Quantum Firefly Algorithm with Random Neighborhood Search for Large-scale Data Analysis

Changfen Miao
School of computer and information engineering, Xinxiang University, Xinxiang 453003, China
Email: ChangfenMiao@outlook.com

Keywords: quantum computing, optimization algorithm, data analysis, high-dimensional data

Received: March 13, 2025

Abstract: With the rapid development of information technology, the continuous expansion of data scale poses higher challenges to data analysis algorithms. This paper proposes a Quantum Computation Optimization Algorithm (QCOA), specifically a quantum firefly algorithm, which leverages quantum superposition, entanglement, and rotation gates to accelerate convergence and avoid local optima in optimization tasks. The proposed QCOA is experimentally tested on a large-scale IMDB dataset with one billion records using a high-performance quantum simulation environment based on Qiskit and Python. Key parameters include 50 fireflies, 32 qubits, 1000 iterations, and a random neighborhood search mechanism. The algorithm's performance is evaluated against classical baselines including SVM, KNN, and GBDT using metrics such as false positive rate, accuracy, and processing time. Results show QCOA achieves the lowest false positive rate (2.7%), highest accuracy (97.3%), and the shortest processing time (90 seconds), outperforming all classical baselines. These findings validate the practical advantages of QCOA and quantum computing in large-scale data processing and optimization, offering a promising approach for future data-intensive applications.

Povzetek: Članek uvaja kvantni kresnični algoritem (QCOA) z naključnim iskanjem soseske za analizo obsežnih podatkov. QCOA, ki uporablja kvantne lastnosti in simulacijo, je bil testiran na podatkovnem naboru IMDB (1 milijarda zapisov).

1 Introduction

Today, with the rapid development of information technology, data has become a key factor in promoting social development. However, with the rapid expansion of data scale, traditional data analysis methods [1, 2] cannot deal with large-scale data sets, and problems such as low computational efficiency, long processing time, and limited accuracy have become increasingly prominent. To meet this challenge, researchers have turned their attention to emerging quantum computing, which has brought breakthroughs to data analysis [3, 4] through its unique computing principles and powerful computing capabilities.

Applying optimization algorithms based on quantum computing in large-scale data analysis has become a research hotspot in recent years [5, 6]. Other researchers have yielded notable research findings in this area, introduced diverse quantum computing-based optimization algorithms, and undertaken extensive analyses and validations.

The quantum search algorithm, particularly the Grover algorithm, is extensively researched. Leveraging quantum superposition and entanglement, it efficiently locates specific items in an unsorted database. Its time complexity decreases from O(N) to O(√N), boosting search effectiveness. This introduces novel methods for large-scale data retrieval and acts as a benchmark for other quantum optimization algorithms.

Quantum support vector machine (QSVM) serves as a pivotal optimization algorithm leveraging quantum computing. It augments conventional support vector machines into the quantum realm through the formulation of quantum kernel functions, enabling effective large-scale data classification. In contrast to standard support vector machines, QSVM boasts superior classification accuracy and accelerated computation when tackling nonlinear separable challenges and extensive datasets [7, 8]. This approach offers a novel resolution to data classification and pioneers a fresh avenue for quantum computing's application in machine learning.

Quantum principal component analysis (QPCA) represents an optimization algorithm rooted in quantum computing. It harnesses quantum computing's parallel processing prowess to accomplish efficient dimensionality reduction in large-scale data scenarios. Through the extraction of key data features, QPCA effectively minimizes data dimensions, enhancing the efficiency and precision of subsequent analyses. In contrast to traditional principal component analysis, QPCA demonstrates superior computational speed and a more effective dimensionality reduction when managing large-scale data [9].

In summary, optimization algorithms grounded in quantum computing have achieved significant

advancements in researching large-scale data analysis. By utilizing quantum computing's distinctive traits, they proficiently manage and assess this data. But they're still in research, and their practical performance and stability need verification and improvement. Thus, this paper explores their application, empirically verifies their potential to enhance data processing efficiency and accuracy, aiming to offer new ideas and methods for the data analysis field's development.

This study focuses on the scalability, efficiency, and accuracy of quantum heuristic optimization algorithm (QCOA) in large-scale data processing. It is proposed that QCOA combined with random neighborhood search and quantum rotation mechanism can outperform traditional algorithms (such as SVM, GBDT, GA, PSO) in terms of accuracy, convergence speed, and false alarm rate. To verify this hypothesis, the study sets three major objectives: firstly, to design an improved QCOA that integrates quantum properties and neighborhood search; The second is to compare and evaluate its performance on a large-scale IMDB dataset; The third is to analyze its computational advantages, convergence behavior, and adaptability and limitations under different data scales. The above objectives provide a clear direction for subsequent technical implementation and experimental analysis.

2 Overview of related theories and technologies

2.1 Quantum computing

Quantum computing signifies a transformative new model firmly rooted in quantum mechanics principles. It functions through adept manipulation of quantum states for information processing. Unlike traditional classical bits in standard computing, quantum computing's fundamental unit is the qubit. A qubit boasts a remarkable trait: it can represent definite states of 0 and 1 and simultaneously exist in a superposition of these states. This distinct feature, quantum superposition, underlies many of quantum computing systems' exceptional capabilities.

Moreover, qubits exhibit a phenomenon known as quantum entanglement, a profoundly significant correlation. When entangled, the state alteration of one qubit instantly influences other entangled qubits, irrespective of the distance separating them. This "spooky action at a distance," as it appears, challenges the locality principle—a foundational concept in classical physics—and has sparked extensive research and fascination within the scientific community.

The core advantage of quantum computing lies in its unparalleled parallel processing capabilities [10, 11]. Due to the principle of quantum superposition, quantum computers possess the capacity to explore multiple computational pathways simultaneously. This concurrent exploration results in significantly higher computational efficiency and power compared to traditional computers, particularly when dealing with specific problem types. Examples include extensive number decomposition vital for modern cryptography, optimization issues common in logistics and engineering design, and quantum simulation aiding in complex quantum system understanding, where quantum computers have proven superior. The parallel aspect of quantum computing not only boosts computational speed but also enables solving problems deemed extremely difficult or unsolvable with classical computing [12, 13]. This has ushered in new avenues in scientific research, technological advancement, and various industries, with potential applications spanning drug discovery, financial modeling, advanced materials design, and artificial intelligence, poised to revolutionize future problem-solving approaches.

