

# A Self-Adaptive Quantum Genetic Algorithm for Network Flow Vehicle Scheduling Problem

Yiming Yan, Aimei Xiao

Department of Computer Science and Technology, Shandong University of Technology, Zibo, China

Email: xam@sdu.edu.cn

**How to cite this paper:** Yan, Y.M. and Xiao, A.M. (2021) A Self-Adaptive Quantum Genetic Algorithm for Network Flow Vehicle Scheduling Problem. *Journal of Computer and Communications*, 9, 43-54.  
<https://doi.org/10.4236/jcc.2021.97005>

**Received:** July 1, 2021

**Accepted:** July 25, 2021

**Published:** July 28, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

Bicycle sharing scheduling is a complex mathematical optimization problem, and it is challenging to design a general algorithm to solve it well due to the uncertainty of its influencing factors. This paper creatively establishes a new mathematical model to determine the appropriate number of vehicles to be placed at each placement point by calculating the traffic weights of the placement points and optimizes the hyperparameters in the algorithm by adaptive quantum genetic algorithm, and at the same time combines the network flow algorithm in graph theory to calculate the most suitable scheduling scheme for shared bicycles by establishing the minimum cost maximum flow network. Through experimental validation, the network flow-based algorithm proposed in this paper allows for a more convenient calculation of the daily bike-sharing scheduling scheme compared to previous algorithms. An adaptive quantum genetic algorithm optimizes the hyperparameters appearing in the algorithm. The experimental results show that the algorithm achieves good results as the transportation cost is only 1/15th of the GA algorithm and 1/9th of the QGA algorithm.

## Keywords

Network Coding, Quantum Genetic Algorithm, Multicast Networks, Optimization

## 1. Introduction

Multicast allows the information transmitted from a single point to multiple points in networks. Compared with the traditional unicast and broadcast methods, multicast dramatically improves transmission efficiency. Multicast technology can provide critical technical support for most services, such as distance learning, video on demand, data distribution in the data center, etc. [1].

Shared mobility has proliferated in recent years due to the rise of the sharing economy and growing environmental, energy and economic concerns. Among the various forms of shared-use cell phones, public bike-sharing systems [1] [2] have become increasingly popular, with data indicating that over 500 bike-sharing programs are operating in at least 49 countries with a total of 1 million bicycles. As the number of bicycles increases, bicycle-sharing systems usually experience an imbalance between “oversupply” and “undersupply”, with the number of rented bicycles consistently higher or lower than the number of returned bicycles within a certain period and recurring in a particular time cycle depending on holidays or weather factors. The “tidal wave phenomenon” is a typical pattern in the design and operation of bicycle-sharing systems. In some remote areas of cities with low land use, the same demand trend, full travel time, and high and frequent fluctuations in demand for vehicles in a short period are the main reasons for forming the tidal phenomenon.

The current approach to the “tidal wave problem” is a bike-sharing rebalancing algorithm [3], which uses additional vehicles to reallocate shared bicycles to minimize the total cost of ownership, improve the flow of public bicycles through efficient and low-cost bicycle dispatching, reduce over-saturation and over-loading, and meet the demand promptly. The bicycle-sharing rebalancing algorithm uses a static bicycle rebalancing algorithm, which is a method to rebalance the supply and demand of bicycles. The bike-sharing rebalancing algorithm uses a static optimization method, a particular commodity pickup and delivery capacity of the vehicle routing problem, by setting a mixed-integer linear programming formula and setting constraints on the formula to solve the vehicle routing problem and transportation problem. The system is typically statically balanced by reallocating the number of vehicles and distributing them according to the demand in each area when the system is shut down or when the usage is low, such as at night. Since the “tidal phenomenon” occurs at multiple times of the day, the long-period static balancing obviously cannot effectively solve the “tidal phenomenon” [4], which increases the uneven distribution of vehicles and idle rate, reduces the effective use of bicycles, and also It also increases the difficulty and timeliness of bicycle deployment between regions. The demand for the redistribution of shared bicycles in each region will be predicted based on the customer demand in the customer network and the time factors such as holidays and working days as references. Such a model lacks stability, and it is challenging to make corresponding processing according to the changes in the environment, and the self-adaptability is relatively poor.

