applied sciences

Article

Research on Application of Improved Quantum Optimization Algorithm in Path Planning

Zuoqiang Du and Hui Li *

School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China; bendian2006@163.com
* Correspondence: hrbcu_lh@163.com

Abstract: For the building emergency evacuation path planning problem, existing algorithms suffer from low convergence efficiency and the problem of getting trapped in local optima. The Bloch Spherical Quantum Genetic Algorithm (BQGA) based on the least-squares principle for single-robot path planning and Bloch Spherical Quantum Bee Colony Algorithm (QABC) for multi-robots path planning are studied. Firstly, the characteristics of three-dimensional path planning are analyzed, and a linear decreasing inertia weighting approach is used to balance the global search ability of chromosomes and accelerate the search performance of the algorithm. Then, the application algorithm can generate a clear motion trajectory in the raster map. Thirdly, the least squares approach is used to fit the results, thus obtaining a progressive path. Finally, multi-robots path planning approaches based on QABC are discussed, respectively. The experimental results show that BQGA and QABC do not need to have a priori knowledge of the map, and they have strong reliability and practicality and can effectively avoid local optimum. In terms of convergence speed, BQGA improved by 3.39% and 2.41%, respectively, while QABC improved by 13.31% and 17.87%, respectively. They are more effective in sparse paths.

Keywords: Bloch sphere; path planning; Quantum Optimization Algorithm; quantum genetic algorithm; quantum bee colony algorithm

check for updates

Citation: Du, Z.; Li, H. Research on Application of Improved Quantum Optimization Algorithm in Path Planning. *Appl. Sci.* **2024**, *14*, 4613. https://doi.org/10.3390/app14114613

Academic Editor: Andrea Prati

Received: 30 March 2024
Revised: 17 May 2024
Accepted: 23 May 2024
Published: 27 May 2024

1. Introduction

In recent years, path planning has been a hot problem in the research field, such as in robot path planning and evacuation path planning [1,2]. Commonly used path planning algorithms include the ant colony algorithm [1], artificial potential field algorithm [3], genetic algorithm [4], and A* algorithm [5]. The core requirement of path planning is to plan the optimal and safest path from the starting point to the target one according to the optimization characteristics, such as shortest distance, shortest time, and lowest energy consumption [6–8]. With the development of emerging technologies, methods based on deep learning and quantum optimization algorithms have also been applied to the field of path planning [9–12].

Sabzekar et al. employed a deep deterministic policy gradient algorithm to model continuous environments, concurrently proposing an innovative inner product reward function, thereby enhancing the objective tracking and obstacle avoidance capabilities of unmanned aerial vehicles [9]. Liu et al. delved into the study of deep Q-learning models, greedy table-based Q-learning models, and online optimization frameworks for addressing the shortest path problem between two points and the issue of random obstacles [10]. The Quantum Optimization Algorithm (QOA) is a brand new evolutionary algorithm proposed by Ajit Narayanan and Mark Moore, who combined quantum computing theory into the classical genetic algorithm in 1996 [13]. Built upon quantum principles, the quantum optimization algorithm utilizes quantum bit encoding and updates populations through quantum gates to pursue the global optimal solution. Contrasting with traditional evolutionary algorithms, it boasts features like smaller population size, faster computation speed, and robust global optimality detection. The Quantum Genetic Algorithm (QGA) amalgamates the merits of quantum computing and genetic algorithms, finding applications in various domains such as function optimization, combinatorial optimization, image processing, data mining, and production scheduling. Consequently, it offers significant advantages and exhibits robust vitality, holding substantial theoretical value and promising application prospects for QOA.

Lewandowski et al. utilized Bloch spherical descriptions to model systems during the design of quantum reversible logic. Subsequently, Bloch spherical coordinates have frequently been employed in quantum logic operations and algorithms [14,15]. A QGA for asymptotic Bloch spherical search was proposed by Sheng Li, and applied to the extreme value optimization problem of multivariate functions [16]. Compared with the traditional optimization algorithm for solving complex nonlinear problems, the quantum genetic algorithm is generated under the guiding idea of de-coarsening, inheriting the strengths of the genetic algorithm while also mitigating some of the drawbacks, such as oversized populations and long convergence time, to some extent [17]. It achieves better optimization results than traditional algorithms.

With the rapid development of the social economy, building designs are becoming more complex and taller, resulting in more intricate internal passage distributions. However, most current algorithm research focuses on solving problems in two-dimensional building environments, which presents numerous limitations when dealing with the relatively complex three-dimensional space of modern buildings [18,19]. Hamacher et al. characterized the indoor path planning problem as a dynamic net-work flow problem, applying dictionary optimization algorithms to minimize evacuation time [20]. Pursals improved evacuation and inverse evacuation functions, proposing a novel evacuation formula to address evacuation problems [21]. Lujak, building upon shortest path indoor path planning, considering the unpredictability of hazardous conditions, introduced evacuation betweenness centrality and evacuation centrality, thereby enhancing the safety of evacuation routes [22]. Hong et al. addressed path planning problems in 3D scene-constrained spaces, presenting a three-stage heuristic approach for constructing route networks based on minimum weighted set covering to enhance the utilization efficiency of escape exits [23].

The path planning problem for the interior of a building is a typical non-deterministic polynomial problem. That is, it is the problem of shortest path usage time formed by a fixed number of branch nodes with constant start and end nodes [24]. In the whole path planning process, it is important to control the direction and distance of the path [25]. In the Bloch Sphere Quantum Genetic Algorithm (BQGA), the entire Bloch sphere is the search range for the optimal solution, especially since there is no clear direction for the search, so the blindness and uncertainty of the search greatly reduce the search efficiency and optimality. At the same time, the size of $\Delta\theta$ and $\Delta\varphi$ in this algorithm usually needs to be determined according to the specific problem. If they take too small a value, they will make the optimization process slow and thus reduce efficiency. If they take too large a value, they will make the algorithm cross the global optimal solution or fall into premature convergence [26].

This paper focuses on researching evacuation path planning for large-scale buildings. To address the issues of local optima and low efficiency in path planning inside buildings, a Bloch spherical coordinate algorithm incorporating least squares theory and the Bloch Spherical Quantum Bee Colony Algorithm (QABC) are employed. Firstly, the characteristics of three-dimensional path planning are analyzed, and a linear decreasing inertia weighting approach is used to balance the global search ability of chromosomes and accelerate the search performance of the algorithm. Then, the application algorithm can generate a clear motion trajectory in the raster map. Thirdly, the least squares is used to fit the results, thus obtaining a progressive path. Finally, multi-robots path planning approaches based on QABC are discussed, respectively. The performance is verified by comparing with Ant Colony Algorithm (ACA), Genetic Algorithm (GA) and Artificial Bee Colony Algorithm (ABA).

2. Bloch Sphere Quantum Genetic Algorithm

The quantum genetic algorithm is a probabilistic optimization algorithm based on quantum computing principles, aiming to search for the optimal solution through the evolution and mutation operations of chromosome genes. Its advantages include a small population size, fast convergence, and robust global search capabilities. However, the quantum genetic algorithm requires measuring the state of quantum bits to obtain binary solutions, involving highly random and blind probabilistic operations. Comparing with the traditional QGA, an improved one uses Bloch spherical coordinates for encoding. It extends the search on each dimension to a three-dimensional search in space, with each chromosome occupying three positions in space, to increase the randomness and diversity of quantum chromosomes, and it constructs a new update strategy for quantum chromosomes by incorporating the coordinate transformation of the least square principle. Meanwhile, a selection formula for the variation angle is constructed to improve the search capability.

2.1. Quantum Chromosome Model

In quantum computing, the smallest unit of information is the quantum bit [27,28]. On the three-dimensional Bloch sphere, a qubit can be represented as (1).

$$|\varphi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, \tag{1}$$

where $\cos\frac{\theta}{2}$ and $e^{i\varphi}\sin\frac{\theta}{2}$ are complex numbers, which are called the probability amplitudes of the corresponding states of a qubit, and $\left|\cos\frac{\theta}{2}\right|^2$ and $\left|e^{i\varphi}\sin\frac{\theta}{2}\right|^2$ represent the probability that the qubit is in $|0\rangle$ or $|1\rangle$, respectively [29,30]. The normalization condition is satisfied as (2).

$$\left|\cos\cos\frac{\theta}{2}\right|^2 + \left|e^{i\varphi}\sin\sin\frac{\theta}{2}\right|^2 = 1, \tag{2}$$

On the Bloch sphere, a point p can be determined by two angles θ and φ, as shown in Figure 1.