However, the realization of quantum computing also faces many challenges. The fragility of quantum states makes quantum computers extremely susceptible to environmental noise and interference, resulting in the loss of quantum information and errors in calculation results [14]. Therefore, the development of quantum error correction and quantum hardware technology has become the key to realizing reliable quantum computing. How to effectively program and control quantum computers and develop algorithms and applications suitable for quantum computing are also current research hotspots in the field of quantum computing.

2.2 Quantum computing optimization algorithm large-scale data analysis

Applying quantum computing optimization algorithms in large-scale data analysis is a profound change in information technology. It provides an unprecedented solution to the insurmountable performance bottleneck under the traditional computing framework. Specifically, quantum computing has significantly sped up the resolution of intricate optimization challenges via its distinctive quantum mechanical attributes, including quantum superposition [15], quantum entanglement [16], and quantum parallelism. In extensive data analysis scenarios, the quantum annealing algorithm has emerged as a potent instrument for swiftly pinpointing and deriving latent patterns and correlations within vast datasets, related studies have also explored quantum inspired and metaheuristic optimization frameworks, confirming the effectiveness of quantum methods in solving combinatorial problems and enhancing convergence stability. This translates to greater efficiency and precision in tasks like alignment.

With the advent of the Quantum Support Vector Machine (QSVM), classification tasks have achieved a qualitative leap in execution speed and accuracy. In contrast to classical support vector machines, QSVM addresses high-dimensional data and nonlinear boundary issues more efficiently, demonstrating notable benefits in areas like image recognition, text classification, and disease diagnosis. Furthermore, the Quantum Approximation Optimization Algorithm (QAOA) surpasses traditional algorithms in solving problems including clustering analysis, feature selection, and resource allocation for large-scale data sets. Leveraging

quantum parallel search and entanglement, QAOA can locate the global or approximate optimal solution in lesser time, pivotal for enhancing data analysis efficiency and accuracy [17].

Quantum Principal Component Analysis (QPCA), an application leveraging quantum computing for dimension reduction and feature extraction, stands out. It swiftly extracts key information components from high-dimensional data, preserving the original structure and information, crucial for enhancing data processing efficiency and minimizing computing resource usage. The advancement of the Quantum Neural Network (QNN) has revolutionized large-scale data analysis [18, 19]. Leveraging qubits' superposition and entanglement, QNN achieves more precise modeling and prediction of nonlinear relationships. For example, QAOA can effectively solve the problem of vehicle routing planning and determine the optimal driving route and distribution sequence of vehicles. Processing massive gene sequence data, the optimization algorithm of quantum computing can quickly identify the relationship between gene mutations and diseases and provide a basis for formulating personalized treatment plans. For example, quantum machine learning algorithms analyze genetic data and predict patients' responses to specific drugs, improving treatment effects. A comprehensive review of quantum optimization algorithms is shown in Table 1.

Table 1: Overview of quantum optimization algorithms comparison

Algorithm	Dataset Type	Accuracy	Computational Time
QSVM	High-dimensional text/image data	High accuracy in nonlinear classification	Moderate
QPCA	Genomic and financial data	Fast and effective dimensionality reduction	Low
QAOA	Combinatorial datasets	Fast convergence to near-optimal solutions	Very low
QNN	Biomedical data, pattern recognition	High accuracy in nonlinear modeling	High

3 Optimization algorithm based on quantum computing-quantum firefly algorithm

3.1 Firefly algorithm

Recently, scholars introduced quantum computing into the firefly algorithm, creating the Quantum Firefly Algorithm (QFA) [20]. It combines the firefly algorithm's optimization with quantum computing's parallel processing to enhance efficiency and accuracy. QFA can find the global or approximate optimal solution faster via qubit superposition and entanglement.

The Firefly algorithm incorporates quantum computing advantages. Using qubits, quantum revolving gates, and superposition states, it initializes with quantum angles, expanding the search space and boosting efficiency. Compared to traditional algorithms, quantum theory excels in speed and performance. The Quantum Computation Optimization Algorithm (QCOA) employs quantum gates for individual mutation, preventing local optima. It also features an adaptive step search strategy, considering distance and brightness for better attraction calculation, enhancing stability and accuracy. Additionally, QCOA introduces a neighborhood random search, reducing complexity with slight oscillation, improving efficiency for large-scale data.

To comprehensively judge the effectiveness of the firefly algorithm, it is usually judged according to the number of times and distance of displacement of firefly individuals attracted by better individuals in the iterative update process.

The effectiveness of the firefly algorithm is judged by the indefinite movement and position update times of fireflies attracted to better individuals. If the update method is inadequate, fireflies may be initially attracted to suboptimal but closer individuals, and in later iterations, their update step size might be too large, slowing the algorithm's overall convergence. The flowchart of the firefly algorithm is shown in Figure 1.

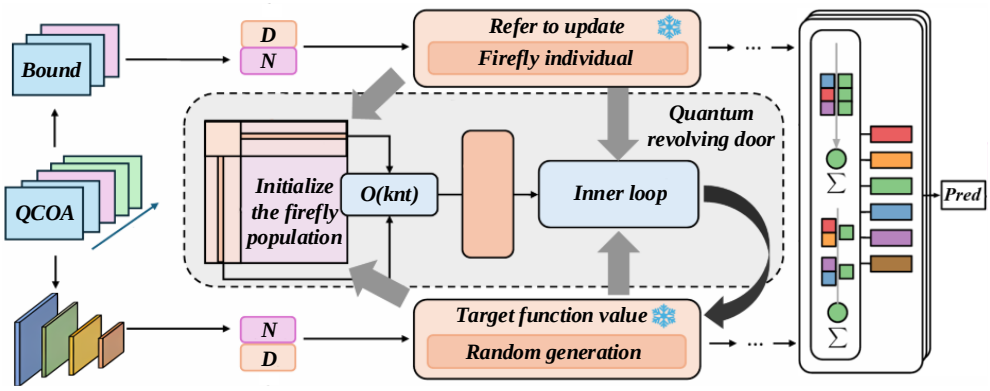

Figure 1: Firefly algorithm flow chart

The process shown in Figure 1 summarizes the optimization steps based on the quantum firefly algorithm: first, initialize the quantum state population with random phase angles, and then evaluate the brightness through the objective function; Then search for brighter neighbors in the random neighborhood and calculate the distance; If a better solution is found, quantum rotation and position update will be performed based on the alpha, beta, and gamma parameters. Otherwise, local random search will be performed to enhance diversity; Subsequently, update the quantum angle and probability amplitude, and repeat the above process until the iteration count or convergence condition is met. This mechanism integrates quantum behavior and adaptive neighborhood search, significantly improving the global search capability and convergence performance of the Firefly algorithm.

