$O\big(T \cdot N_{pop} \cdot \big(m + \text{cost}(f(x))\big)\big).$$

Algorithm 2 CQEA to determine the Markowitz efficient frontier

Input: Parameters N_{pop}, T, θ and a termination condition
Output: The set of optimal solutions on the Markowitz frontier

Initialization

Step 1. Choose N_{pop}, T, θ and define a termination condition.
Step 2. Initialize $Q(t)$ using a chaotic approach.

Main loop

Step 3. Choose $1 \leq a < b$ and form the diagonal matrix $O = \text{diag}([a, b])$.
Step 4. Define for each j, $\Gamma_1(q_j^t) = \left[\langle \psi_{j1}^t \mid O \mid \psi_{j1}^t \rangle, \ldots, \langle \psi_{jm}^t \mid O \mid \psi_{jm}^t \rangle\right]$.
Step 5. Define for each j, $\Gamma_2(z_j^t) = \frac{1}{b-a}\, z_j^t + \frac{a}{a-b}$.
Step 6. Define for each j, $\Gamma_3(y_j^t) = \frac{y_j^t}{\sum_{i=1}^m y_{ji}^t}$.
Step 7. Composite transformation: $\Gamma(q_j^t) = \Gamma_3\Big(\Gamma_2\big(\Gamma_1(q_j^t)\big)\Big)$.
Step 8. Population update: $x_j^t = \Gamma(q_j^t),\ P(t) = \{x_1^t, \ldots, x_j^t\} = \{\Gamma(q_1^t), \ldots, \Gamma(q_n^t)\}$.
Step 9. Evaluation: Compute the fitness/objective for the new population $P(t)$.
Step 10. Store the optimal solution of (24) in $B(t)$.
Step 11. Examine termination condition.
Step 12. If the termination condition is satisfied then $B(t)$ is final optimal solutions (24), else go to Step 13.
Step 13. Update $Q(t)$ by U-gate (9) and go to Step 4.

Theorem 1. Suppose that x^* is the minimum of continuous function $f(x)$ and $\{x^t\}$ is the sequence generated by Algorithm 2, then $\{f(x^t)\}$ converges in mean to $f(x^*)$.

Proof. Let $f(x^*) = \min_{x \in [a,b]} f(x)$, where $x^t = \{x_1^t, x_2^t, \ldots, x_N^t\}$ and x_j^t is the j-th person of the t-th generation. Obviously, for each $j = 1, \ldots, N_{pop}$,

$$f(x_j^t) \geq f(x^*), \qquad \sum_{j=1}^{N_{pop}} f(x_j^t) \geq N_{pop} f(x^*). \tag{33}$$

Hence,

$$\mathbb{E}\big[f(x^t)\big] \geq f(x^*). \tag{34}$$

According to Step 13 of Algorithm 2, Subsection 2.1, and [2], one can infer that

$$\mathbb{E}\big[f(x^{t+1})\big] \leq \mathbb{E}\big[f(x^t)\big]. \tag{35}$$

Therefore, the sequence

$$\left\{ \mathbb{E}[f(x^t)] \right\}_{t=1}^{T}, \tag{36}$$

is monotone non- increasing and bounded sequence. Therefore, this sequence is a convergent sequence.

Let $\mathbb{E}[f(x^t)] \to \hat{f}^*$. From (33), it immediately follows that $\hat{f}^* \geq f(x^*)$. If $\hat{f}^* = f(x^*)$, in this case, proof was completed. Otherwise, suppose that $\hat{f}^* > f(x^*)$. This implies that in none of the generations produced by the algorithm does the optimal solution x^* occur. Since the process x^t is a Markov process, and according to Algorithm 2 (where each generation is produced solely based on the preceding generation in an optimally manner), the probability of obtaining the optimal solution x^* in any generation is non-zero [8]. Therefore,

$$P\left(x_k^{t+1} = x^* \,\middle|\, x^t\right) \geq p > 0. \tag{37}$$

Let

$$A_t = \left\{ x_j^t \,\middle|\, P\left(x_j^t \neq x^*\right) = p_1 \right\}, \quad 0 < p_1 < 1. \tag{38}$$

From (37) and (38), using conditional probability, we have

$$P(A_{t+1}) \leq P(A_t)(1-p). \tag{39}$$

With a same and repetitive process, it can be inferred that

$$P(A_{t+1}) \leq P(A_1)(1-p)^t. \tag{40}$$

Consequently,

$$\lim_{t\to\infty} P(A_{t+1}) = 0, \tag{41}$$

since $0 < P(A_1) < 1$ and $0 < 1-p < 1$. Therefore, the probability that x^* is not produced by the algorithm in any generation is zero. Furthermore, from Equation (34) and since the sequence $\{\mathbb{E}[f(x^t)]\}$ is monotone and bounded below, it follows that $\hat{f}^* = f(x^*)$. Equivalently, the sequence generated by the proposed algorithm converges in mean, which completes the proof. □

Note that the CQEA is prescribed for the determination of the Markowitz efficient frontier. Now, if the same algorithm is applied to other optimization problems, it is typically unnecessary to choose $a > 1$; in fact, a may be a negative. Moreover, the operations Γ_2 and Γ_3 are often chosen as identity mappings in such general settings.