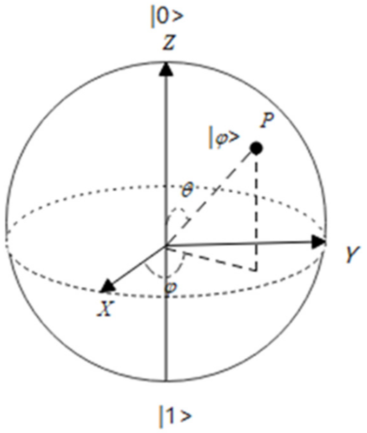


Figure 1. Bloch spherical representation of quantum bits.

On the three-dimensional Bloch sphere, any qubit corresponds to a point on the sphere, after which the qubit can be expressed in Bloch spherical coordinates as:

$$|\varphi\rangle = [\cos\varphi\sin\theta \ \ \sin\varphi\sin\theta \ \ \cos\theta]^{\mathrm{T}}, \tag{3}$$

In the BQGA process, the Bloch spherical coordinate encoding of the qubit is directly adopted. Considering the three coordinates of the qubit as three juxtaposed genes, each chromosome contains three juxtaposed gene chains, and each gene chain represents an optimal solution [31]. Therefore, let P_i be the i chromosome in the population, and the BQGA encoding method is shown as (4).

$$P_i = \begin{vmatrix} \cos\varphi_{i1}\sin\theta_{i1} \\ \sin\varphi_{i1}\sin\theta_{i1} \\ \cos\theta_{i1} \end{vmatrix} \cdots \begin{vmatrix} \cos\varphi_{in}\sin\theta_{in} \\ \sin\varphi_{in}\sin\theta_{in} \\ \cos\theta_{in} \end{vmatrix}, \tag{4}$$

where $0 \le \theta \le \pi, 0 \le \varphi \le 2\pi, i = 1, 2, \cdots, m.j = 1, 2, \cdots n.$

2.2. Solution Space Transformation and Coordinate Fitting

Then, apply solution space transformation and coordinate fitting to the chromosomes encoded with Bloch coordinates. Each chromosome within the population comprises 3n Bloch coordinates, representing n quantum bits [32]. Through linear transformation, these 3n coordinates can be mapped onto the solution space of the optimization problem, with each coordinate corresponding to an optimization variable within the solution space [33,34].

Let the corresponding solution space transformation formula of the j quantum bit on the i chromosome p_i be

$$\begin{cases} X^j_{ix} = \frac{1}{2}\left[b_j(1+x_{ij}) + a_j(1-x_{ij})\right] \\ X^j_{iy} = \frac{1}{2}\left[b_j(1+y_{ij}) + a_j(1-y_{ij})\right], \\ X^j_{iz} = \frac{1}{2}\left[b_j(1+z_{ij}) + a_j(1-z_{ij})\right] \end{cases} \tag{5}$$

where $i = 1, 2, \cdots, m$ and $j = 1, 2, \cdots n$. m is the population size and n is the number of qubits.

Least squares space fitting is introduced to seek the best function, which is matched for the data. The least squares criterion is a tool that has been widely used in disciplines such as uncertainty prediction [35]. Least-squares fitting is used in fitting linear equations to points in three-dimensional space, and can provide insight into the trend of variation between points, predict future changes in points, and ultimately enable the optimal solution to be determined.

Assuming that there are n data points that need to be fitted, $(x_1, y_1, z_1), \cdots, (x_i, y_i, z_i)$. It can be obtained from the two plane equations as shown in (6).

$$Ax + By + Cz + D = 0(C \neq 0), \tag{6}$$

Then it can be deduced that

$$z = -\frac{A}{C}x - \frac{B}{C}y - \frac{D}{C}, \tag{7}$$

marked as $a = -\frac{A}{C}, b = -\frac{B}{C}, c = -\frac{D}{C}$, and $z = ax + by + c$.

The standard equation is

$$\frac{x - x_0}{X} = \frac{y - y_0}{Y} = \frac{z - z_0}{Z}, \tag{8}$$

So the spatial straight line can be obtained from the intersection of the two planes above, and it van be fitted to fit the equations of these two planes. The least-squares principle is as follows.

$$M = \sum_{i-1}^{n}(ax_i + by_i + c - z)^2, \tag{9}$$

The partial derivative of (9) can be obtained, that is, the model parameters obtained by the least squares fitting.

$$\begin{cases} \sum 2(ax_i + by_i + c - z)x_i = 0 \\ \sum 2(ax_i + by_i + c - z)y_i = 0, \\ \sum 2(ax_i + by_i + c - z) = 0 \end{cases} \tag{10}$$

Formula (10) can be derived.

$$\begin{cases} a\sum {x_i}^2 + b\sum x_i y_i + c\sum x_i = \sum x_i z_i \\ a\sum x_i y_i + b\sum {y_i}^2 + c\sum y_i = \sum y_i z_i\,, \\ a\sum x_i + b\sum y_i + n = \sum z_i \end{cases} \tag{11}$$

$$\begin{bmatrix} \sum {x_i}^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum {y_i}^2 & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix}, \tag{12}$$

Solutions include

$$\begin{cases} a = \frac{n\sum x_i z_i - \sum x_i \sum z_i}{n\sum {x_i}^2 - \sum x_i \sum x_i} - b\frac{n\sum x_i y_i - \sum y_i \sum x_i}{n\sum {x_i}^2 - \sum x_i \sum x_i} \\ b = \frac{(n\sum y_i z_i - \sum y_i \sum z_i)\left(n\sum {x_i}^2 - \sum x_i \sum x_i\right) - (n\sum x_i z_i - \sum x_i \sum z_i)(n\sum x_i y_i - \sum x_i \sum y_i)}{(n\sum {y_i}^2 - \sum y_i \sum y_i)(n\sum {x_i}^2 - \sum x_i \sum x_i) - (n\sum x_i y_i - \sum x_i \sum y_i)(n\sum x_i y_i - \sum x_i \sum y_i)}, \\ c = \frac{\sum z_i - a\sum x_i - b\sum y_i}{n} \end{cases} \tag{13}$$

We can substitute the obtained result into (7) to get:

$$\begin{aligned} z = & \left(\frac{n\sum x_i z_i - \sum x_i \sum z_i}{n\sum {x_i}^2 - \sum x_i \sum x_i} - b\frac{n\sum x_i y_i - \sum y_i \sum x_i}{n\sum {x_i}^2 - \sum x_i \sum x_i}\right)x \\ & + \frac{(n\sum y_i z_i - \sum y_i \sum z_i)\left(n\sum {x_i}^2 - \sum x_i \sum x_i\right) - (n\sum x_i z_i - \sum x_i \sum z_i)(n\sum x_i y_i - \sum x_i \sum y_i)}{(n\sum {y_i}^2 - \sum y_i \sum y_i)(n\sum {x_i}^2 - \sum x_i \sum x_i) - (n\sum x_i y_i - \sum x_i \sum y_i)(n\sum x_i y_i - \sum x_i \sum y_i)}y\ , \\ & + \frac{\sum z_i - a\sum x_i - b\sum y_i}{n} \end{aligned} \tag{14}$$

Finally, the variation formula of the solution space coordinate fitting that integrates the least squares principle can be obtained.

2.3. Updating and Vaviation of Quantum Chromosome

Thirdly, perform update and variation operations on the quantum chromosomes. In BQGA, the update of quantum chromosomes is generally realized by changing the quantum phase through a revolving gate [36]. The purpose of qubit phase rotation is to use each chromosome in the current population to approximate the contemporary optimal chromosome. In the process of approximation, it is possible to produce better contemporary optimal chromosomes, such that the population continues to evolve. The form of updating will be:

$$U = \begin{bmatrix} \cos\Delta\varphi\cos\Delta\theta & -\sin\Delta\varphi\cos\Delta\theta & \sin\Delta\theta\cos(\varphi + \Delta\varphi) \\ \sin\Delta\varphi\cos\Delta\theta & \cos\Delta\varphi\cos\Delta\theta & \sin\Delta\theta\sin(\varphi + \Delta\varphi) \\ -\sin\Delta\theta & -\tan\left(\frac{\varphi}{2}\right)\sin\Delta\theta & \cos\Delta\theta \end{bmatrix}, \tag{15}$$

The corresponding rules are proposed for the direction of the turning angle, which can be judged by the algebraic operation of the three-qubit coordinates of the current chromosome and the optimal chromosome. Based on the optimization theory and the changing trend of the objective function, this paper proposes a progressive Bloch spherical search method, which makes the search process more convenient, and avoids the blindness of the search.