In recent years, many scholars at home and abroad have conducted in-depth researches on bike-sharing scheduling strategies. The demand estimation of bicycle sharing includes the location estimation of popular areas and the frequency estimation of each popular area. Based on the bicycle demand estimation, the bicycle sharing area and the number of bicycles placed in each area are determined [5] [6]. The prediction of shared bicycle parking includes predicting the

location of shared bicycle parking areas and the prediction of the number of bicycles in each parking area. Most of the existing studies on predicting bicycle use and parking frequency are based on historical record data. Alvarez-Valdez *et al.* proposed the IPPI algorithm [7], which predicts the frequency of bicycle use and parking through a non-flush Poisson process, and predicts the ratio of bicycle use and parking based on historical records; Chen *et al.* proposed the HM algorithm, which uses the historical record data with bicycle demand as the predicted values of bicycle use and parking [7]. Chen *et al.* proposed the HM algorithm, which uses the historical records of car demand as the predicted values of car use and parking frequency [8]. In addition, some models consider multiple influencing factors to improve the prediction accuracy of car use and parking frequency; for example, the MSI algorithm proposed by Li *et al.* considers the similarity of weather, temperature, wind speed and time [9], and the likelihood function is the multiplication of these three similarities, but the weights of these factors are not studied; the MSEWfc algorithm proposed by Liu *et al.* uses a hybrid Gaussian algorithm to forecast demand by taking the same weights for different factors affecting demand forecasting [10]. The above studies on bicycle demand and parking prediction can improve the frequency of use and parking prediction accuracy, but they all lack real-time and more accurate demand and parking prediction. By introducing the network flow algorithm, this paper can minimize the overall bicycle dispatching cost on the one hand, and provide the actual dispatching solution conveniently on the other, which can be said to supplement the gap of the existing bicycle dispatching solution.

There is competition when there are too many requests for bike-sharing parking services, and with the number of constraints between vehicles and parking points, the different order of processing requests makes the allocation effect vary, resulting in different total distance costs. More than one optimization objective in the parking point allocation problem requires a multiobjective optimization algorithm [11]. The genetic algorithm belongs to the random search class of algorithms, which is easy to fall into the local optimum in practical applications, so it needs to be improved. The literature [12] proposed an adaptive quantum revolving gate update approach to change population evolution, thus improving the global search capability of the algorithm. The literature [13] proposed a genetic algorithm incorporating a greedy algorithm in storage vehicle scheduling optimization, which makes crossover and variation more flexible to use and effectively improves the scheduling efficiency of the algorithm. The literature [14] added a neural network to the traditional sliding film control method and used Lyapunov functions to derive an adaptive law for the neural network weights, which improved the accuracy of the original method to a certain extent. In the literature [15], a new prototype of an intelligent parking system was proposed, and a genetic algorithm was used to solve the parking lot vehicle scheduling problem. The objective of the literature [16] is to minimize the sum of weighted flow and inventory cost. The above literature has optimized the genetic algorithm

according to their tasks' characteristics, but there is a lack of adaptive adjustment solutions for the single-vehicle scheduling task. In order to solve the model effectively, this paper proposes an adaptive quantum genetic algorithm-based optimization of the network flow single-vehicle scheduling algorithm for the single-vehicle scheduling task.

## 2. Problem Formulation

Although most current bicycles are dockless, each bicycle company still selects several points with high traffic flow as bicycle drop-off points. After a period of operation, the points where the current number of bicycles is stored more can be easily classified through the background data, and we stipulate that such points are called bicycle drop-off points.