5 Examples and Case Studies Simulations

In the following, we want to implement the proposed algorithm for different types of optimization problems, including convex and non-convex optimization and non-convex constrained nonlinear optimization problems. To implement the simulations, MATLAB 2023a on an Intel Core-i5 1.7 GHz Turbo 3.4 GHz PC.

Example 1. Consider a portfolio consisting of three stocks S_1, S_2 and S_3. The mean and covariance-variance matrix of stocks are listed in Tables 3 and 4.

Table 3: Mean of stock.

	S_1	S_2	S_3
Mean of return	0.1206	0.0785	0.0632

Table 4: Covariance–variance matrix.

	S_1	S_2	S_3
S_1	0.0284	0.0040	0.0002
S_2	0.0040	0.0114	−0.0002
S_3	0.0002	−0.0002	0.0012

Now, we want to implement the presented continuous quantum evolution algorithm. In this example, the population size, namely $N_{pop} = 200$ and the number of generation $T = 100$ are considered.

By implementing the proposed algorithm risk values, the expected return of the portfolio and the weight of stocks are presented in Table 5. For example, if we allocate about 71% of the budget to stock 1, about 27% to stock 2, and about 2% to stock 3, we expect a maximum return of 10.83% with risk of 13%.

Figure 3 indicates the mean-variance efficient frontier obtained by implementing the proposed CQEA algorithm.

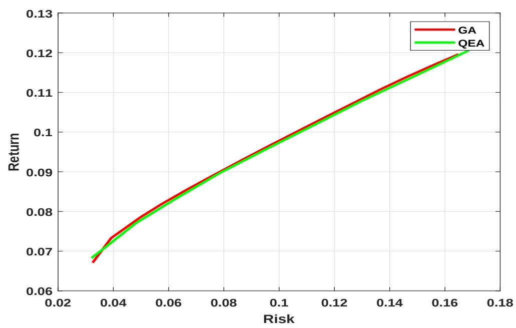

Figure 3: Efficient frontiers with CQEA and Genetic algorithms.

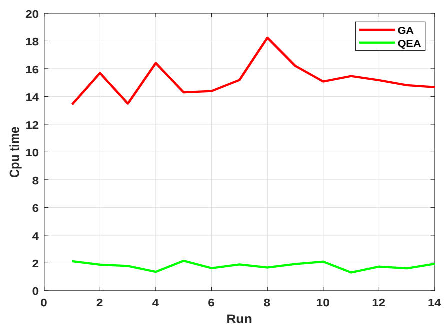

Figure 4: CPU time for CQEA and Genetic algorithms.

To evaluate the accuracy and efficiency of the proposed algorithm, the results obtained from its implementation were compared with the results obtained from the genetic algorithm. The

Table 5: Expected return, standard deviation, and corresponding weights of each stock for optimal portfolios based on the mean–variance model using CQEA.

Expected return	Risk (var)	x_1	x_2	x_3
0.0680	0.033	0.0513	0.1192	0.8294
0.0736	0.040	0.1453	0.1377	0.7170
0.0785	0.050	0.2350	0.1163	0.6487
0.0828	0.060	0.2959	0.1699	0.5342
0.0867	0.070	0.3621	0.1775	0.4604
0.0905	0.080	0.4302	0.1686	0.4012
0.0941	0.090	0.4752	0.2383	0.2865
0.0977	0.100	0.5375	0.2391	0.2234
0.1048	0.120	0.6704	0.2056	0.1239
0.1084	0.130	0.7167	0.2624	0.0209
0.1118	0.140	0.7926	0.2007	0.0067
0.1150	0.150	0.8676	0.1315	0.0009
0.1181	0.160	0.9399	0.0589	0.0011
0.1206	0.170	1.0000	0.0000	0.0000

agreement between the results of both methods confirms the accuracy of the proposed algorithm. In addition, the comparison of the CPU Time execution time in both algorithms shows that the proposed algorithm always spends less time on solving the problem, which is a significant advantage in practical applications. These results confirm that the proposed algorithm has a relative advantage not only in terms of accuracy, but also in terms of efficiency (See Figure 4). Additionally, at the 5% significance level ($\alpha = 0.05$), the test results showed that the $t-$value is -3.34 and the $p-$value is 0.0087. Since the p-value is less than α, the results are statistically significant. Therefore, it can be concluded that the difference under study is significant.

Example 2. In Example 1, we had a convex optimization problem, but in the following, we present two functions that are non-convex. We want to minimize them by the proposed continuous quantum evolutionary algorithm. Here, it is sufficient that put $\Gamma = \Gamma_1$. The shapes of the functions are shown in Figures 5 and 7, respectively. Now, let:

$$f_1(x) = A + x^2 - Acos(2\pi x), \ A = 10, \quad x \in [-5.12, 5.12], \tag{42}$$

$$f_2(x) = -xsin(\sqrt{|x|}), \quad x \in [-300, 500]. \tag{43}$$

As Figures 6 and 8 indicate, the proposed method converges all solutions to the minima of the underlying functions. Moreover, the errors between the exact solutions (for example, $min\ f_1(x) = 0, min\ f_2(x) = -418.9826$) and the approximate solutions are reported in Table 6.