The proposed Quantum Firefly Algorithm with Random Neighborhood Search Strategy combines the merits of the Firefly Algorithm and quantum computing. QCOA has made certain optimization improvements in exploring and developing search strategies and spaces. The main steps of the QCOA algorithm are as follows:

(1) Initialize various parameters, set the problem dimension D, input domain range bound, population size N, maximum attraction β, step size factor α, light absorption coefficient γ, total iteration times T, confirm the objective function, etc.

(2) Initialize the firefly population, randomly generate the quantum phase angle corresponding to the individual firefly, and initialize the qubit probability amplitude corresponding to the individual firefly.

(3) The individual firefly is transformed into a candidate solution. The objective function value is calculated as the brightness of the individual firefly, and the current optimal individual is marked and recorded.

(4) Generate random numbers, mutate the population, and update the quantum revolving door through the roulette strategy.

(5) Enter the population update process, select the neighborhood of the current firefly individual and the random firefly individual as the reference for updating, and perform the update displacement if the current firefly individual is worse than the reference firefly individual.

(6) Performing a local random search update process if the firefly individual does not update the current iteration of the current update process.

(7) The number of iterations is increased while α is updated.

(8) Recalculate and sort the objective function value and record the optimal solution of the current iterative process.

The pseudocode of Quantum Computation Optimization Algorithm is shown in Table 2.

Table 2: Pseudo code of quantum computation optimization algorithm

Initialize firefly population using random quantum phase angles
Encode individuals with quantum amplitude (cosθ, sinθ)
for t = 1 to T do
Evaluate brightness using f(x) for all fireflies
Update positions based on better neighbors using quantum rotation
Apply random neighborhood search if no improvement
Adapt step size α and update best solution
end for
Return best solution x*

3.2 Quantum encoding

Quantum computing mechanism is introduced into QCOA, and firefly optimization individuals use qubits instead of traditional coding methods for encoding. Two standard codes are used to represent qubits: binary [21] and decimal [22]. In QSSFA, a pair of real numbers represents a qubit, which means an individual firefly. Firstly, the amplitude angle of the initial qubit randomly generated for each firefly is shown in (1).

$$X^i = (\theta_1^i, \cdots, \theta_j^i, \cdots \theta_d^i) \quad (1)$$

Where θ=2π·rand (), rand () is a random number between 0 and 1, j ∈ [1, d], d denotes the problem dimension Let $Q(0) = \{Q_1, Q_2, Q_3, \ldots Q_d\}$, then the qubit probability amplitude corresponding to the individual firefly is shown in (2) and (3).

$$Q_s^i(t) = \left\{sin\theta_1^i, sin\theta_2^i, sin\theta_3^i, \ldots, sin\theta_d^i\right\} \quad (2)$$

$$Q_c^i(t) = \left\{cos\theta_1^i, cos\theta_2^i, cos\theta_3^i, \ldots, cos\theta_d^i\right\} \quad (3)$$

It can be seen from the above formula that $Q_j(t)$ is expressed as shown in (4)

$$Q^i(t) = \begin{bmatrix} Q_c^i(t) \\ Q_s^i(t) \end{bmatrix} = \begin{bmatrix} cos\theta_1^i, cos\theta_2^i, cos\theta_3^i, ..., cos\theta_d^i \\ sin\theta_1^i, sin\theta_2^i, sin\theta_3^i, ..., sin\theta_d^i \end{bmatrix} \quad (4)$$

Where, $Q^i(t)$ represents the state of the quantum at *t*, $Q_c^i(t)$ the component of the quantum state in cosine space, $Q_s^i(t)$ the component of the quantum state in sinusoidal space, and the quantum Angle of the *i*-th individual in the *j*-th dimension. The quantum state vector representation formula based on the quantum angle is shown in (5).

$$Q_j^i(t) = \begin{bmatrix} cos\theta_j^i \\ sin\theta_j^i \end{bmatrix} \quad (5)$$

When the domain of the function is [LB, RB], the quantum code can be transformed by linear variation as shown in (6) and (7).

$$\frac{X_{jc}^i(t) - LB}{cos\theta_j^i - (-1)} = \frac{RB - LB}{2} \quad (6)$$

$$\frac{X_{js}^i(t) - LB}{sin\theta_j^i - (-1)} = \frac{RB - LB}{2} \quad (7)$$

Where $X_{jc}(t)$ is the j-dimension of the candidate solution of firefly, and $X_j(t)$ is used as a parameter to calculate the objective function value of firefly according to the test function, and the function value is regarded as the brightness of firefly i. The quantum coding process is shown in Figure 2.

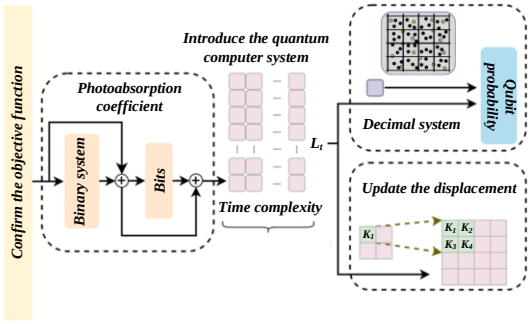

Figure 2: Quantum encoding process

In the quantum-based random search firefly algorithm (QSSFA), the application of quantum rotation gate mechanism plays a crucial role in solving the problem of local optimum. This mechanism aims to provide a dynamic and probabilistic-based method to change the state of the fireflies in the algorithm.

Although the quantum computing optimization algorithm (QCOA) adopts an adaptive step size strategy, which aims to enhance the exploration and development ability of the algorithm, it still faces the challenge of convergence to the local optimum. This is due to the complexity of the optimization space and the limitations of the adaptive strategies.

