

*Research Article*

# **Multiobjective Quantum Evolutionary Algorithm for the Vehicle Routing Problem with Customer Satisfaction**

**Jingling Zhang,<sup>1</sup> Wanliang Wang,<sup>2</sup>  
Yanwei Zhao,<sup>1</sup> and Carlo Cattani<sup>3</sup>**

<sup>1</sup> College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310014, China

<sup>2</sup> College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014, China

<sup>3</sup> Department of Mathematics, University of Salerno, Via Ponte Don Melillo, 84084 Fisciano, Italy

Correspondence should be addressed to Yanwei Zhao, [zyw@zjut.edu.cn](mailto:zyw@zjut.edu.cn)

Received 20 August 2012; Accepted 10 October 2012

Academic Editor: Sheng-yong Chen

Copyright © 2012 Jingling Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The multiobjective vehicle routing problem considering customer satisfaction (MVRPCS) involves the distribution of orders from several depots to a set of customers over a time window. This paper presents a self-adaptive grid multi-objective quantum evolutionary algorithm (MOQEA) for the MVRPCS, which takes into account customer satisfaction as well as travel costs. The degree of customer satisfaction is represented by proposing an improved fuzzy due-time window, and the optimization problem is modeled as a mixed integer linear program. In the MOQEA, nondominated solution set is constructed by the Challenge Cup rules. Moreover, an adaptive grid is designed to achieve the diversity of solution sets; that is, the number of grids in each generation is not fixed but is automatically adjusted based on the distribution of the current generation of nondominated solution set. In the study, the MOQEA is evaluated by applying it to classical benchmark problems. Results of numerical simulation and comparison show that the established model is valid and the MOQEA is effective for MVRPCS.

## **1. Introduction**

The vehicle routing problem (VRP) is one of the most important and widely studied combinatorial optimization problems, with many real-world applications in logistic distribution and transportation [1]. Since the VRP was firstly proposed by Dantzig and Ramser in 1959 [2], it has been focused in the field of operational research and combinatorial optimization [3–5].

The aim of VRP is to find optimal routes for a fleet of vehicles serving a set of customers with known demands. Each customer is serviced exactly once and must be assigned a

satisfactory vehicle without exceeding vehicle capacities. A solution for this problem is to find out a set of minimum cost routes that are used to represent vehicles distribution and clients' permutation. However, current studies on VRP [6, 7] are mainly focused on the single objective problem and the objective is to optimize the number of vehicles dispatched and the travel distance, that is, reducing the service costs of the provider.

Actually, to achieve competitive advantage, a service provider needs to consider not only service costs but also service quality that can determine customers' satisfaction. Most of the research on multiobjective VRP (MOVRP) does not take into account this objective, only focusing on the traditional objectives of minimum costs and the length of the longest route [8, 9]. Hong and Park [10] constructed a linear goal programming (GP) model for the biobjective vehicle routing with time window constraints (BVRPTW) and proposed a heuristic algorithm to relieve the computational burden inherent to the application of the GP model. Zitzler and Thiele [11] proposed a multiobjective evolutionary algorithm based on the Pareto approach for VRP. Lim and wang [12] proposed a method to deal with MOVRP by assigning different weights of the objectives. Tan et al. [13] proposed a hybrid multiobjective evolutionary algorithm (HMOEA) that incorporates various heuristics for local exploitation in the evolutionary search and the concept of Pareto's optimality for solving multiobjective optimization in vehicle routing problem with time window constraints (VRPTW). Garcia-Najera and Bullinaria [14] studied an improved multiobjective evolutionary algorithm for VRP with time windows.

The VRPTW is developed from VRP and has been widely studied in the last decade [15–19]. In the VRPTW, each customer has a time window with values about the deadline and the earliest time constraints for the service he/she requires. Thus this problem involves a routing combination and scheduling component. Routes must be designed to minimize the total cost, but, at the same time, scheduling must be performed to ensure time feasibility.

In practice, this time window actually does not well describe customers' satisfaction. A major reason is that customers are asked to provide a fixed time window for service, but in reality they really hope to be served at a desired time. Cheng and Gen [20, 21] called such a desired time the due-time and proposed to use the concept of fuzzy due-time to replace this time window because, as they claimed, it can describe customers' satisfaction better. Thus customers' satisfaction can be also described as a convex fuzzy number [22–24].