As the number of iterations increases, $\Delta\theta$ and $\Delta\varphi$ should gradually become stable. Therefore, $\Delta\theta$ and $\Delta\varphi$ can be calculated asymptotically on the Bloch sphere with the least-squares theory. The sizes of the two corners of the quantum revolving doors can be derived by fitting the calculation to prevent the algorithm from losing the global optimal solution or prematurely converging.

Assuming that the algorithm evolves to the t generation, the updated formulas of $\Delta\theta_{i+1}$ and $\Delta\varphi_{i+1}$ would be:

$$\Delta\theta_{i+1} = (i+1)\frac{\sum_{i=1}^{t} i\Delta\theta_i - \sum_{i=1}^{t}\Delta\theta_i}{\sum_{i=1}^{t} i^2 - \sum_{i=1}^{t} i}, \tag{16}$$

$$\Delta\varphi_{i+1} = (i+1)\frac{\sum_{i=1}^{t} i\Delta\varphi_i - \sum_{i=1}^{t}\Delta\varphi_i}{\sum_{i=1}^{t} i^2 - \sum_{i=1}^{t} i}, \tag{17}$$

where $i = 0, 1, 2, \cdots, t$.

Since the purpose of mutation operation is to achieve population diversity and break through premature convergence, in BQGA, a quantum revolving gate is used to operate θ and φ.

$$\theta = \frac{1}{t}\sum_{i=1}^{t}\theta_i, \tag{18}$$

$$\varphi = \frac{1}{t}\sum_{i=1}^{t}\varphi_i, \tag{19}$$

The quantum mutation operation seeks to change the state of the superposition of the qubit states, so that the original tendency to collapse to the state "1/0" becomes the tendency to collapse to the state "0/1". At the same time, the variation of quantum chromosomes helps to increase the diversity of the population and helps to avoid premature convergence.

2.4. Selection of Fitness Function

The selection of the fitness function directly affects the convergence speed of GA and whether it can find the optimal solution. This is because the genetic algorithm does not use external information in the evolutionary search, and it searches with the adaptation of each individual of the population degree only based on the fitness function [37–39]. The complexity of the fitness function is the main component of the complexity of GA, and the design of the fitness function should be as simple as possible to minimize the time complexity of the calculation. By establishing a mapping relationship between the objective function of the optimization problem and the fitness of the individual, the optimization of the objective function of the optimization problem can be realized in the process of group evolution.

There are three gene chains for each chromosome, which correspond to three objective function values. The chromosome should be represented by the best gene chain. Considering the size of the obstacle region and the distance between the starting point and the target one, an adaptive individual initialization mechanism is proposed. By calculating the Euclidean distance between the starting point and the target one, the fitness function can be derived as

$$Fit(i) = sqrt(B1(1) - B2(1))^2 + (B1(2) - B2(2))^2 + (B1(3) - B2(3))^2, \tag{20}$$

The chromosome with the smallest fitness value in the population is selected as the contemporary optimal chromosome, and the one with the smallest objective function value among its three gene chains is the contemporary optimal chain. B_i $(i = 1,\ 2,\ 3)$ represents the position of each point, and the shortest distance or the fitness function is obtained according to the shortest distance formula of the three-dimensional graph.

The quality of the chromosomes in the iterative process needs to be measured in order to drive the chromosomes to keep approaching the best position in the whole region. The main consideration of the metric fitness in three-dimension path planning in intelligent construction is the shortest path. To make the planned paths shorter in time, a three-sample method is used for interpolation, and the path length of the scatter is obtained by calculating the three samples after differencing the interpolated values. The adaptation value formula is shown as (21).

$$Fitness = \sum_{k=1}^{K} sqrt(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2 + (z_{k+1} - z_k)^2, \tag{21}$$

where K represents the number of interpolations, and k represents the order of interpolation. The total path length is obtained by summing the distances of all path segments. The smaller the value is, the shorter the path is.

The inertial weight ω is an important parameter for the speed of chromosome memory predecessors. In the early stage of the algorithm, for a larger search range, larger inertia weights are needed to improve the performance of a large search range, while in the later stage a more refined search is needed. Therefore, a linear decreasing strategy is used to adjust the weights, and the inertia weights will vary linearly between the maximum and minimum values to balance the search capability. Also, to balance the range difference between the inertia weight values and the chromosome velocity, the path range coefficient A which prevents the divergence of the weight values, is introduced as follow.

$$\omega(x) = \mathrm{A} * \omega_{\max} * (\omega_{\max} - \omega_{\min})(maxgen - x)/maxgen, \tag{22}$$

where $\omega_{\max}$ is the maximum inertia weight, $\omega_{\min}$ is the minimum inertia weight and *maxgen* is the maximum number of iterations.

2.5. Algorithm Steps

In the ideal algorithm, the object is considered as a point without dimensions. The building as a whole is simplified by the grid method [40,41] to obtain a grid of 21 × 21 of equal size, and each grid is assigned a value, and the size of the value in the grid represents the height. The following are the steps of BQGA.

Step 1—Population initialization, setting the population size *popsize*, the number of iterations *maxgen*, the corner step *shiftstep*, the variation probability p_m, the maximum inertia weight $\omega_{\max}$ and the minimum inertia weight $\omega_{\min}$;

Step 2—Three-stranded gene coding, encoding the chromosomal genes by expressions (1), (2), (3) and (4);

Step 3—Spatial transformations solving. Combine the specific optimization problem with expression (5) to perform the solution space transformation and coordinate mapping to the solution space of the optimization problem. Then the obtained solution space coordinates are fit with (14);

Step 4—Chromosome updating. Update the chromosomes by using the quantum rotation gate in expression (15) to obtain new populations;

Step 5—Chromosomal variation. Correction of the direction of chromosome updating by (18) and (19);

Step 6—Adaptation function for screening. The obtained chromosomes are used to calculate the fitness of individual chromosomes based on Equations (20) and (21), and the optimal chromosomes and optimal gene chains are also used as the historical optimal chromosomes and gene chains. If $Fit(i) < Fit(i-1)$, the contemporary optimal solution is updated. Otherwise, the previous solution is kept unchanged;

Step 7—Iterate. Retain the optimal solution of the algorithm. Determine whether the number of iterations reaches the set value. If yes, the iteration ends and stops, otherwise, go back to Step 3;

Step 8: Algorithm termination. When the iteration reaches the maximum number of evolutionary generations, the algorithm stops iterating.

3. Bloch Sphere Quantum Bee Colony Algorithm

3.1. Bloch Coordinate Coding of Nectar Source

Firstly, encode the nectar sources using Bloch coordinate encoding. The coding approach is similar to the quantum genetic algorithm. The coordinates of nectar source are determined by defining a point p for each pair of parameters θ and φ, and any qubit corresponds to a point on the spherical surface. Finally, P_i which is the ith food source in the population, is obtained, and three feasible solutions in the corresponding space are calculated.

The original nectar source is coded with Bloch coordinates of qubits directly, and the ith individual R_i in the population is encoded as follows:

$$R_i = \begin{vmatrix} \cos\varphi_{i1}\sin\theta_{i1} \\ \sin\varphi_{i1}\sin\theta_{i1} \\ \cos\theta_{i1} \end{vmatrix} \cdots \begin{vmatrix} \cos\varphi_{ij}\sin\theta_{ij} \\ \sin\varphi_{ij}\sin\theta_{ij} \\ \cos\theta_{ij} \end{vmatrix} \cdots \begin{vmatrix} \cos\varphi_{in}\sin\theta_{in} \\ \sin\varphi_{in}\sin\theta_{in} \\ \cos\theta_{in} \end{vmatrix}, \tag{23}$$

$$\begin{cases} \varphi_{ij} = 2\pi \times rand(0,1) \\ \theta_{ij} = 2\pi \times rand(0,1) \end{cases}, \tag{24}$$

where $1 \le i \le m$, $1 \le j \le n$, m is the population size, and n is the spatial dimension.