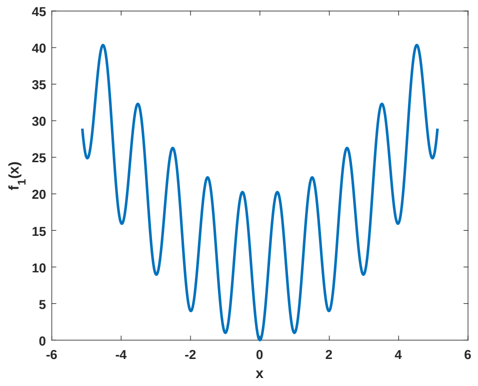

Figure 5: The function $f_1(x) = A + x^2 - Acos(2\pi x)$.

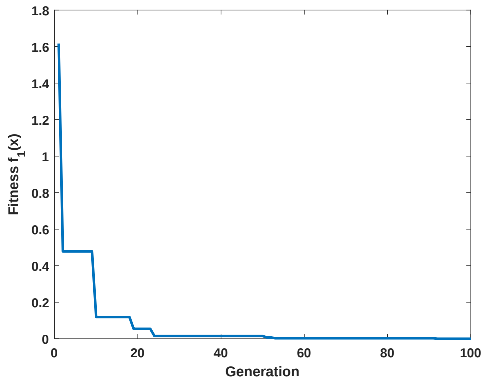

Figure 6: The convergence behavior of $f^{CQEA}_{1(min)}$.

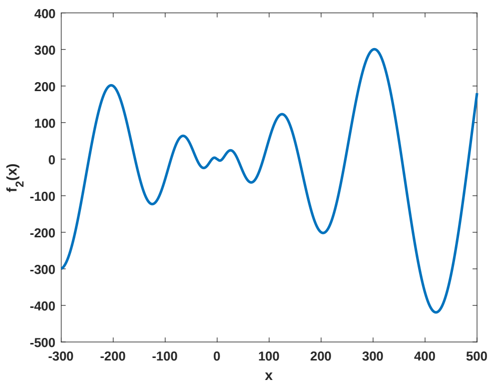

Figure 7: The function $f_2(x) = -xsin(\sqrt{|x|})$.

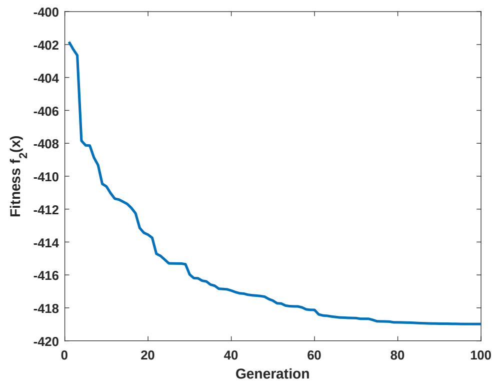

Figure 8: The convergence behavior of $f^{CQEA}_{2(min)}$.

To solve optimization problems (42) and (43), two additional algorithms, namely MFO (Magnet Fish Optimization) and GWO (Gray Wolf Optimizer), are implemented; their results are demonstrated in Tables 6 and 7. As Table 6 demonstrates, the presented method converges to optimal solutions with substantially higher accuracy, while incurring only a modest and acceptable increase in CPU time compared to the other methods.

Table 7 shows the results of the significance test between the algorithms. In the comparison between the algorithm CQEA and the algorithm MFO (also, CQEA and GWO), the results of the t-tests showed that the comparison was significant at the 0.05 significance level.

Example 3. Consider the following pseudo-convex optimization problem [9]:

$$\begin{aligned} & min \quad \frac{x^T Ax + c^T x - 4}{d^T x + 2} \\ & s.t. \quad x \in \Omega = \{x \in \mathbb{R}^4 \mid 0 \leq x_i \leq 10, i = 1, \ldots, 4\}. \end{aligned} \tag{44}$$

The $CQEA$ algorithm was implemented to solve Example 3 and the results are listed in Table 8 ($T = N_{pop} = 100$). Also, the fitness function is plotted in Figure 9.

Table 6: Results of different algorithms for Example 2 ($T = N_{\text{pop}} = 100$).

Algorithm	Best fitness $f_1(x)$	Best fitness $f_2(x)$	Error	CPU time (s)
MFO	-2.5017×10^{-6}	-418.9825974983	$e_1 = -2.5017 \times 10^{-6}$ $e_2 = 0.0185$	$t_1 = 0.0156$ $t_2 = 0.1719$
CQEA	3.3289×10^{-12}	-418.982600000003	$e_1 = 3.3289 \times 10^{-12}$ $e_2 = -1.6435 \times 10^{-4}$	$t_1 = 0.9688$ $t_2 = 0.2512$
GWO	1.4888×10^{-9}	-418.9826000014888	$e_1 = -3.7771 \times 10^{-8}$ $e_2 = 0.0378$	$t_1 = 0.2031$ $t_2 = 0.1875$

Table 7: Results of significance tests between algorithms.