Cheng and Gen [20, 21] introduced the fuzzy due-time, used triangle fuzzy numbers to describe customers' satisfaction, and solved the VRP by genetic algorithms (GAs). Zhang et al. [25] proposed a multiobjective fuzzy VRP and used trapezoid fuzzy numbers to describe customers' satisfaction. Jia [26] used multiobjective hybrid GA for this problem. Wu [27] studied the open VRP based on customers' satisfaction. Lin [28] proposed a GA-based multiobjective decision-making method for optimal vehicle transportation, which is focused on a fuzzy vehicle routing and scheduling problem (FVRSP) based on five attributes, namely, space utility, service satisfaction, waiting time, delay time, and transportation distance. Wang and Li [29] proposed a hybrid algorithm based on GA and incorporated some methods based on greedy algorithms to solve the MOP model for which depot desires and clients expectations are considered simultaneously.

The above studies use the weighted sums of objectives to solve the multiobjective problem; the higher an objective's importance, the larger its corresponding weight coefficient. In general, no single solution can attain the optimum of all objectives at the same time. Therefore, it is desirable to obtain a set of Pareto optimal solution, that is, the Pareto set. The points in the objective space that correspond to the results in the set are usually called the Pareto front.

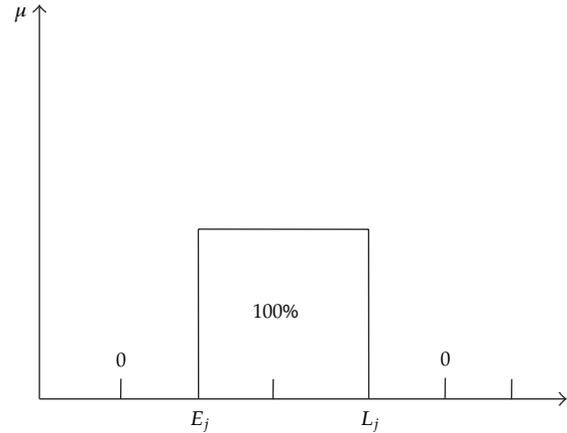


Figure 1: Traditional time windows.

In this paper, a self-adaptive grid multiobjective quantum evolutionary algorithm (MOQEA) is proposed to solve the MVRPCS problem. In particular, the quantum evolutionary algorithm (QEA) is used in the MOQEA due to its high efficiency, convergence speed, strong full-searching optimization ability [30]. With the MOQEA, an optimal or a nearly optimal set of vehicle routes solution with the minimal total travel cost and maximal customers' satisfaction can be obtained by decoding the chromosome and simultaneously obtains several solution sets. This method can support the dispatcher to more efficiently determine how to distribute the shipment to serve customers by available vehicles.

The remainder of the paper is organized as follows. Section 2 presents the mathematical model for the MVRPCS with consideration of customer satisfaction. Section 3 describes the proposed MOQEA in detail. In Section 4, the application of the proposed algorithm to a classic problem is introduced and the simulation results are discussed and compared with an early algorithm. Finally, conclusions are given in Section 5.

## 2. Model for the MVRPCS

### 2.1. Representation of Customer Satisfaction

In traditional VRP, customers' time constraints are represented by time windows as shown in Figure 1. For this method, if the customer is serviced at a time within the window, then the satisfaction degree is 100%, otherwise the satisfaction degree is 0. It is really unrealistic to measure the degree of satisfaction accordingly based on time window because customer satisfaction is not necessarily the same if they are serviced at different times within the window. In fact, service time can be divided into two categories; namely, the service time can be tolerated and the desirable service time.

Fuzzy due-time windows have been introduced to describe different degrees of satisfaction. Generally, the tolerable service time for customer  $i$  can be described as  $[E_i, L_i]$ , where  $E_i$  is the earliest time and  $L_i$  is the latest time. For example, in [20, 25] customers satisfaction is represented by the triangular fuzzy number, as shown in Figure 2. When the desirable service time is taken into account, customer satisfaction can be described using trapezoidal fuzzy number. In [26], this method is used and the desirable service time is

described by the due-time  $[a_i, b_i]$ , as shown in Figure 3. In this case, customer satisfaction is zero no matter vehicles arrive early or late if the expected service time slot is not achieved.