To simplify the study, this paper stipulates that the required solution to the bicycle scheduling problem is described as follows: given  $W$  bicycle drop-off points  $W_i$ , the bicycle drop-off company drops shared bicycles to  $W$  bicycle drop-off points, and the initial number of bicycles dropped at each drop-off point is known. The number of existing bikes at each drop-off point  $W_i$  is  $Num[i]$ , the number of bikes passing this drop-off point on the previous day is  $Num\_Pre[i]$ , and the number of bikes passing this drop-off point on the current day is  $Num\_by[i]$ , and the best number of bikes to be dropped at each drop-off point  $W_i$  is  $Best[i]$ , and the scheduling scheme of bikes between drop-off points are given. The format is  $W_i \Rightarrow W_j$ , which are the drop point to be transferred, the drop point to receive the transfer, and the number of transferred vehicles, respectively.

## 3. Algorithm Description

### 3.1. Algorithm Flow

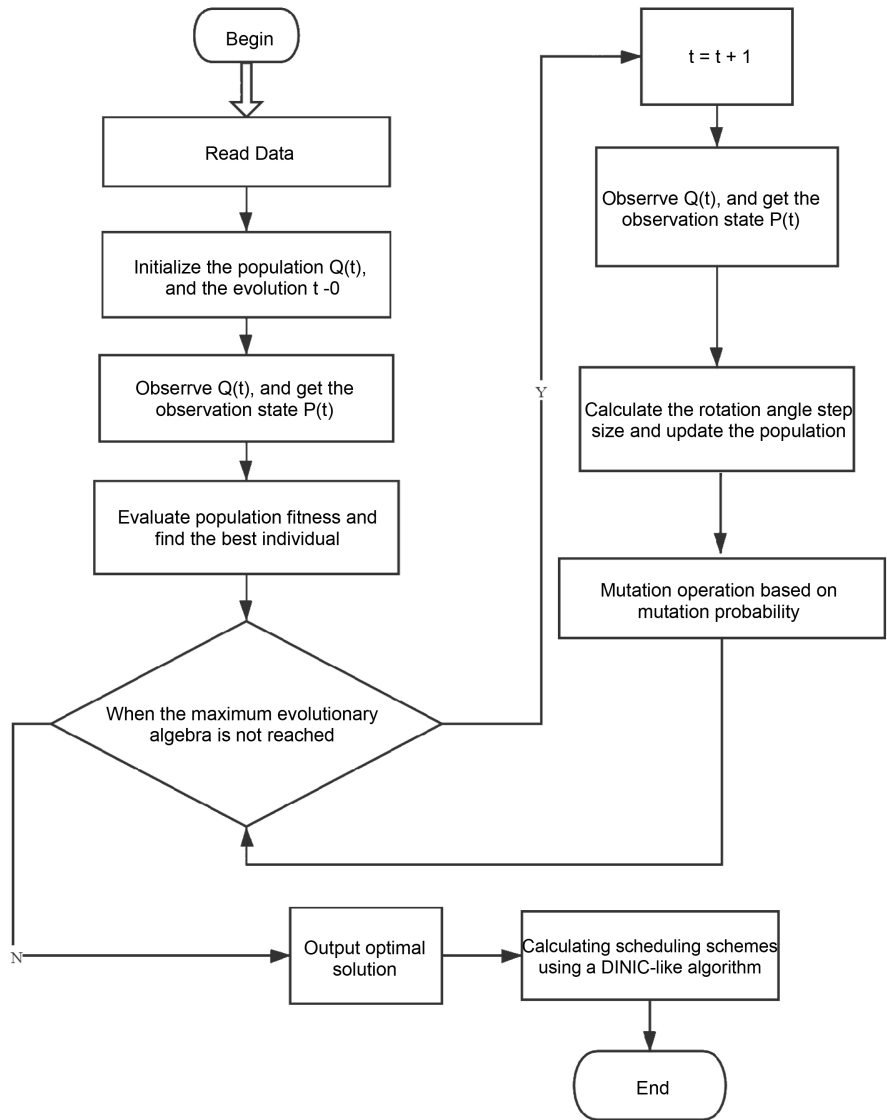
The whole algorithm includes three parts: the rotation angle adaptive adjustment mechanism, the multi-operator synergy mutation mechanism, and the population fitness evaluation. The overall algorithm flow can be described, as shown in **Figure 1**.