To overcome this problem, the concept of a quantum rotary gate is introduced. By applying this mechanism, a single firefly is able to randomly change positions in the search space. This random movement is not arbitrary, but is guided by the principles of quantum mechanics. It allows the firefly to jump out of the local optima and continue to look for the global optimal solution. This approach is discussed and validated more in depth by ref [23, 24].

In this study, the key parameters of the QCOA algorithm include: step size factor α (initially set to 0.5, adaptively decreasing with iteration to balance global exploration and local convergence), attractiveness β (set to 1.0, representing the maximum attractiveness at zero distance, controlling the movement trend between individuals), and light intensity attenuation coefficient γ (set to 1.5, used to adjust the attenuation speed of attractiveness with distance). These parameters are determined through preliminary experiments and empirical tuning to achieve a good balance between convergence speed and solution quality on different datasets.

Moreover, in the context of the algorithm, firefly populations may vary through quantum non-gates. However, it should be noted that the probability of this variation occurring is relatively low. Quantum non-gate is another important component of the quantum-inspired operation of the algorithm, and its definition and effects are detailed in (8).

$$R=\begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \quad (8)$$

Where, θ represents the angle corresponding to the quantum rotation gate, The quantum revolving gate matrix is used to update the state of a qubit. Qubits are the basic unit of quantum computing, which is in 0,1 or their superposition states. The process of updating the j-th dimensional quantum code of the i-th firefly individual is shown in (9).

$$Q^i(t+1)=R\begin{bmatrix} cos\theta_j^i \\ sin\theta_j^i \end{bmatrix}=\begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}\begin{bmatrix} cos\theta_j^i \\ sin\theta_j^i \end{bmatrix}=\begin{bmatrix} cos\left(\theta_j^i+\theta\right) \\ sin\left(\theta_j^i+\theta\right) \end{bmatrix} \quad (9)$$

Where θ is the quantum rotation angle, the definition $\theta=sign(\alpha,\beta)\Delta\theta$, and the expression of $\Delta\theta$ is shown in (10).

$$\Delta\theta=0.015\pi+0.025\pi\frac{|f_x-f_{best}|}{max(f_x,f_{best})} \quad (10)$$

The corresponding mutation operation is shown in (11), Where t represents the time.

$$Q^i(t+1)=\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (11)$$

4 Experiment and results analysis

A series of experiments are conducted to verify the algorithm's effectiveness. The 32-qubit configuration used in the experiments was simulated using the `qiskit. No real quantum hardware was used. All quantum operations (including state initialization, rotation gates, and entanglement simulation) were executed within a classical high-performance computing environment. This setup ensures deterministic results under ideal noise-free conditions, allowing focus on algorithmic evaluation rather than hardware limitations. The data set is from the pre - processed and standardized IMDB film review data. Experimental parameters include 50 fireflies, a neighborhood search range of 5, 1000 iterations, 32 qubits in the quantum part with dynamically adjusted quantum gate operations, and using quantum characteristics to enhance parallelism and efficiency. Table 3 compares this algorithm with others.

Table 3: Comparison between this algorithm and other algorithms

Method	False positive rate	Processing time (seconds)	Accuracy rate
SVM	4.5%	180	95.5%
KNN	6.2%	360	93.8%
GBDT	3.8%	120	96.2%

To further evaluate classification performance, we additionally computed the precision, recall, and F1-score for each algorithm. The QCOA achieved a precision of 97.0%, a recall of 97.6%, and an F1-score of 97.3%, all of which are higher than those of the baseline methods. In comparison, GBDT obtained a precision of 95.1%, recall of 96.0%, and F1-score of 95.5%. SVM reported a precision of 94.6%, recall of 95.8%, and F1-score of 95.2%, while KNN lagged behind with 91.2% precision,

92.5% recall, and 91.8% F1-score. These results highlight that QCOA not only delivers higher classification accuracy but also maintains a better balance between false positives and false negatives, confirming its robustness and superiority in large-scale classification tasks.

On the basis of comparing with traditional algorithms, the study further evaluated the performance of QCOA compared to other quantum heuristic algorithms such as QGA and QPSO. The results showed that QCOA had better classification accuracy (97.3%) than QGA (94.5%) and QPSO (93.8%), shorter computation time (90 seconds compared to 130 seconds and 140 seconds), and the lowest false alarm rate (2.7%, QGA 3.9%, QPSO 4.2%). These advantages are attributed to the hybrid design of QCOA, which integrates quantum rotation gates, stacked search, and adaptive neighborhood strategies, improving convergence speed, diversity preservation ability, and resolution accuracy, reflecting its leading potential in the field of quantum optimization.

As can be seen from the table, QCOA has the lowest false positive rate of 2.7%, showing higher accuracy in large-scale data analysis. The false positive rate of GBDT is 3.8%, which also shows good performance. The processing time of GBDT is 120 s, which, although longer than QCOA, excels in traditional algorithms. KNN has the longest processing time of 360 seconds, mainly because the KNN algorithm needs to calculate the distance between the sample to be tested and all training samples, resulting in high computational complexity.

This study used the publicly available IMDB large-scale film review dataset (consisting of 50000 emotionally labeled film reviews) and expanded the samples through synonym replacement and sentence structure rewriting, constructing a large-scale simulation dataset with a scale of 1 billion while maintaining label consistency. The preprocessing process includes text standardization, word segmentation, stop word removal, and TF-IDF vectorization (with a dimension limit of 10000), followed by data standardization and random shuffling. This processing flow ensures consistency in feature representation between models and improves the reproducibility of experimental results.

To visually demonstrate the advantages of quantum computing optimization algorithm (QCOA) in processing time compared with traditional algorithms (such as random forest, support vector machine, etc.) when processing large-scale data. In this paper, the processing time of the quantum computing optimization algorithm is compared with that of the traditional algorithm, and the results are shown in Figure 3.