In this paper, an improved fuzzy due-time window is proposed, as shown in Figure 4. In this method, if a customer is served at the desired time, the degree of satisfaction is 1; otherwise, the degree of satisfaction decreases as the difference between the actual service time and the desired time increases. The degree of satisfaction for the cases in which vehicles arrive early than the earliest expected time is not zero, but equals to that of the case when the earliest expected time is met. The degree of customer satisfaction is represented by the membership function of the improved fuzzy due-time window, that is, an improved trapezoidal fuzzy number. For customer  $i$ , mark his/her satisfaction as  $\mu_i(t_i)$  for a given service time  $t_i$ . Then the degree of satisfaction can be calculated using the membership function as follows:

$$\mu_i(t_i) = \begin{cases} \exp k(E_i - a_i), & t_i < E_i \\ \exp k(t_i - a_i), & E_i < t_i < a_i \\ 1, & a_i < t_i < b_i \\ \exp k(b_i - t_i), & b_i < t_i < L_i \\ 0, & t_i > L_i. \end{cases} \quad (2.1)$$

## 2.2. Mathematics Model

The MVRPCS can be described as follows: there are  $M$  depots each of which has  $K_m$  ( $m = 1, 2, \dots, M$ ) vehicles with a capacity of  $b_k$ . These vehicles will be dispatched to  $L$  customers to meet their demands.

Mark the demand of customer  $i$  as  $d_i$  ( $i = 1, 2, \dots, L$ ), and assume  $d_i < b_k$ . Each customer can be served by any vehicles from a depot, but the service will only take place once. In addition, each vehicle can complete the shipping task without having to return to the original depot. Thus an appropriate vehicle scheduling program is required to meet the needs of all customers. The meanings of variables used in this research are described as follows.

Customer number is  $1, 2, \dots, L$ . Depot number is  $L + 1, L + 2, \dots, L + M$ .

Fixed cost of sending a vehicle is  $F_k$  ( $m \in \{L + 1, L + 2, \dots, L + M\}; k \in \{1, 2, \dots, K_m\}$ ).

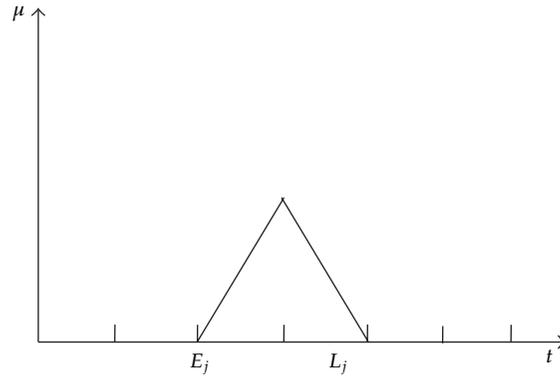
Distribution cost from customer  $i$  to customer  $j$  is  $c_{ij}$ , and  $i, j \in (1, 2, \dots, L, L + 1, \dots, L + M)$ .

Time window of customer  $i$  is  $[E_i, L_i]$ ; the arrival time of customer  $i$  is  $t_i$ ; the travel time from  $i$  to  $j$  is  $t_{ij}$ ,  $s_i$  is the service time of customer  $i$  and  $\mu_i(t_i)$  is the degree of satisfaction for customer  $i$ .

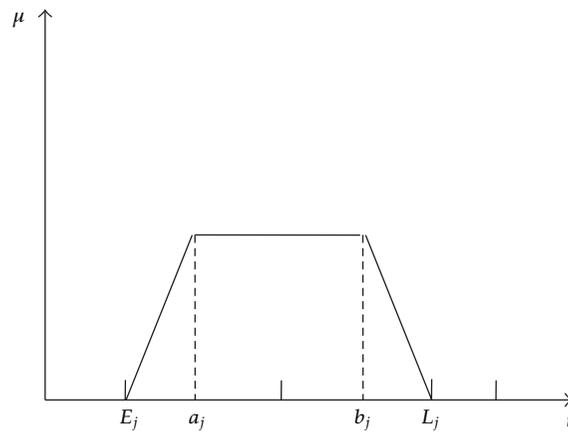
If vehicle  $k$  travels directly from customer  $i$  to customer  $j$  and arrives too early at  $j$ , it will wait;  $w_j(t_j)$  is the waiting time of customer  $j$  for a vehicle. Thus, the MVRPCS model can be established, as discussed in detail in the following paragraphs. Two decision variables are defined as follows:

$$x_{ijk}^m = \begin{cases} 1 & \text{vehicle } k \text{ from } i \text{ to } j \text{ of depot } m, \\ 0 & \text{other,} \end{cases} \quad (2.2)$$

$$y_k^m = \begin{cases} 1 & \text{vehicle } k \text{ dispatched from depot } m, \\ 0 & \text{other.} \end{cases}$$



**Figure 2:** Triangular fuzzy number.



**Figure 3:** Trapezoidal fuzzy number.

This problem has two objectives that are determined by both the service quality and the service costs. Considering the two objectives can help the dispatcher make better decision without compromising any of the two objectives. The dispatcher can still have preferences on different objectives by selecting different parameters.