### 3.2. Rotation Angle Adaptive Adjustment Mechanism

The method of population update in this algorithm uses a quantum revolving door update strategy [17]. Its update matrix can be expressed as follows.

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (1)$$

In Equation (1),  $(\alpha_i, \beta_i)$  is the  $i$ -th qubit in the chromosome and  $\theta_i$  is a rotation angle. The adaptive adjustment mechanism modifies the rotation angle of individual evolution according to the fitness of different individuals. The rotation angle lookup table [18] [19] [20] used in this algorithm is shown in **Table 1**.



**Figure 1.** The flow chart for the AM-QGA.

**Table 1.** Rotation angle step lookup table.

| $x_i^j$ | $b_i$ | $\Delta\theta_i^j$ | $S(\alpha_i^j, \beta_i^j)$ |                            |                  |                 |
|---------|-------|--------------------|----------------------------|----------------------------|------------------|-----------------|
|         |       |                    | $\alpha_i^j \beta_i^j > 0$ | $\alpha_i^j \beta_i^j < 0$ | $\alpha_i^j = 0$ | $\beta_i^j = 0$ |
| 0       | 0     | $\theta_1^j = 0$   | -                          | -                          | -                | -               |
| 0       | 0     | $\theta_2^j = 0$   | -                          | -                          | -                | -               |
| 0       | 1     | $\theta_3^j = 0$   | +1                         | -1                         | 0                | $\pm 1$         |
| 0       | 1     | $\theta_4^j = 0$   | -1                         | +1                         | $\pm 1$          | 0               |
| 1       | 0     | $\theta_5^j = 0$   | -1                         | +1                         | $\pm 1$          | 0               |
| 1       | 0     | $\theta_6^j = 0$   | +1                         | -1                         | 0                | $\pm 1$         |
| 1       | 1     | $\theta_7^j = 0$   | -                          | -                          | -                | -               |
| 1       | 1     | $\theta_8^j = 0$   | -                          | -                          | -                | -               |

In **Table 1**,  $f(x)$  represents the fitness value of the individual  $x$ ,  $x_i^j$  represents the  $i$ -th position of the  $j$ -th individual, represents the  $i$ -th position of the optimal individual in the current population, and  $S(\alpha_i^j, \beta_i^j)$  represents the rotation angle in the polar coordinates.  $\theta^j$  is the rotation angle step used by the  $j$ -th individual [21]. Its definition is as follows.

$$\theta^j = \begin{cases} \frac{f_j - f_{\min}}{f_{\max} - f_{\min}} \cdot (K_2 - K_1) + K_1, & f_{\max} \neq f_{\min} \\ K_1, & f_{\max} = f_{\min} \end{cases} \quad (2)$$

In the current population, the  $i$ -th evolutionary rotation angle step and rotation direction of the  $j$ -th individual can be calculated by Equation (3).

$$\Delta\theta_i^j = \theta^j \cdot S(\alpha_i^j, \beta_i^j) \quad (3)$$

### 3.3. Fitness Function Design

In order to calculate the optimal number of shared bicycles that should be placed at each bicycle placement point, this paper sets the calculation factor of the flow rate at the placement point, and the calculation factor of the existing bicycles establishes a bicycle scheduling evaluation model evaluated by the variance of bicycle movement and designs the corresponding adaptation evaluation function.

Firstly,  $R_1$  is defined as the proportion factor of the number of existing vehicles at each drop-off point,  $R_2$  is the proportion factor of the daily traffic at each drop-off point, and the relationship between the two is shown in formula 4.

$$R_1 + R_2 = 1 \quad (4)$$

Define  $R_3$  as the weighted sum of existing vehicles and the current day's traffic at each drop-off point, as shown in Equation (5).

$$R_3 = \sum_{i=1}^n (R_1 * Num_i + R_2 * Num\_by_i) \quad (5)$$

The optimal number of bikes to be stored at each drop-off point is calculated dynamically, and the optimal number of bikes to be stored for each chromosome  $X_i$  is shown in Equation (6).  $Total\_num$  is the total number of bicycles placed in the market.