	P-value (significance level = 0.05)	
	CQEA vs. MFO	**CQEA vs. GWO**
$f_1(x)$	$t = -2.59,\ p = 0.0290$	$t = -2.38,\ p = 0.0412$
$f_2(x)$	$t = -2.32,\ p = 0.0412$	$t = -2.52,\ p = 0.0318$

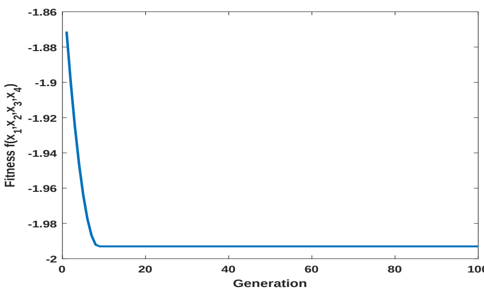

Figure 9: Convergence behavior of fitness function for Example 3.

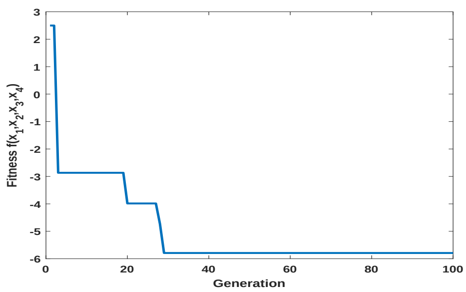

Figure 10: Convergence behavior of fitness function for Example 4.

Table 8: The results of Example 3.

Algorithm	x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$	CPU time (s)
CQEA	0.0002	0.0500	0.0350	0.0007	-1.9930	18.7188
Neurodynamic model [9]	0.0000	0.2820	0.4230	0.0000	-1.7428	–

Example 4. The $CQEA$ algorithm is implemented with $T = 100$ and $N_{pop} = 100$ for the following example [4].

$$min \quad x_1^{0.6} + 2x_2^{0.6} - 2x_2 + 2x_3 - x_4$$

$$s.t.$$

$$\begin{aligned} & x_1 + 2x_3 \leq 4, \\ & -3x_1 + x_4 \leq 1, \\ & x_i \in [0,4], \quad i = 1,\dots,4. \end{aligned} \tag{45}$$

Table 9: Results of Example 4.

Algorithm	x_1	x_2	x_3	x_4	$f(x_1,x_2,x_3,x_4)$	f^* (claimed)	CPU time (s)
CQEA	1.1292	3.8598	0.0134	3.8038	-5.9236	-2.07	0.7188

Example 5. Consider the following convex non-linear program [7]:

$$\begin{aligned} min \quad & -2x_1 - x_2 \\ & s.t. \\ & x_1^2 + x_2^2 \leq 1, \\ & x_1 + 2x_3 \leq 2, \\ & x_i \in [-6,6], \quad i = 1,\dots,4. \end{aligned} \tag{46}$$

The exact solution of this problem is $f(0.8944, 0.4472) = -2.2360$. We use the CQEA algorithm to solve the problem (46) with $T = N_{pop} = 100$. The results are given in Table 10.

Table 10: The result of Example 5.

Algorithm	x_1	x_2	$f(x_1,x_2)$	f^* (exact)	CPU time (s)
CQEA	0.9019	0.4283	-2.2321	-2.2360	1.2344

Example 6. (Three-bar truss design problem) This problem was first examined by Ray and Saini [24]. The main goal is to minimize the weight of bar structures. Three-bar truss problem is modeled as the following nonlinear constraint mathematical model. The problem is prescribed mathematically as follows:

$$\begin{aligned} & \min_{x_1,x_2} f(x) = \left(2\sqrt{2}x_1 + x_2\right) \times L \\ & s.t. g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0, \\ & g_2(x) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0, \\ & g_3(x) = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0. \end{aligned} \tag{47}$$

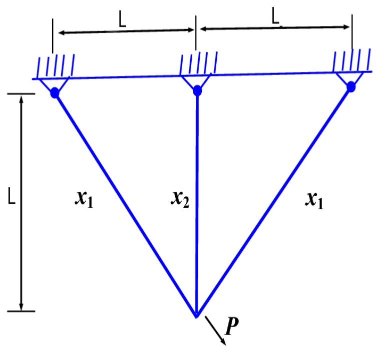

Figure 11: Three bar truss design.

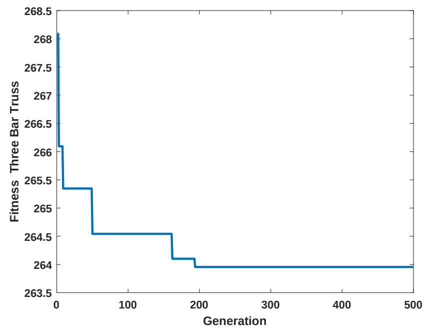

Figure 12: The convergence behavior of fitness function.

where $0 \leq x_1, x_2 \leq 1$.