Each food source R_i consists of three chains, each corresponding to a feasible solution x_{ix}, x_{iy} or x_{iz} in the original solution space.

$$\begin{cases} x_{ix} = (x_{i1}^{(1)}, x_{i2}^{(1)}, \ldots, x_{in}^{(1)}) \\ x_{iy} = (x_{i1}^{(2)}, x_{i2}^{(2)}, \ldots, x_{in}^{(2)}) \\ x_{iz} = (x_{i1}^{(3)}, x_{i2}^{(3)}, \ldots, x_{in}^{(3)}) \end{cases}, \tag{25}$$

3.2. Solution Space Transformation and Food Sources Updating

Apply solution space transformation to the chromosomes encoded with Bloch coordinates, and perform an update operation on the quantum chromosomes. The search space for each dimension of the QABC algorithm is [−1,1]. In order to calculate the fit value of the artificial bee, the solution space needs to be transformed, and the corresponding transformation of the jth qubit of the ith food source R_i in the Bloch spherical coordinate system is

$$\begin{cases} X_{ij}^{(1)} = \frac{1}{2}\left[b_j\left(1+\cos\varphi_{ij}\sin\theta_{ij}\right) + a_j\left(1-\cos\varphi_{ij}\sin\theta_{ij}\right)\right] \\ X_{ij}^{(2)} = \frac{1}{2}\left[b_j\left(1+\sin\varphi_{ij}\sin\theta_{ij}\right) + a_j\left(1-\sin\varphi_{ij}\sin\theta_{ij}\right)\right] \\ X_{ij}^{(3)} = \frac{1}{2}\left[b_j\left(1+\cos\theta_{ij}\right) + a_j\left(1-\cos\theta_{ij}\right)\right] \end{cases}, \tag{26}$$

where $i = 1, 2, \cdots, m$ and $j = 1, 2, \cdots n$.

In QABC, updating quantum chromosomes is usually finished by changing the quantum phase with revolving gates. Updating the location of the honey source essentially involves changing the angles between each Bloch coordinate.

The honey sources have the capacity for evolution, that is, they gradually move towards the optimal solution at the same time. Consider adjusting for the change in φ by the cosine of the difference between θ and θ_{gbest} and for the change in θ by the cosine of the difference between φ and θ_{gbest}.

$$\begin{cases} \varphi_{ij} = \varphi_{ij} + \cos\left(\theta_{ij} - \theta_{gbest}\right) * \left(\varphi_{ij} - \varphi_{gbest}\right) \\ \theta_{ij} = \theta_{ij} + \cos\left(\varphi_{ij} - \varphi_{gbest}\right) * \left(\theta_{ij} - \theta_{gbest}\right) \end{cases}, \tag{27}$$

where φ_{ij} and θ_{ij} are the Bloch coordinate phase angles of the ith food source at the jth quantum position, and φ_{gbest} and θ_{gbest} are the globally optimal Bloch coordinate phase angles in each generation.

3.3. Multi-Robots Path Planning Based on QABC

A two-level planning method is adopted for multi-robots based on QABC: the first level is for each robot to plan its global trajectory with QABC, and the second level is to solve the trajectory conflict problem between multi-robots and the external circular collision method to detect other robots in the nearby environment.

3.3.1. External Circle Collision Detection Algorithm

In this paper, the priority method is used to deal with the collision strategy. The priorities of robots in the whole system are defined. When a path conflict occurs, the low-priority robot obeys the command of the high-priority robot and coordinates with the action of the high-priority robot to resolve the conflict. The robot operates on the grid using the action of the octagonal structure, and the motion step size is 1 or $\sqrt{2}$. The position of the robot at time t is (x_t, y_t, z_t). Then,

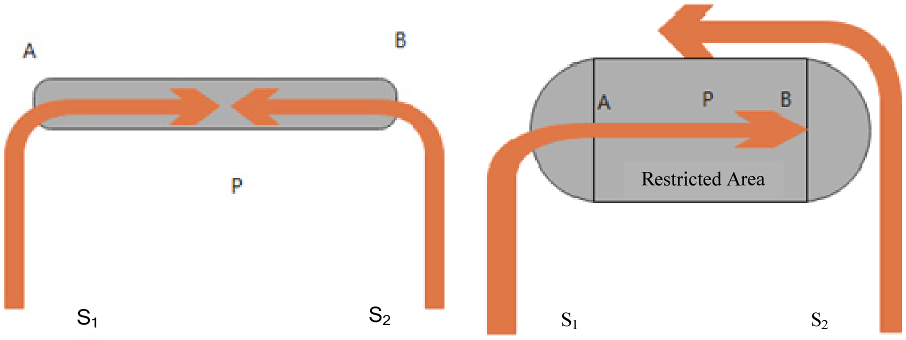
$$\begin{cases} x_t - x_{t-1} = 0 \\ y_t - y_{t-1} = 0 \\ z_t - z_{t-1} = 0 \end{cases} or \begin{cases} x_t - x_{t-1} = 1 \\ y_t - y_{t-1} = 1 \\ z_t - z_{t-1} = 1 \end{cases}, t \in [1, end], \tag{28}$$

3.3.2. Conflict Resolution Algorithm

If the priorities of the robots are the same, then the robot with the shortest expected initial path will avoid the robot with the longest expected initial path for optimal efficiency.

1. Conflict type;

Conflict type I and the collision prediction are shown in Figure 2. Two robots travel in opposite directions on segment AB, and S_2 is moving along BA, while S_1 is moving along AB. S_1 is predicted to collide with S_2 at P point.



(**a**) (**b**)

Figure 2. (**a**) Collision type I; (**b**) collision prevention approach I.

In order to avoid collision, S_2 passes in accordance with the established path, while S_1 defines the safety range around the collision course of the outer ring as the restricted area of S_1, according to the priority, and S_1 waits for S_2 to pass. The shadow area in the figure shows the superposition of the circular part with radius d.

2. Conflict type II

When the paths of two robots engaged in circumferential collision walking intersect, a collision may occur. Suppose that S_1 moves along the line S_1P, and S_2 moves along the line S_2P; the path intersection is P. If S_1 moves along the planned path to point C and S_2 arrives at point D with the same speed, their direction of movement is d. When the distance between two robots DC is less than the safe distance, that is $0 \leq d \leq a$, $0 \leq d \leq b$, they are expected to collide. Conflict type I and the collision prediction are shown in Figure 3. The collision prediction diagram is shown in Figure 4.

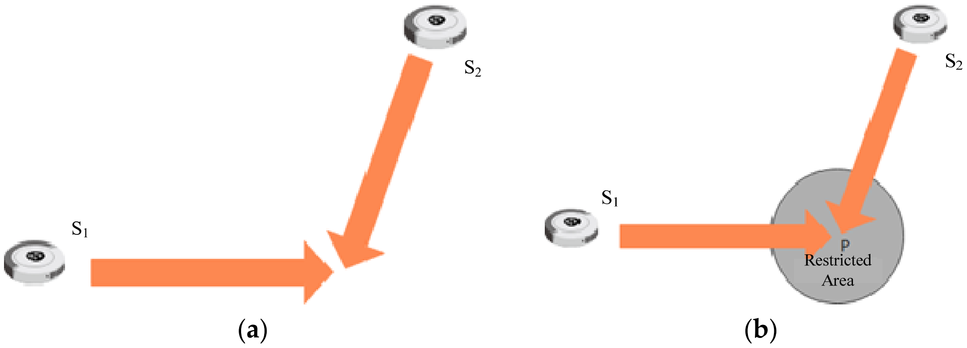


Figure 3. (**a**) Collision type II; (**b**) collision prevention approach II.

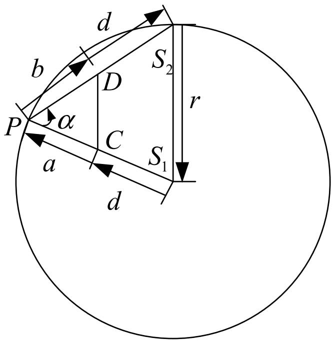


Figure 4. Collision prediction.

To avoid a collision, S_2 with low priority actively avoids the robot S_1 and waits at a safe distance, while S_1 maintains its initial speed until it leaves the outer circle of S_2. Finally, S_2 continues along its original path and speed.

3. Conflict type III

Taking the position of the parked robot as the center of the circle, the circle corresponding to the radius and safety distance is divided into small areas, and the local route is designed according to the improved algorithm. To avoid collision, the robot path needs to change, as shown in Figure 5.

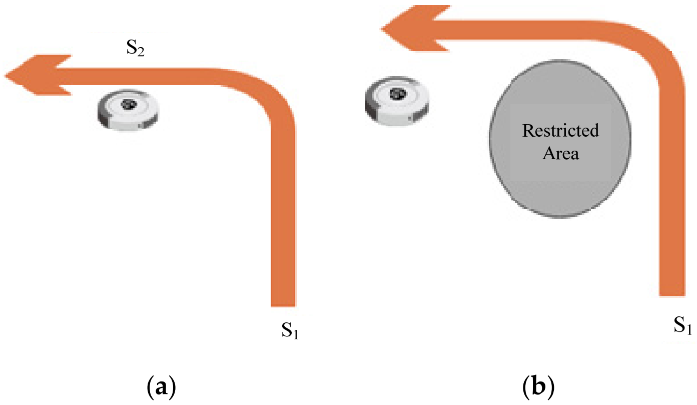


Figure 5. (**a**) Collision type III; (**b**) collision prevention approach III.