$$Best(X_i) = (R_2 * Num\_by_i + R_1 * Num_i) / R_3 * Total\_num \quad (6)$$

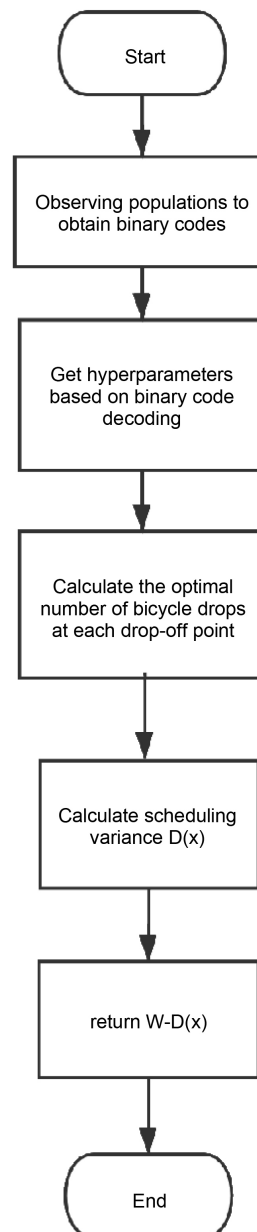
We use the variance between the optimal number of bikes placed and the daily existing bikes to account for the reasonableness of the current scheduling scheme. Define the scheduling variance for each chromosome  $X_i$  as shown in Equation (7).

$$D(X_i) = \sum_{i=1}^n ((Best(X)_i - Num_i) * (Best(X)_i - Num_i)) \quad (7)$$

For the convenience of solving problems, we translate the minimization problem into the maximization problem. For a single chromosome  $X$ , the fitness formula is designed as follows:

$$Fitness(X) = W - D(X) \quad (8)$$

The whole process of chromosome fitness evaluation is to observe the population to obtain the binary code and get two real hyperparameters  $R_1$  and  $R_2$ , by decoding the binary code, then calculate the optimal number of bicycles to be stored at each drop-off point, and the variance between the optimal number of bicycles to be dropped off and the daily existing bicycle workshop, and finally return the fitness value of chromosome. The hyperparameters  $R_1$  and  $R_2$  are used as the hyperparameters to be optimized, and an adaptive quantum genetic algorithm completes the value optimization. The whole process of the fitness evaluation function is shown in **Figure 2**.



**Figure 2.** Fitness assessment function flow chart.

### 3.4. Coding Design

In this paper, we use binary actual number coding, with 10 bits of binary coding for each chromosome, representing the number range from 0.0009765625 to 0.9990234375, and its schematic diagram is shown in **Figure 3**.

### 3.5. Calculation Scheme of Bicycle Scheduling

In operations research, directed graphs are called networks. In graph theory, a network flow is a directed graph distribution flow in which each edge has a capacity (Capacity) so that the flow of an edge does not exceed its capacity. The vertices are called nodes, and the edges are called Arcs. A flow must conform to the same restrictions as the flow in and out of a node, except for the source and sink. To calculate how to allocate the daily available bicycle scheduling to the best situation, build the network as shown in the figure, first the source point, for each. The minimum cost maximum flow class Dinic algorithm is run to calculate the minimum scheduling cost, and the scheduling scheme is output by traversing the residual network with a depth-first search.

The existing number of bikes at each drop-off point and the optimal number of dropped bikes can determine whether the bikes at that drop-off point should move in or move out. If the number of *Num\_shift* is greater than 0, the point will be moved in, and if it is less than 0, the point will be moved out, as shown in the formula.

$$Num\_shift_i = Best_i - Num_i \tag{9}$$

In this paper, the cost flow model is shown in **Figure 4** is the source point, T is the sink point, N outflow points P1-PN and M inflow points C1-CN are created, and an arc with the capacity of the number of vehicles moved out from the outflow point, and the cost of 0 is added from S to the N outflow points. Then, we add an arc with the capacity of *Total\_num*, and the cost of the distance between two points from the N outgoing points to the M incoming points in turn, and add the reverse arc, and calculate the sum of all flows into each outgoing point. Finally, an arc with the capacity of *Total\_num* and cost of 0 is created from the M migration points to the sink T in turn.