Let $L = 100\,cm, P = 2KN/cm^2$ and $\sigma = 2KN/cm$. Firstly, we implement $CQEA$ for $T = 500$, $N_{pop} = 200$ and then compare the results obtained with two another algorithms GWO and MFO. For implementing the proposed algorithm, put $\Gamma = \Gamma_2\Gamma_1$. To statistically analyze the performance of the proposed method in this example, the P-Value index was used, and its results are shown in Table 11.

Table 11: Comparison of CQEA with other algorithms for the three-bar truss design optimization.

Algorithm	x_1	x_2	$f(x)$	**CPU time (s)**	**P-value (t-test)**
GWO	0.7839	0.4227	263.9958	0.6719	CQEA vs. GWO: $t = -2.25$, $p = 0.0462$
MFO	0.7908	0.4030	263.9624	0.8594	CQEA vs. MFO: $t = -2.17$, $p = 0.0486$
CQEA	**0.8004**	**0.3954**	**263.9389**	1.0025	–

Table 11 compares the performance of $CQEA$ with the GWO and MFO algorithms in solving the three-bar truss problem. As shown in this table, for the same parameters, the simulation results show that the proposed algorithm can approximate the optimal solution better than the other two algorithms. The computational time of applying the optimization of the three-bar truss design with $CQEA$ was 1.0025 seconds. This period shows that $CQEA$ is at an acceptable level in terms of execution time.

5.1 Performance Indices

Now, we will examine the performance of the proposed algorithm (CQEA). For this aim, we will implement a CQEA for test functions and then we will compute two performance indices to show the convergence of this algorithm. Assume, we want to find the minimum of the test function $f(x)$ based on CQEA method. Optimal solutions computed by the presented algorithm are considered as $f^*_Q(x)$.

Let:

$$I_m(f) = \frac{1}{m}\sum_{i=1}^{m} f^*_{Q^i}, \quad J_m(f) = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}\left(f^*_{Q^i} - f^*\right)^2}, \quad m = 100,$$

where $f^*_{Q^i}$ is i-th run of CQEA. For more investigation, we also calculated the minimum, maximum, and median of the mean value. The results are presented in Table 12. The convergence plots of test functions are drawn in Figures 13 to 16.

6 Test Optimization Functions

In applied mathematics, the test functions are useful to evaluate characteristics of optimization algorithms, such as convergence rate, precision, robustness, and general performance. In this paper, we test the proposed algorithm with the famous test functions. The test functions are divided into two categories, unimodal and multimodal test functions.

1. **Goldstein-Price's function.** The Goldstein-Price function is a global optimization test function. It has only two variables and is modeled as follows:

$$\begin{aligned} f(x_1, x_2) = &\left[1 + (x_1^2 + x_2^2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \\ &\left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]. \end{aligned}$$

   The test area is usually restricted to the square $-2 \leq x_i \leq 2,\ (i = 1, 2)$. Its global minimum value is $f(x) = 3$ with optimum point $(x_1, x_2) = (0, -1)$.

2. **Cross-in-Tray function.** The Cross-in-Tray function has multiple global minima. The function is usually evaluated on the square $x_i \in [-10, 10], \quad (i = 1, 2)$.

$$f(x_1, x_2) = -0.0001\left(\mid sin(x_1)sin(x_2)exp\left(\mid 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \mid\right) \mid +1\right)^{0.1}.$$

   Its global minimum value is $f(x) = -2.06261$ corresponding to the points:

$$(x_1, x_2) = (1.3494, -1.3494), (1.3494, 1.3494), (-1.3494, 1.3494), (-1.3494, -1.3494).$$

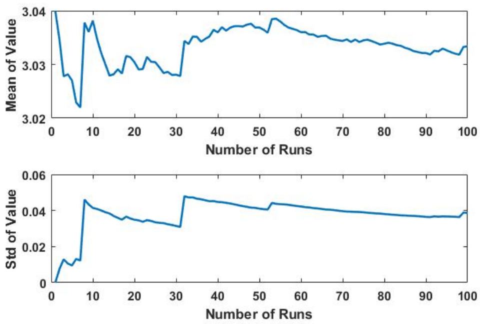

Figure 13: Convergence behavior for Goldstein-Prices test function.

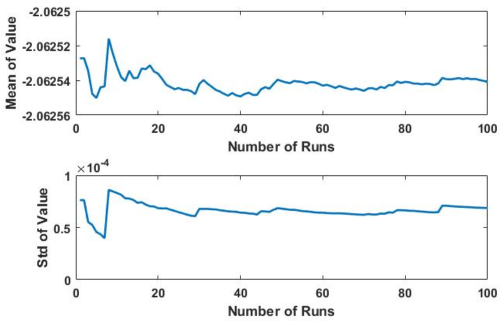

Figure 14: Convergence behavior for Cross-In-Tray function.

Table 12: Benchmark functions used in the experiments.