The service quality objective is to maximize average customers satisfaction:

$$\max \frac{1}{L} \sum_{i=1}^L \mu_i(t_i). \quad (2.3)$$

This objective is equivalent to minimizing the average customer dissatisfaction:

$$\min \left( 1 - \frac{1}{L} \sum_{i=1}^L \mu_i(t_i) \right). \quad (2.4)$$

The other objective of the service costs is to minimize travel cost, fixed cost and waiting cost. For this objective, the fixed cost of sending a vehicle is considered because

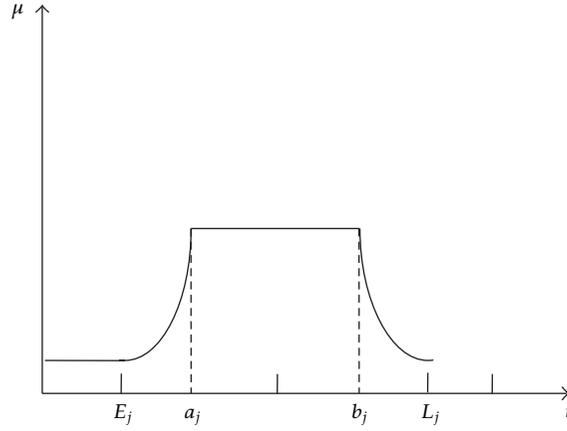


Figure 4: Improved trapezoidal fuzzy number.

vehicles in operation have depreciation and fuel consumption. Also the fixed cost is related to the number of vehicle, that is, the more vehicles, the higher fixed cost. To the best of our knowledge, no previous work has been done to take into account this fixed cost when solving multiobjective VRP with fuzzy due-time. Based on the above discussion, the mathematical model for the MVRPCS can be established as follows:

$$\min Z = \sum_{i=1}^{L+M} \sum_{j=1}^{L+M} \sum_{k=1}^{K_m} \sum_{m=1}^M x_{ijk}^m c_{ij} + \sum_{k=1}^{K_m} \sum_{m=1}^M y_k^m F_k + \sum_{i=1}^L w_i(t_i) \quad (2.5)$$

s.t.,

$$\sum_{i=1}^L \sum_{j=1}^{L+M} x_{ijk}^m d_i \leq b_k, \quad m \in \{L+1, L+2, \dots, L+M\}; k \in \{1, 2, \dots, K_m\}, \quad (2.6)$$

$$\sum_{j=1}^L \sum_{k=1}^{K_m} x_{ijk}^m \leq K_m, \quad i = m \in \{L+1, L+2, \dots, L+M\}, \quad (2.7)$$

$$\sum_{j=1}^{L+M} \sum_{k=1}^{K_m} \sum_{m=1}^M x_{ijk}^m = 1, \quad i \in \{1, 2, \dots, L\}, \quad (2.8)$$

$$\sum_{i=1}^{L+M} \sum_{k=1}^{K_m} \sum_{m=1}^M x_{ijk}^m = 1, \quad j \in \{1, 2, \dots, L\}, \quad (2.9)$$

$$t_j = \begin{cases} t_i + s_i + t_{ij}, & t_i \geq E_i, \\ E_i + s_i + t_{ij}, & t_i < E_i, \end{cases} \quad (2.10)$$

$$t_j < L_j, \quad (2.11)$$

$$w_j(t_j) = \text{Max}\{0, E_j - t_j\}. \quad (2.12)$$

In the model, formula (2.6) assures the carrying capacity of each vehicle; formula (2.7) assures the number of vehicles that are dispatched from each depot does not exceed the capacity of the depot; formulas (2.8) and (2.9) assure each client is served only by one vehicle; formulas (2.10), (2.11), and (2.12) assure customers can be served within time windows; and from (2.10) and (2.12), the waiting cost for the vehicle can be obtained.

### 3. MOQEA for the MVRPCS

In this research, the multiobjective optimization method of Pareto optimal solution [31] is used. Its main advantage is to approximate the Pareto front in order to provide a set of equivalent solutions to the decision maker [32, 33]. The algorithm to solve multiobjective optimization problem of Pareto optimality involves two main questions:

- (i) how to construct a Pareto optimal solution set, namely non-dominated solutions set, and make it close to the Pareto optimal front as much as possible?
- (ii) how to attain the diversity and variety of solutions?

To address these two issues, a self-adaptive grid multiobjective quantum evolutionary algorithm (MOQEA) to solve the MVRPCS problem is proposed. The method of constructing non-dominated solution set and attaining the diversity and variety of solutions is described in the following sections.