## 4. Simulations and Analysis

In order to verify the effectiveness of the algorithm, this paper compares GA, QGA, and AM-QGA and carries out the following experiments.

The software and hardware environment used in the experiment is: CPU: Intel(R) Core(TM) i7-6200U CPU4 core 2.30 GHz; memory: 8G; Windows10; Codeblocks 16.0; GCC 5.4.0.

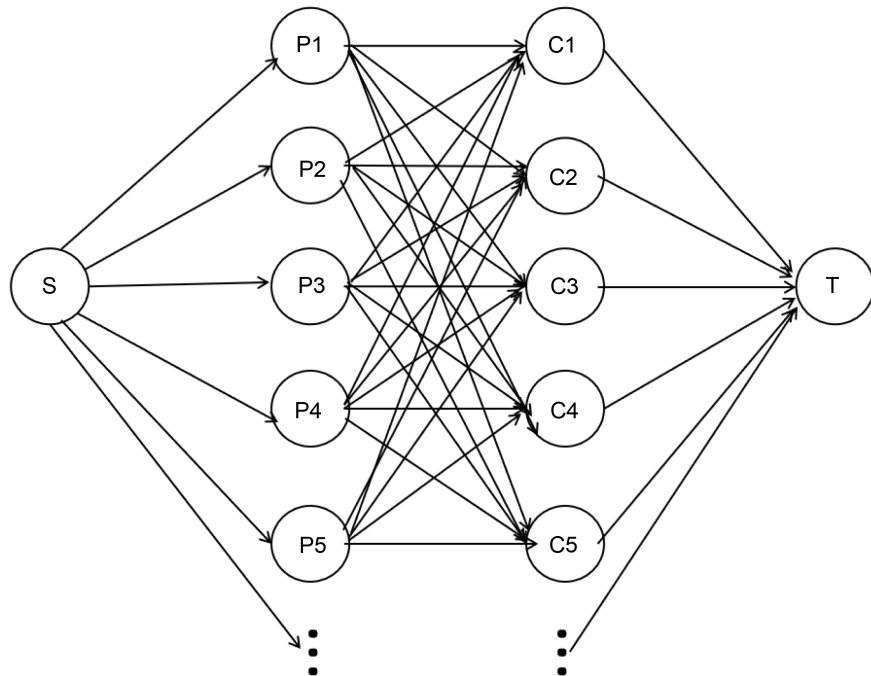
|          |          |          |          |          |          |          |          |          |           |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ | $2^{-7}$ | $2^{-8}$ | $2^{-9}$ | $2^{-10}$ |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|

**Figure 3.** Chromosome binary actual number encoding.

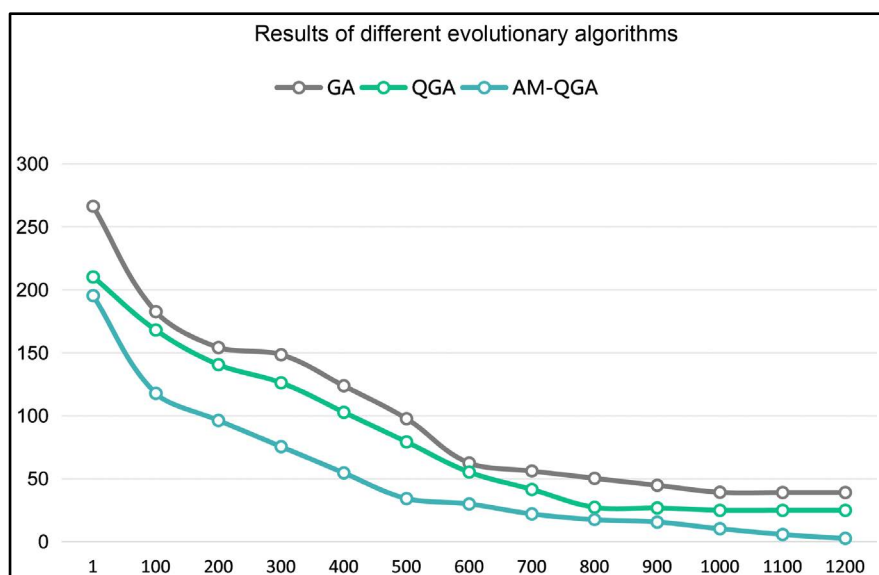


The data of this experiment uses the 2018 Mobiles travel dataset in Beijing. In the experiments of this paper, the population size of GA, QGA and AM-QGA algorithms is 50, and the number of termination generations is 1200.