Test function	Mean	Std	Min	Max	Median	$f_{\min}$ (Exact)
Goldstein–Price	3.0334	0.0387	3.0220	3.0399	3.0342	3
Cross-in-Tray	−2.0625	0.0001	−2.0626	−2.0625	−2.0625	−2.06261

High dimensional test functions

4. **De Jong's1 function.** The first De Jong's function is a continuous, convex and unimodal test function. This function is defined as follows (for $n = 10$):

$$f(x) = \sum_{i=1}^{n} x_i^2.$$

The test area of this function is usually restricted to the hypercube

$$-5.12 \leq x_i \leq 5.12, \ (i = 1, \ldots, n).$$

Global minimum is $f(x) = 0$.

5. **Rastrigin's function.** Rastrigin's function is based on the function of De Jong with the addition of cosine modulation in order to produce frequent local minima. Thus, the test function is highly multimodal. However, the location of the minima are regularly distributed. This function is defined as follows ($n = 10$):

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10cos(2\pi x_i)\right).$$

Test area is usually restricted to hyphercube $-5.12 \leq x_i \leq 5.12$ for $i = 1, \ldots, n$. Its global minimum equal $f(x) = 0$ is obtainable for $x_i = 0, \ (i = 1, \ldots, n)$.

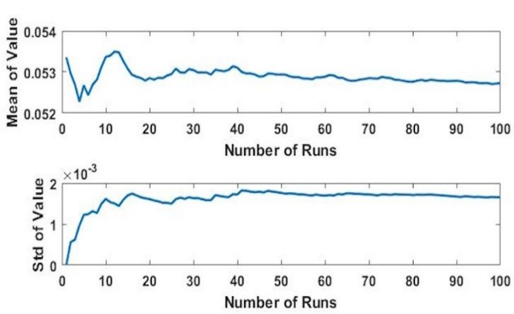

Figure 15: Convergence behavior and boxplot for De Jong's1 Function.

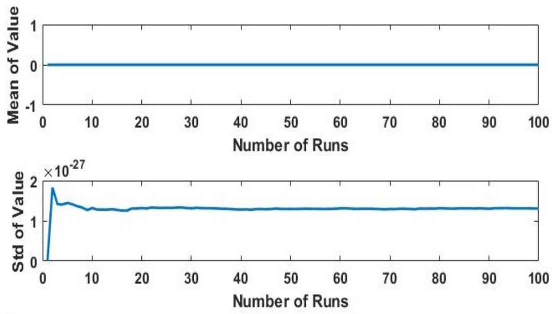

Figure 16: Convergence behavior and boxplot for Rastrigin's Function.

Table 13: Benchmark functions used in the experiments.

Test function	Mean	Std	Min	Max	Median	$f_{\min}$ (Exact)
De Jong's 1	0.0528	0.0016	0.0522	0.05349	0.05286	0
Rastrigin	0	1.3057×10^{-27}	0	0	0	0

7 Sensitivity Analysis

In this study, to evaluate the impact of key parameters of the optimization algorithm, sensitivity analysis on population parameters and number of generations was performed. The aim of this analysis is to identify the parameters that have the greatest impact on the final performance, especially the fitness value.

First, the effect of changing the parameter of the number of generations while keeping the number of populations constant (fixed value 100) was investigated. In this experiment, the generation was changed in different ranges (30 to 400 for functions f_1 andf_2 and 30 to 350 for the Three-bar truss problem) and the results of the error and fitness in each case were recorded.

The results showed that the change in the generation to more than 100 has very little effect to more than 100 has very little effect on the value of the fitness or error and the trend of the error reduction was almost descending and finally from one generation onwards the error and fitness were constant, which shows that this parameter has little sensitivity to the number of generations. In the next step, the number of populations was changed in the case where the generation was kept constant. This experiment was carried out in different ranges (30 to 400 for functions f_1 and f_2 and 30 to 350 for the Three-bar truss problem) and the results were recorded for each case. The results show that increasing the number of populations to more than 100 has very little effect and produces negligible improvements in error and fitness values. See Figures 17 to 19.

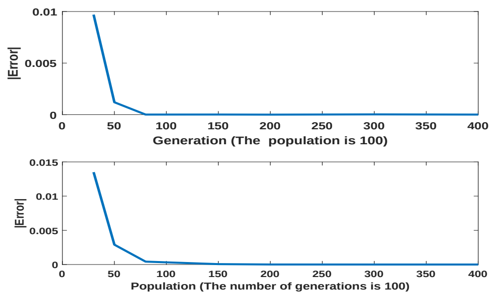

Figure 17: Sensitivity analysis of the CQEA algorithm for $f_1(x)$.

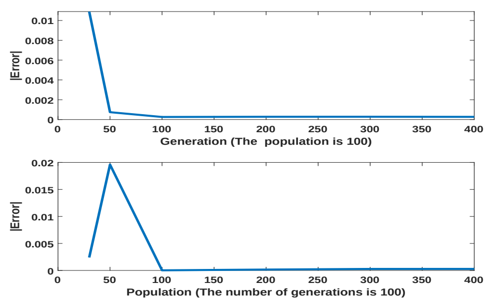

Figure 18: Sensitivity analysis of the CQEA algorithm for $f_2(x)$.