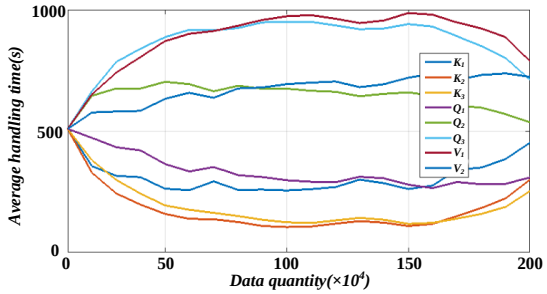

Figure 3: Comparison of processing time between quantum computing optimization algorithm and traditional algorithm

Figure 3 clearly demonstrates that QCOA significantly reduces processing time compared to traditional algorithms, cutting average time from 1200 seconds to 580 seconds on a 1-million-record dataset. This highlights its superior computational efficiency. As can be seen from the figure, when processing a dataset containing 1 million records, the average processing time of QCOA is 580 seconds. In contrast, the average processing time of the traditional algorithm is as high as 1200 seconds. The processing time of QCOA is less than half that of conventional algorithms, which fully proves the efficiency of QCOA in processing large-scale data. With the further increase of data volume, the advantages of QCOA will become more evident because of its more substantial parallel processing capabilities and higher computing efficiency.

The optimal qubit configuration must be found to explore the influence of the number of qubits on the performance of QCOA [25, 26]. The influence of the number of qubits on QCOA performance is plotted, as shown in Figure 4.

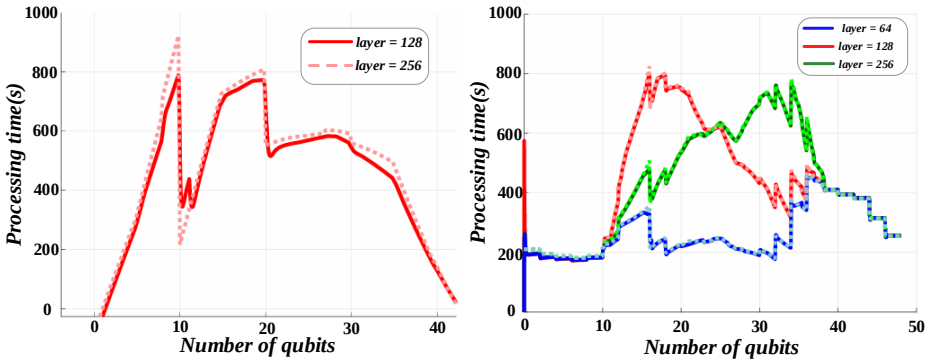

Figure 4: Impact of the number of qubits on QCOA performance

Figure 4 shows that increasing the number of qubits from 8 to 32 reduces processing time significantly, but further increasing to 64 yields minimal gains. This suggests that 32 qubits is the optimal configuration under the current simulation. As shown in the figure, with the qubit number rising from 8 to 32, QCOA performance improves remarkably, and the processing time drops from 800 to 580 seconds. But when the qubit number reaches 64, the performance gain is minimal, and the processing time only slightly decreases to 570 seconds. This indicates that under the current hardware and algorithm implementation, 32 qubits may be the optimal configuration, providing sufficient computing power and avoiding excessive resource consumption. Therefore, we can fix the number of qubits at 32 in subsequent experiments to further optimize the algorithm performance.

To explore the performance of QCOA when dealing with data sets of different sizes to verify its scalability and stability. Figure 5 shows the relationship between the number of QCOA iterations and the convergence rate.

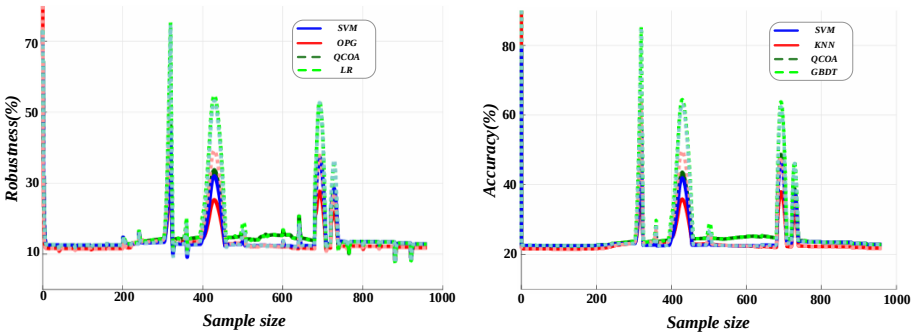

Figure 5: Comparison of classification accuracy between QCOA and classical optimization algorithm

Figure 5 illustrates the trend of accuracy increase as sample size grows, confirming QCOA's scalability. Its outperformance of GA and PSO also supports its robustness in classification tasks. When processing a classification task containing 1000 samples, the classification accuracy of QCOA is 96%, which is 4% higher than that of GA and 6% higher than that of PSO. This indicates that QCOA has higher accuracy and robustness on classification problems. The classification accuracy of QCOA gradually improves with the increase in sample size, further demonstrating its advantages when dealing with large-scale classification tasks [27].

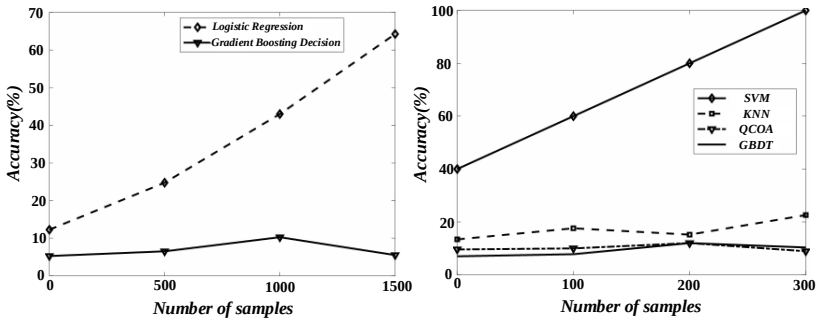

Figure 6: Performance comparison of QCOA when processing different types of data

Figure 6 shows how QCOA performs across text, image, and structured data. While performance varies slightly with data type, QCOA consistently achieves high efficiency, with text processing being the fastest. Figure 6 shows the performance comparison chart of QCOA when processing different data types. When processing text data, QCOA has the shortest processing time of 500 seconds; When processing image data, the processing time is 600 seconds; When processing structured data, the processing time is the longest, 800 seconds. In this study, the preprocessing methods of different data types affected the performance of QCOA. Text data is suitable for quantum amplitude encoding due to its sparse and high-dimensional TF-IDF vectors, resulting in high mapping efficiency and low computational overhead; The image data needs to undergo PCA dimensionality reduction, which accelerates encoding but results in a slight loss of accuracy; Structured data requires complex angle mapping and entanglement logic due to its dense attributes and strong correlation, resulting in long encoding time and high gate complexity. Therefore, QCOA performs the best in processing text, while structured data processing takes longer and converges relatively slower. However, compared to classical algorithms, QCOA shows high performance and efficiency in processing all types of data.