From the data in **Figure 5**, we can find that the AM-QGA algorithm with the adaptive mechanism has the fastest convergence speed and the best global search ability and anti-premature ability. The final transport convergence cost of the AM-QGA algorithm is only 1/15th that of the GA algorithm and 1/9th that of the QGA algorithm. The convergence performance of QGA is second only to



**Figure 4.** Fitness assessment function flow chart.



**Figure 5.** Evolution result graph.

AM-QGA, while the convergence performance and anti-local searchability of GA are the worst. Since AM-QGA adopts an adaptive adjustment mechanism, the search efficiency of the algorithm is greatly improved, and AM-QGA does not easily fall into local optimum, and it is the best among the three algorithms. Since QGA adopts quantum bit encoding and population diversity is better than a genetic algorithm, its algorithm performance is better than the GA algorithm. From the overall view of the figure, all three algorithms gradually converge with the increase of generations, thus proving that the population evaluation algorithm proposed in this paper using the network flow bicycle scheduling algorithm is effective. Overall, AM-QGA did not fall into the optimal local solution as QGA and GA did during the experiment, probably because the algorithm can still maintain the diversity of the population well in the later stages of the search. It can be concluded that the AM-QGA algorithm has better stability and better global convergence performance after fully considering the distribution of individuals in the population and adjusting the mutation probability.

## 5. Conclusion

As a new and efficient way to solve the last mile problem, bicycle sharing is convenient, economical, green, and flexible. The advantages of being fast and flexible in the short and medium distance range can make up for the shortage of residents' walking trips and fill the gap in the coverage of public transport trips, thus improving the efficiency of residents' end trips. The main reason is that operators lack a reasonable deployment plan, and the bicycle deployment system is outdated and inefficient. The main reason is that operators do not have a reasonable deployment plan, and the bicycle deployment system is outdated and inefficient. In this paper, we use the bicycle traffic as the reference basis to establish a bicycle deployment quantity calculation model and apply the adaptive quantum genetic algorithm to optimize the parameters of the calculation model and use the Dinic network flow algorithm to calculate the best bicycle-sharing scheduling allocation scheme. After experiments, it is shown that the algorithm can steadily find out the convergent solution of the allocation scheme and achieve good results. Considering that the algorithm does not incorporate the influence of the dispatching cost factor, the algorithm model can be improved by combining the dispatching cost in the future.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Gavalas, D., Konstantopoulos, C. and Pantziou, G. (2016) Design and Management of Vehicle Sharing Systems: A Survey of Algorithmic Approaches. In: Obaidat, M.S. and Nicopolitidis, P., Eds., *Smart Cities and Homes*, Morgan Kaufmann, San Francisco, 261-289. <https://doi.org/10.1016/B978-0-12-803454-5.00013-4>