4. Analysis of Evacuation Path Planning Model Experiment

4.1. Basic Assumptions

The following assumptions and parameter settings need to be made for the set-up algorithm before establishing the mathematical model of the evacuation path optimization problem:

(a) The initial node is the accident site, the endpoint is the safe area, and all the nodes between them are evacuation points;
(b) Each evacuation point can be passed only once;
(c) When starting the operation, the route must be executed to completion without interruption;
(d) During the execution of the task, the speed change is uniform;
(e) Human factors and obstacle changes are ignored when the accident occurs.

4.2. Raster Data Capture

Commonly used methods for modeling the environment include the grid map method [42], the free space method [43] and the geometric information method [44]. In this paper, the grid method is chosen to establish the three-dimensional path planning space, which is an approach of map modeling to fill in the obstacles in the environment, i.e., environment optimization. The planning space is decomposed into a number of network cells with binary information consisting of a matrix of 0s and 1s. 0 represents a free form, indicating that there are no obstacles and there is a free passage, colored white. 1 represents an obstacle grid, indicating that there are obstacles that need to be moved around, colored black [45]. The workspace is decomposed into cells and then an algorithm is used to search for safe paths in the cells. The method is simple to implement and easy to extend to three-dimensional environments. We use a plane to uniformly divide the three-dimensional space, from which the discrete points required for three-dimensional path planning are extracted. Assuming that each grid represents a sparse point, and that each grid has the same size, the size length set to unit 1.

Firstly, the eight adjacent grids in the square grid are anlyzed from a steering perspective, with the horizontal positive direction as the 0° reference, 45°, 90°, 135° clockwise, 45°, 90°, 135° counterclockwise, and 180° backwards. Next, the OABC-$O_1A_1B_1C_1$ three-dimensional space is constructed with O as the vertex, shown in Figure 6.

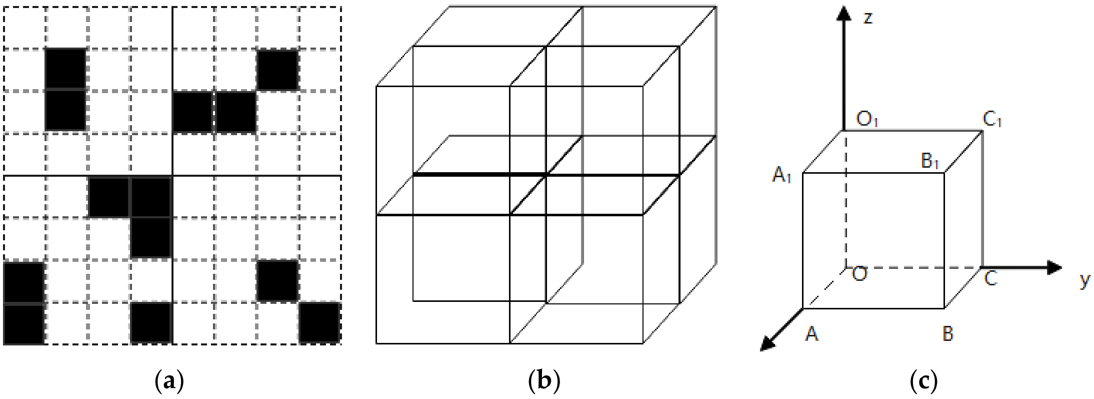


Figure 6. (**a**) Grid plan model; (**b**) three-dimensional space occupation; (**c**) three-dimensional model space.

The three-dimensional space is divided uniformly with a plane, from which the discrete points required for three-dimensional path planning are extracted. The space OABC-$O_1A_1B_1C_1$ is first divided into n equal parts along the OC side to obtain $n + 1$ planes $\Omega_i(i = 1, 2, \cdots n)$, and then the $n + 1$ planes are divided into m equal parts along the OA side and l equal parts along OO′ to obtain each intersection point in the space. The planes are divided as shown in Figure 7.

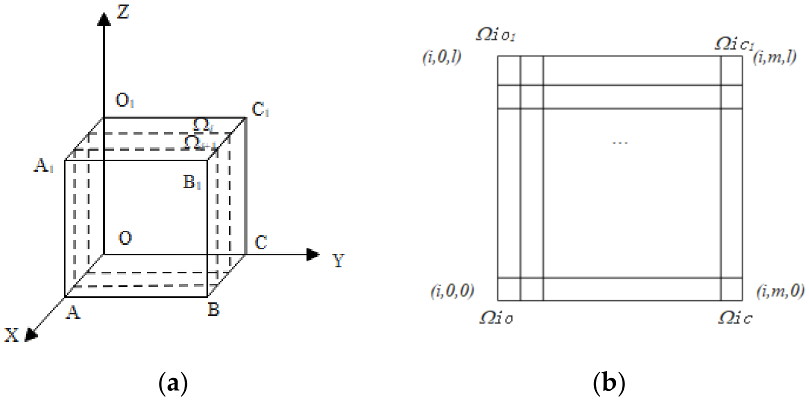


Figure 7. (**a**) Plane segmentation of three-dimensional space; (**b**) two-dimensional display of three-dimensional space division.

Through the above steps, the entire planning space OABC-$O_1A_1B_1C_1$ is decomposed into a collection of spatially discrete points, which form the paths searched by the BQGA.

The schematic diagrams of the simulated obstacle avoidance movement in different raster maps are shown in Figure 8, and the adjacent 8 rasters are used as optional evacuation locations under different real-life conditions. The specific evacuation locations depend on the update direction and the size of the quantum revolving door.

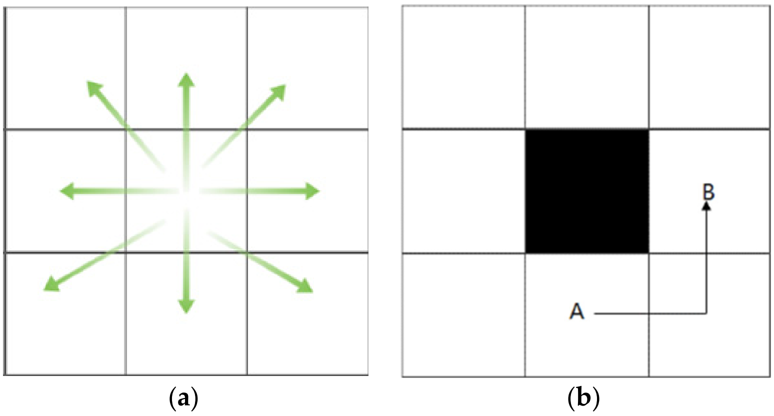


Figure 8. (**a**) Direction of movement; (**b**) demonstration of obstacle avoidance.

The data, such as the length, width and height of the building, are simulated, and the point data obtained are converted into a grid of 21 × 21. Next, each grid to be interpolated is assigned the value of the building heights. Its coordinates are set as (x_i, y_j, z_k) by labeling each grid in the raster layer. The original path planning space is transformed into a set of raster numbers, and one only needs to select this set of data points when path planning is carried out. In the rasterized three-dimensional terrain, only 19 raster points need to be selected from the accident occurrence point to the safety zone to form an evacuation path in turn.

4.3. Experimental Analysis of Single-Robot Path Planning

In order to verify the effectiveness of the Bloch Spherical Quantum Genetic Algorithm, the model is solved by simulation. The simulations of BQGA, ACA and GA were conducted for comparison tests, respectively. In order to reflect the fairness of the comparison results, the same population size and evolutionary generation were used for three algorithms. The parameters were set as follows: evolutionary generation *maxgen* = 100, number of populations *popsize* = 100, number of quantum bits n = 2, and 19 evacuation nodes. Then the performances of the three algorithms are verified in Figures 9–11.

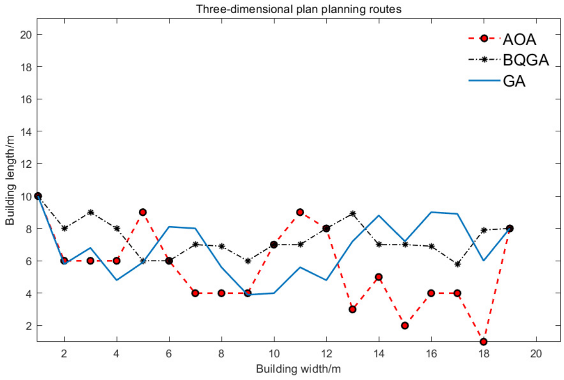


Figure 9. Evacuation path planning route.