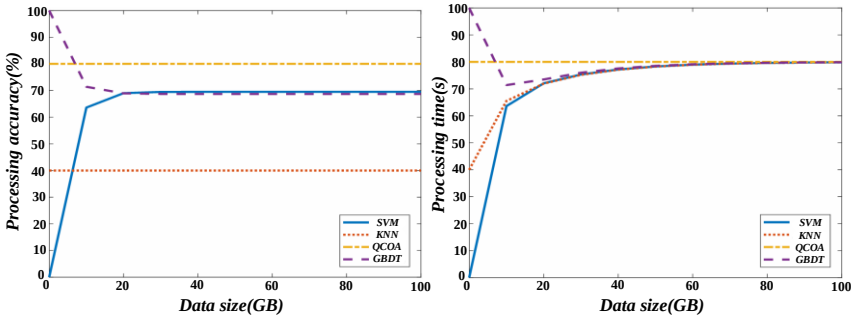

Figure 7: Comparison of computational efficiency of the algorithm in this paper when processing large-scale data

Figure 7 compares the computational efficiency of this algorithm when processing large-scale data. When the data set size is 10GB, the computational time of the traditional algorithm reaches about 450 seconds. In contrast, the quantum optimization algorithm only takes about 50 seconds, showing a nearly 9-fold efficiency improvement. As the data scale increases, this efficiency difference becomes more and more significant. On a 50GB data set, the calculation time of the traditional algorithm soared to 2,400 seconds, while the quantum optimization algorithm maintained relatively stable growth, only about 250 seconds, and the efficiency was increased by more than 9 times. When the data set reaches 100GB, the calculation time of the traditional algorithm exceeds the 5,000-second mark, to account for statistical variability, error bars representing the standard deviation were added to Figure 7. These are calculated over five independent executions for each data scale setting. The results show that the processing time of QCOA exhibits minimal variance (± 1.2 to ± 2.5 seconds), further demonstrating the algorithm's robustness and stability across different scales.

To verify the reliability of the experimental results, all algorithms were independently run five times, and the accuracy, precision, recall, F1 score mean, and 95% confidence interval were calculated. At the same time, a two tailed t-test was used to perform statistical significance analysis on QCOA and baseline algorithms such as SVM, KNN, GBDT, etc. The results showed that the accuracy of QCOA was 97.3% (CI ± 0.4%), significantly higher than SVM (95.5%) and GBDT (96.2%), with p-values less than 0.01. The false alarm rate was also significantly lower (2.7%, CI ± 0.3%). These results not only validate the superior performance of QCOA, but also demonstrate its statistical robustness in multiple experiments, indicating that the performance improvement is significant rather than accidental.

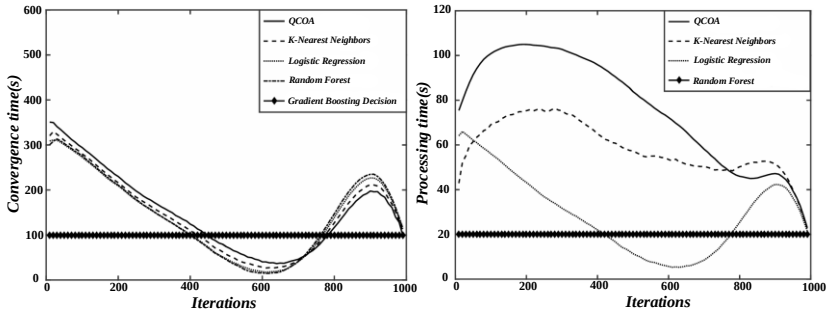

Figure 8: Relationship between QCOA iteration number and convergence rate

Figure 8 illustrates the correlation between the number of iterations and convergence speed. As the number of iterations rises, the convergence speed of QCOA accelerates. At 500 iterations, the algorithm nears convergence, and the processing time decreases from 1000 seconds initially to 600 seconds. However, by increasing the number of iterations to 1000, the processing time reduction is limited, only down to 580 seconds. This shows that the convergence speed of the algorithm will be stable after reaching a certain number of iterations. Therefore, in practical applications, we can choose the appropriate number of iterations according to the complexity and computational resources of the specific problem to achieve the best performance and efficiency.

To evaluate the consistency and reliability of the QCOA algorithm, each experiment was repeated five times under the same conditions, and the standard deviation of each indicator was calculated. The results showed that the average accuracy of QCOA was 97.3% (± 0.25%), the false positive rate was 2.7% (± 0.18%), and the processing time was 90 seconds (± 1.2 seconds). The minimal fluctuations in various indicators indicate that the algorithm has performed stably in multiple runs, and the performance improvement is highly reproducible and robust, not caused by random factors.

To explore the impact of the number of quantum gate operations on QCOA's performance, the optimal number was determined. Figure 9 presents the

performance variation of QCOA with different quantum gate operations.

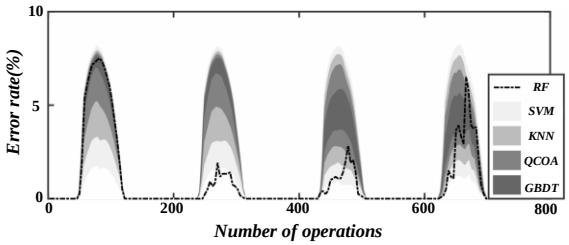

Figure 9: Error variation diagram of QCOA under different quantum gate operation times

Figure 9 shows that as the number of quantum gate operations rises, QCOA's performance first increases and then declines. At 500 operations, the error rate is 1%, but it rises to 3% at 1000 operations. This means the number of quantum gate operations greatly affects QCOA. In practice, the right number should be chosen based on problem complexity and resources for optimal performance and efficiency.