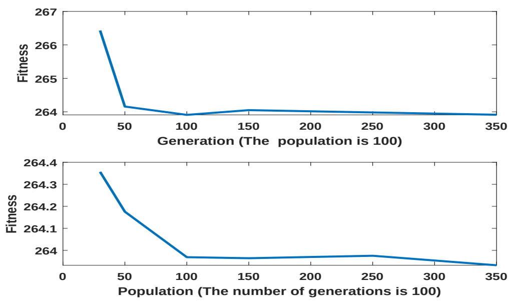

Figure 19: Sensitivity analysis of the CQEA algorithm for three-bar truss.

Overall, the results show that the algorithm is sensitive to the parameters of the number of generations and the population size, and changing these parameters significantly affects performance. However, after the parameters reach a certain value, increasing those parameters no longer causes a significant change in either the error or the fitness, in which case, we say that the sensitivity in that parameter has decreased and its value no longer has a significant effect on the error.

8 Conclusion

Traditional quantum evolutionary algorithms are restricted to problems with discrete and binary decision variables and are not suited for optimization problems involving continuous decision variables. To address this limitation , this study presents a novel quantum evolutionary optimization algorithm. The proposed algorithm is constructed around three operators and a quantum gate within a quantum computing framework. To evaluate the algorithm, three representative problem classes, convex, non-convex and efficiency-evaluation optimization problems are considered. For each class, three benchmark test functions and two high-dimensional functions are implemented, all based on the proposed algorithm, and the results are benchmarked against the Grey Wolf Optimizer (GWO) and Magnet Fish Optimization (MFO) algorithms. The simulation outcomes indicate that the proposed algorithm yields solutions closer to the optimum than both comparative methods.

Declarations

Availability of Supporting Data

All data generated or analyzed during this study are included in this published paper.

Funding

The authors conducted this research without any funding, grants, or support.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Author Contributions

Tahereh Azizpour: Conceptualization; Methodology; Formal analysis; Investigation; Software; Writing – original draft; Visualization. Majid Yarahmadi: Methodology; Validation; Resources; Writing – review & editing; Supervision; Theoretical developments; Project administration.

Artificial Intelligence Statement

Artificial intelligence (AI) tools, including large language models, were used solely for language editing and improving readability. AI tools were not used for generating ideas, performing analyses, interpreting results, or writing the scientific content. All scientific conclusions and intellectual contributions were made exclusively by the authors.

Publisher's Note

The publisher remains neutral regarding jurisdictional claims in published maps and institutional affiliations.

References

[1] Ardakani, J.S., Farahmand Rad, S., Kanzi, N. (2019). "Characterization of properly efficient solutions for convex multiobjective programming with nondifferentiable vanishing constraints". *Control and Optimization in Applied Mathematics*, 3(2), 49-58, doi:https://doi.org/10.30473/coam.2019.42442.1094.

[2] Arjmandzadeh, A., Yarahmadi, M. (2017). "Quantum genetic learning control of quantum ensembles with Hamiltonian uncertainties". *Entropy Journal*, 19(8), doi:https://doi.org/10.3390/e19080376.

[3] Bidlingmaier, M.E. (2023). "An evaluation algorithm for datalog with equality". *Computer Science, arXiv preprint* arXiv:2302.05792, doi:https://doi.org/10.48550/arXiv.2302.05792.

[4] Bunnag, D., Sun, M. (2005). "Genetic algorithm for constrained global optimization in continuous variables". *Applied Mathematics and Computation*, 171(1), 604-636, doi:https://doi.org/10.1016/j.amc.2005.01.075.

[5] Chan, L.K.C., Karceski, J., Lakonishok, J. (1998). "The risk and return from factors". *Journal of Financial and Quantitative Analysis*, 33(2), 159-188, doi:https://doi.org/10.2307/2331306.

[6] Ebrahimia, M.A., Dehnavia, H.D., Mirabib, M., Honaria, V., Sadeghian, A. (2023). "Solving NP-hard problems using a new genetic algorithm". *International Journal of Nonlinear Analysis and Applications*, 14(1), 275-285, doi:10.22075/ijnaa.2022.26610.3364.

[7] Effati, S., Ghomashi, A., Nazemi, A.R. (2007). "Application of projection neural network in solving convex programming problems". *Applied Mathematics and Computation*, 188(2), 1103-1114, doi:https://doi.org/10.1016/j.amc.2006.10.088.

[8] Eiben, A.E., Smith, J.E. (2015). "Introduction to evolutionary computing". *Springer Berlin, Heidelberg*, doi:https://doi.org/10.1007/978-3-662-44874-8.

[9] Eshaghnezhad, M., Effati, S., Mansoori, M. (2017). "A neuro dynamic model to solve nonlinear pseudo-monotone projection equation and its applications". *IEEE Transactions on Cybernetics*, 47(10), 3050-3062, doi:https://doi.org/10.1109/TCYB.2016.2611529.

[10] Eshaghnezhad, M., Effati, S., Mansoori, A. (2022). "A compact MLCP-based projection recurrent neural network model to solve shortest path problem". *Journal of Experimental & Theoretical Artificial Intelligence*, 35(7), 1101-1119, doi:https://doi.org/10.1080/0952813X.2022.2067247.