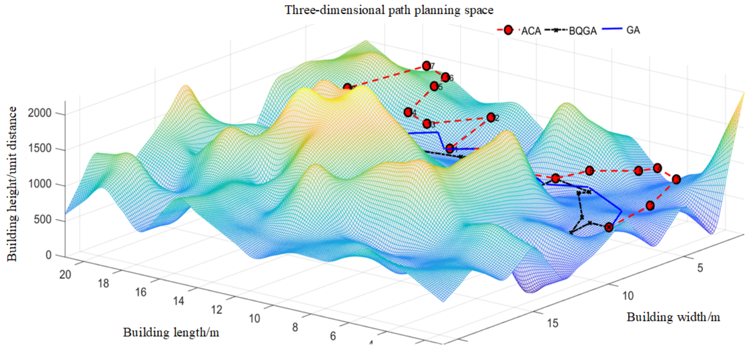
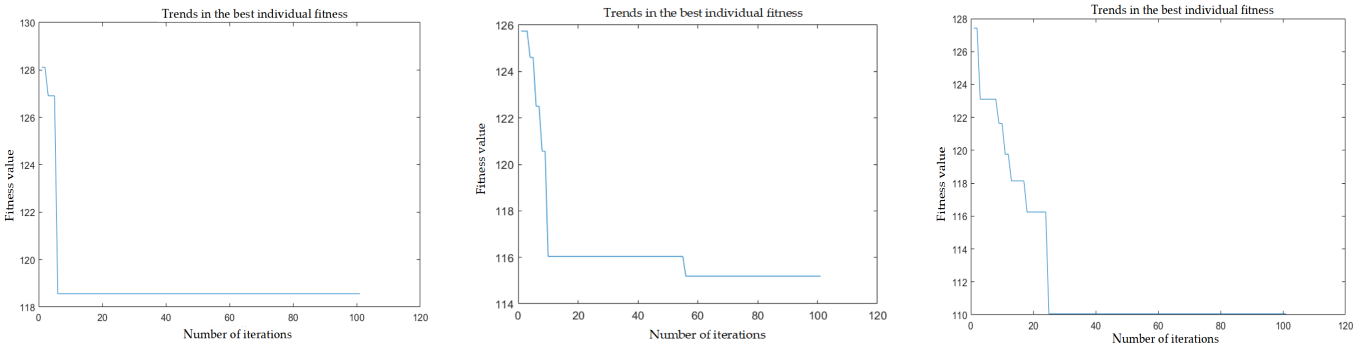


Figure 10. Three-dimensional route planning space.



(**a**) (**b**) (**c**)

Figure 11. (**a**) Trends in the best individual fitness of the BQGA model; (**b**) trends in the best individual fitness of the ACA model; (**c**) trends in the best individual fitness of the GA model.

4.3.1. Results Analysis

The visualization of the spatially optimal route of the evacuation path three-dimensional raster model is given in Figure 11. The best path route based on the ACA search took 1.362073 s to run, the best path route based on the GA search took 1.304934 s to run, and the best path route based on the BQGA search took 1.256061 s.

In Figure 11, the trends of the best individual fitnesses under different algorithms are represented. Figure 11 shows the trend of the best individual fitness for the BQGA, from the initial 128 to 118, with a difference of 10, the trend of the best individual fitness for the ACA, from the initial 126 to 115, with a difference of 11, and the trend of the best individual fitness for the GA, from the initial 128 to 110, with a difference of 18.

For ACA, it is easy to fall into local optimum prematurely, and the computational search time is too long because of its complexity. It is easy to encounter a stagnation phenomenon, that is, after the search is carried out to a certain degree, it cannot search the solution space further, which is not conducive to discovering better solutions. ACA's optimization efficiency is low.

For GA, the encoding of ordinary algorithms does not fully represent the constraints of the optimization problem, and therefore it requires the consideration of thresholds for infeasible solutions, which in turn increases the workload and solution time, making the iteration time larger.

BQGA avoids the random characteristics brought by generating binary codes with measuring quantum bits, and expands the planar unit circle description of quantum bits to Bloch spherical description, so that the quantum behavior can be fully reflected. Then, the frequent binary number decoding process can be avoided, and the time will be reduced. Thirdly, BQGA expands the number of global optimal solutions and improves the probability of obtaining global optimal solutions. Finally, the number of global optimal solutions can be expanded and the probability of obtaining global optimal solutions will be improved.

4.3.2. Path Instance Analysis

The three algorithms were run 30 times, respectively, and the evaluation indicators were recorded. Three groups of different populations with *popsize* = 100, *popsize* = 150, and *popsize* = 200 were set. The comparisons of the optimization results of the best individual fitness functions of the three algorithms are listed in Tables 1–3.

Table 1. Comparison of ACA, GA and BQGA under evacuation conditions of *Maxgen* = 100, *Popsize* = 100.

Algorithm	Best Result	Worst Result	Average Result	Time/s
ACA	126	110	118	1.2914
GA	127	113	120	1.2784
BQGA	128	118	123	1.2476

Table 2. Comparison of ACA, GA and BQGA under evacuation conditions of *Maxgen* = 100, *Popsize* = 150.

Algorithm	Best Result	Worst Result	Average Result	Time/s
ACA	127	110	118.5	1.8490
GA	126	112	119	1.8043
BQGA	128	118	123	1.7658

Table 3. Comparison of ACA, GA and BQGA under evacuation conditions of *Maxgen* = 100, *Popsize* = 200.

Algorithm	Best Result	Worst Result	Average Result	Time/s
ACA	126	112	119	2.3565
GA	125	111	118	2.3367
BQGA	128	118	123	2.3102

For the continuous optimization problem, in BQGA, the two parameters θ and φ of the quantum bit can be adjusted simultaneously by using the current quantum bit to rotate around a certain rotation axis toward the target quantum bit, and the corresponding Bloch coordinates can be directly obtained by the projection measurement of the quantum bit, which achieves the best matching of the amount of adjustment of the two parameters in the quantum bit adjustment. Obviously, this best-matched rotation has higher optimization efficiency. Therefore, the optimization efficiency of BQGA is higher than that of ACA and GA, i.e., the shorter the time, the larger the average result value, which indicates its effectiveness and efficiency. The optimization efficiency of BQGA is improved by approximately 3.39% and 2.41% compared to the other two algorithms, respectively.

4.3.3. Influence Analysis of BQGA and ACA, GA Stability Results

With the expansion of the problem scale, the size of the problem scale has a strong influence on the algorithm's running time, and the overall trend of the running time of the algorithms increases from the analysis of the path instance. So the algorithms should be evaluated according to the size of the actual problem. Therefore, the results are further explored for each of the two algorithms in the case of increasing problem size, based on the fitting, using the control variables method. The number of location points and the running time are roughly linear in BQGA, while in ACA and GA, the number of location points and the running time tend to show a more quadratic correlation. When the number of location points is large, the running time of ACA and GA is obviously longer than that of BQGA. The relationship between the number of iterations and the degree of adaptation can be plotted according to the data obtained from the two algorithms, as shown in Figure 12.

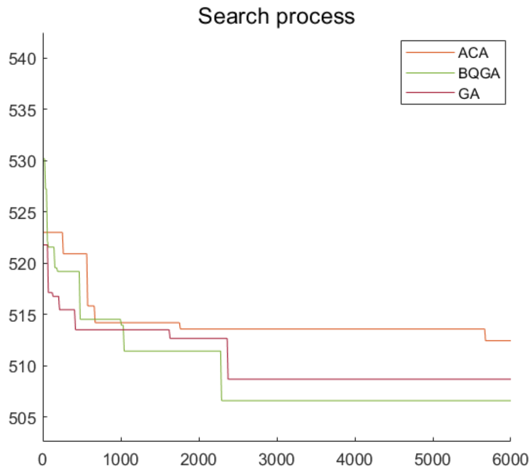


Figure 12. Trends in the best individual fitness of the GA model.