- [2] Laporte, G., Meunier, F. and Calvo, R. (2015) Shared Mobility Systems. *A Quarterly Journal of Operations Research*, **13**, 341-360. <https://doi.org/10.1007/s10288-015-0301-z>
- [3] Dell'Amico, M., Hadjicostantinou, E., Lori, M., *et al.* (2014) The Bike Sharing Rebalancing Problem: Mathematical Formulations and Benchmark Instances. *Omega*, **45**, 7-19. <https://doi.org/10.1016/j.omega.2013.12.001>
- [4] Ma, T., Liu, C. and Ergodan, S. (2015) Bicycle Sharing and Public Transit. *Transportation Research Record: Journal of the Transportation Research Board*, **2534**, 1-9. <https://doi.org/10.3141/2534-01>
- [5] Chang, H.-W., Tai, Y.-C. and Hsu, J.Y.-J. (2010) Context-Aware Taxi Demand Hotspots Prediction. *International Journal of Business Intelligence and Data Mining*, **5**, 3-18.
- [6] Li, X.l., Pan, G., Wu, Z.h., Qi, G.d., Li, S.j., Zhang, D.q., Zhang, W.s. and Wang, Z.h. (2012) Prediction of Urban Human Mobility Using Large-Scale Taxi Traces and Its Applications. *Frontiers of Computer Science*, **6**, 111-121.
- [7] Alvarez-Valdes, R., Belenguer, J.M., Benavent, E., Bermudez, J.D., *et al.* (2015) Optimizing the Level of Service Quality of a Bike-Sharing System. *Omega*, **62**, 163-175.
- [8] Chen, L.B., Zhang, D.Q., Pan, G., *et al.* (2015) Bike Sharing Station Placement Leveraging Heterogeneous Urban Open Data. *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Osaka Japan, 7-11 September 2015, 571-575.
- [9] Li, Y.X., Zheng, Y., Zhang, H.C., *et al.* (2015) Traffic Prediction in a Bike Sharing System. *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle Washington, 3-6 November 2015, 1-10.
- [10] Liu, J., Sun, L., Chen, W., *et al.* (2016) Rebalancing Bike Sharing Systems: A Multi-Source Data Smart Optimization. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 13-17 August 2016.
- [11] Wang, Y.L. (2010) Research and Implementation of Container Number Identification Based on Video. Master's Thesis, Dalian Maritime University, Dalian.
- [12] Sun, L.H. (2012) Study of Container Number Smart Recognition Algorithm. Master's Thesis, Wuhan University of Technology, Wuhan.
- [13] Zhang, M., Wang, X.H. and Xiao, J.M. (2009) A New Method of the Container Codes' Localization and Character Segmentation. 2009 *Chinese Control and Decision Conference*, Guilin, China, 17 June 2009, 2024-2028.
- [14] Wang, Y. and He, J.J. (2015) Mathematical Morphology based Algorithm for Fast Container Code Locating Process. *Computer Engineering and Design*, 2162-2166.
- [15] Huang, S.G., Weng, M.N., Shi, Y. and Liu, Q. (2018) Identification of Container Numbers Based on Computer Vision. *Port Operation*, 1-4.
- [16] Shen, H.L., Xu, J. and Zou, B. (2018) Research on Container Number Correction with Low Rank Based on MSER. *Chinese Journal of Engineering Mathematics*, **35**, 123-136.
- [17] Han, K. H. and Kim, J.H. (2000) Genetic Quantum Algorithm and Its Application to Combinatorial Optimization Problem. *Proceeding of the 2000 Congress on Evolutionary Computation*, **2**, 1354-1360.
- [18] Han, K. H. and Kim, J.H. (2002) Quantum-Inspired Optimization Algorithm for a Class of Combinatorial Optimization. *IEEE Transactions on Optimization Compu-*

- tation*, **16**, 580-593. <https://doi.org/10.1109/TEVC.2002.804320>
- [19] Han, K. H. and Kim, J.H. (2004) Quantum-Inspired Optimization Algorithms with a New Termination Criterion, H-Gate, and Two-Phase Scheme. *IEEE Transactions on Optimization Computation*, **8**, 156-169. <https://doi.org/10.1109/TEVC.2004.823467>
- [20] Li, B. and Wang, L. (2007) A Hybrid Quantum-Inspired Genetic Algorithm for Multiobjective Flow Shop Scheduling. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, **37**, 576-591. <https://doi.org/10.1109/TSMCB.2006.887946>
- [21] Qu, Z.J., Chen, Y.H., Li, P., Liu, X.H. and Li, C.H. (2019) Cooperative Evolution of Multiple Operators Based Adaptive Parallel Quantum Genetic Algorithm. *Acta Electronica Sinica*, **47**, 266-273.