[11] Gao, J., He, G., Liang, L., Feng, Z. (2014). "A quantum-inspired artificial immune system for the multiobjective 0-1 knapsack problem". *Applied Mathematics and Computation*, 230(1), 120-137, doi:https://doi.org/10.1016/j.amc.2013.12.088.

[12] Ghysels, E., Plazzi, A., Valkanov, R.I. (2016). "The risk-return relationship and financial crises". *January Electronic Journal*, doi:http://dx.doi.org/10.2139/ssrn.2776702.

[13] Guo, X., Chan, R.H., Wong, W.K., Zhu, L. (2019). "Mean-variance, mean-VaR, and mean-CVaR models for portfolio selection with background risk". *Risk Management*, 21, 73-98, doi:http://dx.doi.org/10.2139/ssrn.3254849.

[14] Henrion, D., Lasserre, J.B. (2004). "Solving nonconvex optimization problems". *IEEE Control Systems Magazine*, 24(3), 72-83, doi:https://doi.org/10.1109/MCS.2004.1299534.

[15] Hindi, H. (2004). "A tutorial on convex optimization". *Proceedings of the 2004 American Control Conference, Boston, MA, USA, pp. 3252-3265 vol.4,* doi:https://doi.org/10.23919/ACC.2004.1384411.

[16] Han, K.H., Kim, J.H. (2002). "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization". *IEEE Transactions on Evolutionary Computation*, 6(6), 580-593, doi:https://doi.org/10.1109/TEVC.2002.804320.

[17] Kim, Y., Kim, J.H., Han, K.H. (2006). "Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 Knapsack problems". *IEEE International Conference on Evolutionary Computation, Vancouver, BC, Canada, 2006, pp. 2601-2606*, doi:https://doi.org/10.1109/CEC.2006.1688633.

[18] Li, Y., Jiao, L. (2008). "Quantum-inspired immune clonal algorithm". *Artificial Immune Systems*, 33(2), 304-317, doi:https://doi.org/10.1109/TSMCB.2008.927271.

[19] Liu, Y. (2021). "Evaluation algorithm of teaching work quality in colleges and universities based on deep denoising autoencoder network". *Mobile Information Systems*, 2021, doi:https://doi.org/10.1155/2021/8161985

[20] Markowitz, H. (1952). "Portfolio selection". *Journal of Finance*, 7(1), 77-91, doi:https://doi.org/10.1111/j.1540-6261.1952.tb01525.x.

[21] Mohammadsalahi, K., Modarres Khiyabani, F., Azarmir Shotorbani, N. (2022). "A new optimization method based on dynamic neural networks for solving non-convex quadratic constrained optimization problems". *Control and Optimization in Applied Mathematics*, 7(2), 35-52, doi:https://doi.org/10.30473/coam.2022.64268.1206.

[22] Muñoz, A., Rubio, F. (2021). "Evaluating genetic algorithms through the approximability hierarchy". *Journal of Computational Science*, 53, doi:https://doi.org/10.1016/j.jocs.2021.101388.

[23] Mulvey, J.M., Rosenbaum, D.P., Shetty, B. (1997). "Strategic financial risk management and operations research". *European Journal of Operational Research*, 97(1), 1-16, doi:https://doi.org/10.1016/S0377-2217(96)00222-6.

[24] Ray, T., Saini, P. (2007). "Engineering design optimization using a swarm with an intelligent information sharing among individuals". *Journal of Engineering Optimization*, 33(6), 7350-748, doi:https://doi.org/10.1080/03052150108940941.

[25] Ruefli, T.W. (1990). "Mean-variance approaches to risk-return relationships in strategy: Paradox lost". *Management Science*, 36(3), doi:https://doi.org/10.1287/mnsc.36.3.368.

[26] Semenov, A. (2021). “Measuring the stock’s factor beta and identifying risk factors under market inefficiency”. *The Quarterly Review of Economics and Finance*, 80, 635-649, doi:https://doi.org/10.1016/j.qref.2021.03.014.

[27] Saridakis, K.M., Dentsoras, A.G. (2008). “Soft computing in engineering design-A review”, *Applied Soft Computing*, 22(2), 202-221, doi:https://doi.org/10.1016/j.asoc.2023.111098.

[28] Szegö, G. (2002). “Measures of risk”. *Journal of Banking & Finance*, 6(7), 1253-1272, doi:https://doi.org/10.1016/j.ejor.2003.12.016.

[29] Tuy, H. (1992). “On nonconvex optimization problems with separated nonconvex variables”. *Journal of Global Optimization*, 2, 133-144, doi:https://doi.org/10.1007/BF00122050.

[30] Xiong H., Wu, Z., Fan, H., Li, G., Jiang, G. (2018). “Quantum rotation gate in quantum-inspired evolutionary algorithm: A review, analysis and comparison study”. *Swarm and Evolutionary*, 42, 43-57, doi:https://doi.org/10.1016/j.swevo.2018.02.020.