4.4. Experimental Analysis of Multi-Robots Path Planning

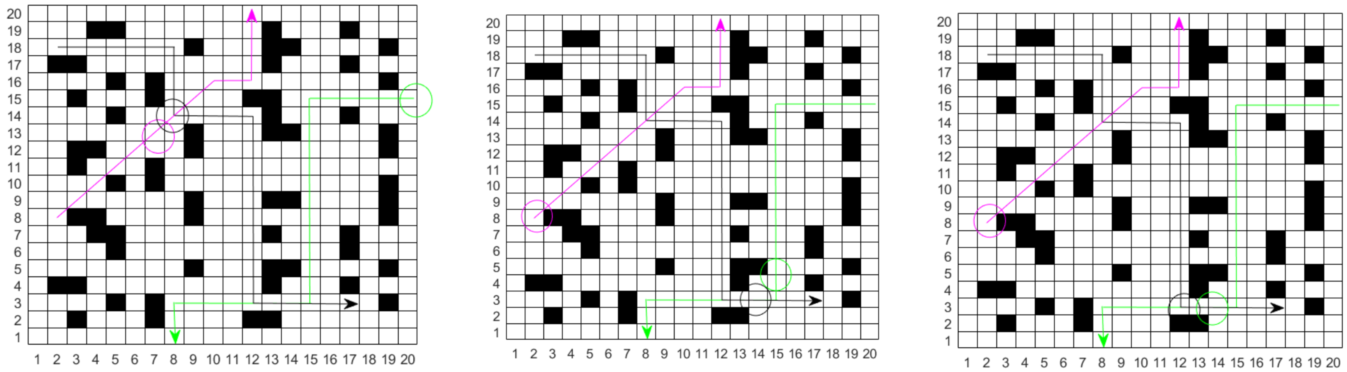
In order to verify the effectiveness of the QABC, the model is solved by simulation. The simulations of QABC, ACA and ABC were conducted for comparison tests, respectively. The parameters were set as follows: population size $N = 200$, limiting frequency $limit = 50$, maximum cycle number $maxcycle = 1000$, number of quantum bits $n = 2$ and 19 evacuation nodes.

4.4.1. Collision Detection

As shown in Figure 13a, each robot moves towards the target point from the initial position, and robots A and C detect each other in their own peripheral circles. In order to prevent collision, the method shown in collision type II is adopted. Meanwhile, the robots with a lower priority will actively avoid the one with a higher one.

As shown in Figure 13b, robots A and B detect the existence of each other by using the external circle method. In this case, collision type III is required for conflict resolution and avoidance, so that the robots can pass smoothly.

As shown in Figure 13c, robots A and B detect each other using external circles, which belongs to collision type I. The corresponding conflict resolution method needs to be used to let the one with higher priority pass first, and the one with lower priority can pass only when the external circle is larger than the security zone.



(**a**) (**b**) (**c**)

Figure 13. (**a**) Conflict type II; (**b**) conflict type III; (**c**) conflict type I.

4.4.2. Results Analysis

The results for the three approaches are shown in Table 4. It can be seen from the data in the table that multiple robots S_1 and S_2, based on the quantized artificial bee colony algorithm, increase the path length and shorten the running time by using priority-based path planning compared with those in S_1 and S_2 in QABC. After the introduction of the circular collision detection operator, the safety and directivity of the second layer's local path planning in the two-layer programming method are enhanced, and the ability to search the optimal solution using QABC is improved. Meanwhile, the application saves time, reduces work cost, and effectively improves the work efficiency of multi-robots. The optimization efficiency of QABC is improved by approximately 13.31% and 17.87% compared to the other two algorithms, respectively.

Table 4. Comparison of ACA, ABA and QABC.

Algorithm	Robot	Estimate Path Length	Actual Path Length	Time/s
ACA	S1	11.983	11.983	1.291
	S2	14.678	14.678	1.339
ABC	S1	11.346	12.237	1.343
	S2	12.327	12.438	1.433
QABC	S1	12.628	12.682	1.153
	S2	15.729	15.720	1.127

4.4.3. Influence Analysis of QABC, ACA and ABC Stability Results

According to the data obtained by the three algorithms, the relationship between iteration times and fitness can be drawn, as shown in Figure 14. The use of QABC reduces the possibility of collision between robots, and improves the safety of robots. When the number of location points is large, the running times of ABC and ACA are obviously longer than that of QABC. Comparing the stability of the three algorithms, it is found that QABC is more stable.

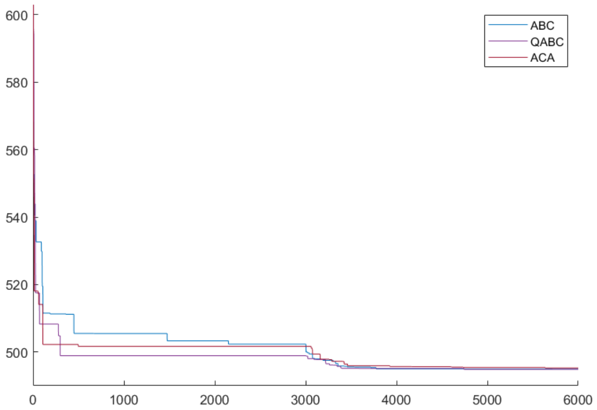


Figure 14. Relationship between the number of iterations and fitness.

5. Conclusions

In this paper, the Bloch Spherical Quantum Genetic Algorithm model fused with the least-squares principle for a single robot and the Bloch Spherical Quantum Bee Colony Algorithm for multi-robots are proposed to seek the optimal path. From a progressive point of view, consider the least-squares method and quantum optimization; the discrete grid division approach deals with the unpredictable evacuation problem caused by emergencies. In the case of uncertain building conditions, a fusion least-squares method is established for studying the three-dimensional path planning problem. The principle of the quantum model and quantum optimization algorithm based on a Bloch sphere is designed. On the basis of verifying the validity of the algorithm, the algorithm comparison is carried out, and the running time and fitness of the algorithm are compared, as well as the influence of stability.

Experiments show that the rasterization of building floors is more convenient for finding the shortest safe path. Compared with ACA, GA and ABC, the methods in this paper can effectively deal with the problems of slow convergence and long execution time caused by the traditional algorithms. The solution to these problems can enable humans to quickly evacuate from dangerous areas when an emergency occurs, thereby improving the safety factor for human activities in the building.

Future research will focus on improved qubit encoding, which can make the food source more evenly traversable throughout the solution space to improve the search efficiency of the algorithm.

Author Contributions: Conceptualization, Z.D. and H.L.; methodology, Z.D.; software, Z.D.; validation, Z.D. and H.L.; formal analysis, Z.D.; investigation, H.L.; resources, Z.D.; data curation, H.L.; writing—original draft preparation, Z.D.; writing—review and editing, H.L.; visualization, H.L.; supervision, Z.D.; project administration, H.L.; funding acquisition, Z.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Science Foundation of Harbin Commerce University, grant number 2023-KYYWF-0983.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Luo, Q.; Wang, H.; Zheng, Y.; He, J. Research on path planning of mobile robot based on improved ant colony algorithm. *Neural Comput. Appl.* **2020**, *32*, 1555–1566. [CrossRef]
2. Xu, L.; Huang, K.; Liu, J.; Li, D.; Chen, Y. Intelligent planning of fire evacuation routes using an improved ant colony optimization algorithm. *J. Build. Eng.* **2022**, *61*, 105208. [CrossRef]