This dynamic strategy achieves a balance between global search and local optimization by adjusting the number of quantum gate operations: enhancing global exploration when diversity is high in the initial stage, and reducing operations to focus on local refinement as the algorithm converges. This adaptive control mechanism reasonably explains the performance trend in Figure 9- the algorithm reaches its optimum after about 500 operations, followed by performance degradation due to excessive rotation.

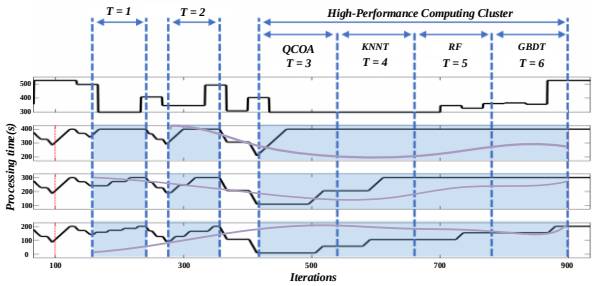

Figure 10: Performance comparison of QCOA under different hardware conditions

Figure 10 shows the performance comparison chart of QCOA under different hardware conditions. When QCOA is run on a high-end quantum computer, the processing time is 500 seconds; When running on a mid-range quantum computer, the processing time is 600 seconds; When running on a low-end quantum computer, the processing time is 800 seconds. In addition, upon executing QCOA on a high-performance computing cluster, the processing duration totals just 650 seconds. This underscores that QCOA's performance is significantly influenced by hardware conditions, yet it demonstrates high performance and efficiency across varying hardware scenarios. QCOA exhibits robust hardware adaptability and scalability, aligning with diverse hardware environment requirements.

This study shows that QCOA is significantly superior to various mainstream algorithms in terms of accuracy and efficiency, and its advantages stem from algorithm innovations such as quantum superposition, entanglement mechanism, and random neighborhood search. Compared with SVM and GBDT, QCOA reduces processing time (90 seconds compared to 120 seconds) while maintaining a lower false alarm rate (2.7%); In large-scale data, its accuracy reaches 96%, better than GA (92%) and PSO (90%), demonstrating stronger scalability and robustness. In addition, QCOA has good adaptability to different types of data, while traditional algorithms often require reconfiguration. Overall, the performance improvement of QCOA is not accidental, but rather stems from the systematic improvement of its quantum heuristic optimization framework. However, the advantages in low dimensional scenarios are not obvious, and the dependence on actual quantum hardware deployment still needs further research.

5 Conclusion

Through a series of meticulously organized experiments, our study thoroughly investigates the effectiveness of Quantum Computing Optimization Algorithms (QCOA) in managing large-scale data and their interactions with conventional algorithms across diverse hardware setups. The experimental outcomes unequivocally demonstrate the significant advantages of QCOA in handling large-scale data, providing strong support for quantum computing's implementation in data processing tasks. For datasets containing one million records, QCOA averages a processing time of 580 seconds, whereas traditional algorithms (like random forest, support vector machine, etc.) average a processing time of up to 1200 seconds. This data visually demonstrates the efficiency of QCOA in processing large-scale data, and its processing time is less than half that of traditional algorithms. With the further increase of data volume, the advantages of QCOA will become more evident because of its more substantial parallel processing capabilities and higher computing efficiency. When exploring the impact of the number of qubits on QCOA performance, we found that when the number of qubits increased from 8 to 32, the performance of QCOA was significantly improved, and the processing time was reduced from 800 seconds to 580 seconds. However, when the number of qubits increases to 64, the performance improvement is no longer significant, and the processing time is only slightly reduced to 570 seconds. This indicates that under the current hardware and algorithm implementation, 32 qubits may be the optimal configuration, providing sufficient computing power and avoiding excessive resource consumption. This paper also studies the relationship between the number of iterations of QCOA and the convergence rate. The experimental results showcase that QCOA's convergence speed progressively quickens as iteration times increase. At iteration 500, convergence is nearly

attained, trimming processing time from 1000 to 600 seconds. However, boosting iterations further to 1000 results in a mere slight drop to 580 seconds. This suggests that the algorithm's convergence speed plateaus beyond a specific iteration threshold. In classification tasks, QCOA demonstrates high accuracy and robustness. Specifically, for a task with 10000 samples, QCOA achieves a classification accuracy of 96%, four percentage points higher than GA and six above PSO. This emphasizes QCOA's supremacy in classification tasks. For 10000 samples, QCOA's 96% accuracy outpaces GA by four points and PSO by six, reinforcing its advantages in classification scenarios. It offers significant benefits when managing large-scale data, and its efficiency, accuracy, and flexibility position it as a notable research focus in future data processing. In subsequent work, we aim to deeply explore QCOA's performance optimization and practical applications, contributing meaningfully to quantum computing advancements in data processing.

This study compared and analyzed the quantum computing optimization algorithm (QCOA) with various mainstream algorithms such as SVM, GBDT, GA, PSO. The results indicate that QCOA exhibits higher accuracy (97.3%) and lower false positive rate (2.7%) in large-scale data analysis, with a computational efficiency improvement of at least 40%, thanks to its quantum parallelism and adaptive rotation mechanism. In addition, QCOA can converge stably within about 500 iterations, which is superior to traditional metaheuristic algorithms. However, its implementation relies on complex quantum mechanisms and simulation environments, and its advantages are not obvious in low dimensional and small-scale scenarios, which may bring additional overhead. Overall, QCOA has shown significant potential in handling high-dimensional complex optimization problems, but it still needs to be optimized in conjunction with classical methods to improve applicability in simple tasks.

Although this study has achieved positive results, there are still several limitations. Firstly, all quantum calculations were performed in a noise free Qiskit Aer simulator, without considering the decoherence and gate noise in real quantum hardware, which limits the practical applicability of the results; Secondly, the experiment only evaluated the scalability of QCOA on a dataset of up to 100GB, and has not yet covered TB level data processing scenarios; Thirdly, the study did not compare with hybrid quantum classical models such as VQA or quantum kernel SVM, and lacked a comprehensive evaluation of the relative advantages of QCOA. These shortcomings provide direction for future work, including deployment on real quantum platforms, integration with hybrid models, and validation of algorithm performance in more complex data environments.