3. Zha, M.; Wang, Z.; Feng, J.; Cao, X. Unmanned vehicle route planning based on improved artificial potential field method. *J. Phys. Conf. Ser.* **2020**, *1453*, 012059. [CrossRef]
4. Xu, X.; Yu, X.Y.; Zhao, Y.; Liu, C.X.; Wu, X. Global path planning of mobile robot based on improved genetic algorithm. *Comput. Integr. Manuf. Syst.* **2022**, *28*, 1659–1672.
5. Li, C.; Huang, X.; Ding, J.; Song, K.; Lu, S. Global path planning based on a bidirectional alternating search A* algorithm for mobile robots. *Comput. Ind. Eng.* **2022**, *168*, 108123. [CrossRef]
6. Gul, F.; Mir, I.; Abualigah, L.; Sumari, P.; Forestiero, A. A consolidated review of path planning and optimization techniques: Technical perspectives and future directions. *Electronics* **2021**, *10*, 2250. [CrossRef]
7. Chen, R.; Hu, J.; Xu, W. An RRT-Dijkstra-Based path planning strategy for autonomous vehicles. *Appl. Sci.* **2022**, *12*, 11982. [CrossRef]
8. Roy, S.; Zhang, Z. Route planning for automatic indoor driving of smart cars. In Proceedings of the 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), Bangkok, Thailand, 16–21 April 2020; pp. 743–750. [CrossRef]
9. Sabzekar, S.; Samadzad, M.; Mehditabrizi, A.; Tak, A. A Deep Reinforcement Learning Approach for UAV Path Planning Incorporating Vehicle Dynamics with Acceleration Control. *Unmanned Syst.* **2023**, 1–22. [CrossRef]
10. Liu, Y.; Vogiatzis, C.; Yoshida, R.; Morman, E. Solving reward-collecting problems with UAVs: A comparison of online optimization and Q-learning. *J. Intell. Robot. Syst.* **2022**, *104*, 35. [CrossRef]
11. Hu, C.; Xia, Y.; Zhang, J. Adaptive Operator Quantum-Behaved Pigeon-Inspired Optimization Algorithm with Application to UAV Path Planning. *Algorithms* **2019**, *12*, 3. [CrossRef]
12. Li, J.; Xu, B.; Yang, Y.; Wu, H. Quantum ant colony optimization algorithm for AGVs path planning based on Bloch coordinates of pheromones. *Nat. Comput.* **2020**, *19*, 673–682. [CrossRef]
13. Narayanan, A.; Moore, M. Quantum-inspired genetic algorithms. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; pp. 61–66. [CrossRef]
14. Yang, S.Y.; Xu, Y.X.; Li, P.C. Quantum-inspired Artificial Fish Swarm Algorithm Based on the Bloch Sphere Search Algorithm. *Inf. Control.* **2014**, *43*, 647–653. [CrossRef]
15. Li, J.C.; Han, K.; Bao, T.Z. An improved Bloch spherical quantum genetic algorithm and its application. *J. Railw. Sci. Eng.* **2016**, *13*, 2262–2269. [CrossRef]
16. Li, S.; Zhang, P.L.; Li, B.; Zhou, Y.C. Quantum Genetic Algorithm for Asymptotic Bloch Spherical Search and Its Application. *Syst. Eng. Theory Pract.* **2016**, *36*, 1042–1046. [CrossRef]
17. Li, J.; Huang, C.; Pan, M. Path planning algorithms for self-driving vehicle based on improved RRT-Connect. *Transp. Saf. Environ.* **2022**, *5*, tdac061. [CrossRef]
18. Kumar, S.; Sikander, A. Optimum mobile robot path planning using improved artificial bee colony algorithm and evolutionary programming. *Arab. J. Sci. Eng.* **2022**, *47*, 3519–3539. [CrossRef]
19. He, X.G.; Liang, J.Z. Genetic algorithms using gradients of object functions. *J. Softw.* **2001**, *12*, 981–986.
20. Hamacher, H.W.; Tufekci, S. On the use of lexicographic min cost flows in evacuation modeling. *Nav. Res. Logist. (NRL)* **1987**, *34*, 487–503. [CrossRef]
21. Pursals, S.C.; Garzón, F.G. Optimal building evacuation time considering evacuation routes. *Eur. J. Oper. Res.* **2009**, *192*, 692–699. [CrossRef]
22. Lujak, M.; Giordani, S. Centrality measures for evacuation: Finding agile evacuation routes. *Future Gener. Comput. Syst.* **2018**, *83*, 401–412. [CrossRef]
23. Hong, Y.; Li, D.; Wu, Q.; Xu, H. Priority-oriented route network planning for evacuation in constrained space scenarios. *J. Optim. Theory Appl.* **2019**, *181*, 279–297. [CrossRef]
24. Luan, P.G.; Thinh, N.T. Hybrid genetic algorithm based smooth global-path planning for a mobile robot. *Mech. Based Des. Struct. Mach.* **2023**, *51*, 1758–1774. [CrossRef]
25. Li, X.; Li, Q.; Zhang, J. Research on global path planning of unmanned vehicles based on improved ant colony algorithm in the complex road environment. *Meas. Control.* **2022**, *55*, 945–959. [CrossRef]
26. Amal, R.S.; Ivan, J.S. A quantum genetic algorithm for optimization problems on the Bloch sphere. *Quantum Inf. Process.* **2022**, *21*, 43. [CrossRef]
27. Nie, Y.; Yu, X. Optimization of deterministic pilot pattern placement based on quantum genetic algorithm for sparse channel estimation in OFDM systems. *IEICE Trans. Commun.* **2020**, *E103.B*, 1164–1171. [CrossRef]
28. Zhang, S.; Du, H.; Borucki, S.; Jin, S.; Hou, T.; Li, Z. Dual resource constrained flexible job shop scheduling based on improved quantum genetic algorithm. *Machines* **2021**, *9*, 108. [CrossRef]
29. Cheng, Z.; Lei, J.; Zhang, Z. Finite element model modification based on improved double-chain quantum genetic algorithm. *J. Wuhan Univ. Technol. (Transp. Sci. Technol.)* **2022**, *46*, 548–551. [CrossRef]
30. Li, Y.; Qin, S.; Jing, L. Research on flight trajectory optimization based on quantum genetic algorithm. *J. Phys. Conf. Ser.* **2020**, *1549*, 022074. [CrossRef]
31. Fan, X.; Wang, J.; Wang, H.; Yang, L.; Xia, C. LQR Trajectory Tracking Control of Unmanned Wheeled Tractor Based on Improved Quantum Genetic Algorithm. *Machines* **2023**, *11*, 62. [CrossRef]
32. Gul, F.; Mir, I.; Alarabiat, D.; Alabool, H.M.; Abualigah, L.; Mir, S. Implementation of bio-inspired hybrid algorithm with mutation operator for robotic path planning. *J. Parallel Distrib. Comput.* **2022**, *169*, 171–184. [CrossRef]

33. Shao, J. Robot path planning method based on genetic algorithm. *J. Phys. Conf. Ser.* **2021**, *1881*, 022046. [CrossRef]
34. Zhang, Z.; Lu, R.; Zhao, M.; Luan, S.; Bu, M. Robot path planning based on genetic algorithm with hybrid initialization method. *IFS* **2022**, *42*, 2041–2056. [CrossRef]
35. Le, D.-K.; Zhu, W.-H.; Zhu, F.; Yao, H.-B.; Geng, Y.; Tang, C.-M.; He, X.; Gui, M.-C.; Zhang, L. Prediction of Human Triglyceride Concentration Based on Quantum Genetic-Partial Least Squares Algorithm and Fluorescence Spectroscopy. *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.* **2021**, *251*, 119461. [CrossRef]
36. Hao, K.; Zhao, J.; Yu, K.; Li, C.; Wang, C. Path planning of mobile robots based on a multi-population migration genetic algorithm. *Sensors* **2020**, *20*, 5873. [CrossRef]
37. Zhao, W.; Han, X.; Li, H.; Li, Z.; Tan, X. Application of Bloch Spherical Quantum Genetic Algorithm in Fire-Fighting Route Planning. In *Business Intelligence and Information Technology: Proceedings of the International Conference on Business Intelligence and Information Technology BIIT 2021, Harbin, China, 16 December 2021*; Springer: Cham, Switzerland; pp. 441–449. [CrossRef]
38. Hou, W.; Xiong, Z.; Wang, C.; Chen, H. Enhanced ant colony algorithm with communication mechanism for mobile robot path planning. *Robot. Auton. Syst.* **2022**, *148*, 103949. [CrossRef]
39. Martinez, F.; Rendon, A. Autonomous motion planning for a differential robot using particle swarm optimization. *IJACSA* **2023**, *14*, 815–821. [CrossRef]
40. Zhu, D.D.; Sun, J.Q. A new algorithm based on Dijkstra for vehicle path planning considering intersection attribute. *IEEE Access* **2021**, *9*, 19761–19775. [CrossRef]
41. Li, H.; Liu, W.; Yang, C.; Wang, W.; Qie, T.; Xiang, C. An optimization-based path planning approach for autonomous vehicles using the DynEFWA-artificial potential field. *IEEE Trans. Intell. Veh.* **2022**, *7*, 263–272. [CrossRef]
42. Tang, G.; Tang, C.; Claramunt, C.; Hu, X.; Zhou, P. Geometric A-Star algorithm: An improved A-Star algorithm for AGV path planning in a port environment. *IEEE Access* **2021**, *9*, 59196–59210. [CrossRef]
43. Zou, A.; Wang, L.; Li, W.; Cai, J.; Wang, H.; Tan, T. Mobile robot path planning using improved mayfly optimization algorithm and dynamic window approach. *J. Supercomput.* **2023**, *79*, 8340–8367. [CrossRef]
44. Han, S.; Wang, L.; Wang, Y.; He, H. A dynamically hybrid path planning for unmanned surface vehicles based on non-uniform Theta* and improved dynamic windows approach. *Ocean. Eng.* **2022**, *257*, 111655. [CrossRef]
45. Tao, Q.Y.; Sang, H.Y.; Guo, H.W.; Wang, P. Improved particle swarm optimization algorithm for AGV path planning. *IEEE Access* **2021**, *9*, 33522–33531. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.