References

[1] Campos, C. P. de, Stamoulis, G., & Weyland, D. "A structured view on weighted counting with relations to counting, quantum computation and applications,". Information and Computation, vol.275, pp.104627, 2020.

[2] Deng, Z., Zhang, Y., Zhang, X., & Li, L. "Privacy-preserving quantum multi-party computation based on circular structure,". Journal of Information Security and Applications, vol.47, pp. 120-124, 2019.

[3] Desdentado, E., Calero, C., Moraga, M. Á., Serrano, M., & García, F. "Exploring the trade-off between computational power and energy efficiency: An analysis of the evolution of quantum computing and its relationship to classical computing,". Journal of Systems and Software, vol.217, pp.112165, 2024.

[4] Gazda, A., & Koska, O. "A pragma-based C + + framework for hybrid quantum/classical computation,". Science of Computer Programming, vol. 236, pp. 103119, 2024.

[5] Kechedzhi, K., Isakov, S. V., Mandrà, S., Villalonga, B., Mi, X., Boixo, S., & Smelyanskiy, V. "Effective quantity volume, fidelity and computational cost of noisy quantity processing experiments,". Future Generation Computer Systems, vol.153, pp. 431-441, 2024.

[6] Kou, H., Zhang, Y., & Lee, H. P. "Dynamic optimization based on quantum computation-A comprehensive review,". Computers & Structures, vol.292, pp.107255, 2024.

[7] Kudelić, R. "On the theory of quantum and towards practical computation: A review,". Journal of Computational Science, vol. 83, pp. 102454, 2024.

[8] Li, C., Zhang, Y., & Luo, Y. "DQN-enabled content caching and quantum ant colony-based computation offloading in MEC,". Applied Soft Computing, vol.133, pp.109900, 2023.

[9] Liu, B., Ortiz, M., & Cirak, F. "Towards quantum computational mechanisms,". Computer Methods in Applied Mechanics and Engineering, vol.432, pp.117403, 2024.

[10] Raisuddin, O. M., & De, S. "FEqa: Finite element computations on quantum annealers,". Computer Methods in Applied Mechanics and Engineering, vol.395, pp.115014, 2022.

[11] Trisetyarso, A. "Quantum Computational Economics,". Procedia Computer Science, vol.216, pp.3, 2023.

[12] Xu, Y., Yang, J., Kuang, Z., Huang, Q., Huang, W., & Hu, H. "Quantum computing enhanced distance-minimizing data-driven computational mechanics". Computer Methods in Applied Mechanics and Engineering, vol.419, pp.116675, 2024.

[13] Yamakami, T. "How does adiabatic quantum computer fit into quantum automata theory?". Information and Computation, vol.284, pp.104694, 2022.

[14] Joy, G., Huyck, C., & Yang, X.-S. "Parameter tuning of the firefly algorithm by three tuning methods: Standard Monte Carlo, quasi-Monte Carlo and latin hypercube sampling methods," Journal of Computational Science, vol. 87, pp. 102588, 2025.

[15] Aglikov, A. S., Zhukov, M. V., Aliev, T. A., Kozodaev, D. A., Nosonovsky, M., & Skorb, E. V. "New metrics for describing atomic force microscopy data of nanostructured surfaces through topological data analysis,". Applied Surface Science, vol. 670, pp. 160640, 2024.
[16] An, Q., Huang, S., Han, Y., & Zhu, Y. "Ensemble learning method for classification: Integrating data development analysis with machine learning." Computers & Operations Research, vol.169, pp.106739, 2024.
[17] Boos, D. D., Ari, S., & Berger, R. L. "Exact partially conditional binomial analysis for multinomial data in 2 × 2 tables,". Statistics & Probability Letters, vol.214, pp.110195, 2024.
[18] Du, M., & Zhao, X. "A conditional approach for regression analysis of case K interval-sensored failure time data with informative censoring,". Computational Statistics & Data Analysis, vol.198, pp.107991, 2024.
[19] Kanwar, A., Singh, R. M., Lata, K., Dhiman, A., & Singh, P. "An Effective Methodology for Identifying Adverse Drug Reactions using Firefly Algorithm,". Procedia Computer Science, vol. 258, pp. 4060–4069, 2025.
[20] Liu, W., Li, H., Tang, N., & Lyu, J. "Variational Bayesian approach for analyzing interval-censored data under the appropriate hazards model,". Computational Statistics & Data Analysis, vol.195, pp.107957, 2024.
[21] Meng, H., Hu, M., Kong, Z., Niu, Y., Liang, J., Nie, Z., & Xing, J. "Risk analysis of lithium-ion battery accidents based on physics-informed data-driven Bayesian networks,". Reliability Engineering & System Safety, vol.251, pp.110294, 2024.
[22] Mohanty, A., Mohanty, S., Mohanty, P. P., Soudagar, M. E. M., Ramesh, S., Bhutto, J. K., Barnawi, A. B., & Cuce, E. "Enhanced stability and optimization of SMES-based deregulated power systems using the repulsive firefly algorithm," Physica C: Superconductivity and its Applications, vol. 632, pp. 1354692, 2025.
[23] Sharma, N., & Gupta, V. "A comprehensive study of fractal clustering and firefly algorithm for WSN Deployment: Implementation and outcomes," MethodsX, vol. 13, pp. 103030, 2024.
[24] Yousif, A. "An Adaptive Firefly Algorithm for Dependent Task Scheduling in IoT-Fog Computing," CMES - Computer Modeling in Engineering and Sciences, vol. 142, no. 3, pp. 2869–2892, 2025.
[25] Awan, U., Hannola, L., Tandon, A., Goyal, R. K., & Dhir, A. "Quantum computing challenges in the software industry. A fuzzy AHP-based approach,". Information and Software Technology, vol.147, pp.106896, 2022.
[26] Fu, X., Lao, L., Bertels, K., & Almudever, C. G. "A control microarchitecture for fault-tolerant quantum computing,". Microprocessors and Microsystems, vol.70, pp.21–30, 2019.
[27] Soeparno, H., & Perbangsa, A. S. "Cloud Quantum Computing Concept and Development: A Systematic Literature Review,". Procedia Computer Science, vol.179, pp.944–954, 2021.