#### 3.1. Constructing Nondominated Solution Set

In this paper, the Challenge Cup rule [34] is used to construct non-dominated solution set. Assuming  $P$  is an evolution group and  $Q$  is a constructed set, let  $Q = P$  initially. Assume  $Ndset$  is a non-dominated set which is empty initially. The basic idea of this rule is to take any individual  $x$  from  $Q$ , followed by comparison with all other individuals  $y$  in  $Q$ : if  $x > y$ , then clear  $y$  from  $Q$ ; if  $y > x$ , then use  $y$  to replace  $x$ , and then  $y$  is the new champion and continues to be compared with other individuals. After a comparison,  $cluster(x) = \{y \mid x > y, x, y \in P\}$  is formed, where  $x$  is the smallest element. Add  $x$  to  $Ndset$  and continue the comparison until  $Q$  is empty.

#### 3.2. Method of Attaining the Diversity and Variety of Set Based on Self-Adaptive Grid

To attain the variety of the set, the individual space is divided into several small areas each of which is called a grid, as shown in Figure 5. Thus each individual is associated with a grid in the figure, and the number of individuals in each grid can be defined as extrusion coefficient.

A grid is used in many different ways to maintain the diversity and variety of solutions. Knowles and Crone [35] proposed a pareto archived evolution strategy for pareto multiobjective optimization. Corne et al. [36] proposed a pareto envelope-based selection algorithm for multiobjective optimization.

When a grid contains more than one individual, these individuals are treated as the same solution. As such, the size of grid is very important. When the grid is too large, multiple individuals will exist in the same grid, and the resultant solution distribution is not accurate.

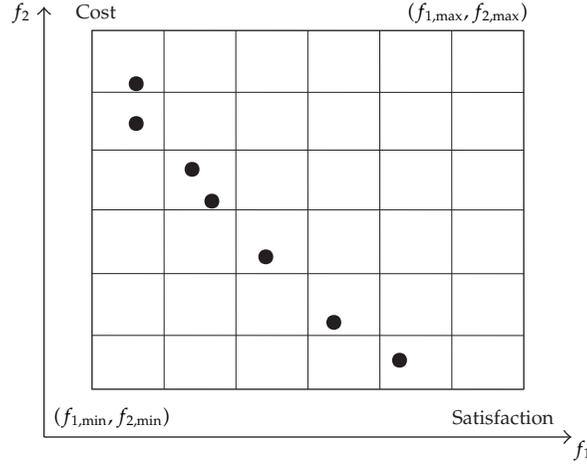


Figure 5: Individual space divided by grid.

When the grid is too small, it is likely that there are no individuals in some grids, and so it takes longer computation time though the resultant solution distribution is accurate. Therefore, computation time and accuracy must be traded off when determining the grid size.

There are two objective functions in this optimization problem. The range of the customers' satisfaction is  $[0, 1]$ , and the range of travel costs and waiting costs is changed with the experimental data. The boundaries of the grid in the target space can be determined by the range of the above two objective functions.

In this paper, the number of grids is not fixed in each generation but automatically adjusted based on the distribution of the current generation of non-dominated solution set. The grid boundary is a fixed value. In the process of each evolutionary generation, the number of grid is adjusted by the  $D$ -value between the maximum and minimum values of each dimension in the non-dominated solution set. The method of self-adaptive grid is designed as follows.

The two objections can be described by  $k = \{1, 2\}$ . Mark the number of each dimension grid in generation 1 as  $N_{1,k}$ ,  $Ndset_1$  as the non-dominated solutions set in generation 1, and  $Ndset_t$  as the non-dominated solutions set in generation  $t$ . If  $f_{t,k}$  is the objection of each dimension, then the  $D$ -value of each dimension can be described as follows:

$$\text{generation 1: } \forall k = \{1, 2\}$$

$$\text{Diffence}_{1,k} = \max\{f_{1,k} \mid f_1 \in Ndset_1\} - \min\{f_{1,k} \mid f_1 \in Ndset_1\}, \quad (3.1)$$

$$\text{generation } t : \forall k = \{1, 2\}$$

$$\text{Diffence}_{t,k} = \max\{f_{t,k} \mid f_t \in Ndset_t\} - \min\{f_{t,k} \mid f_t \in Ndset_t\}. \quad (3.2)$$

The number of each dimension grid in generation  $t$  is

$$N_{t,k} = \left[ N_{1,k} * \frac{\text{Diffence}_{1,k}}{\text{Diffence}_{t,k}} \right], \quad \text{where } [ ] \text{ means being rounded.} \quad (3.3)$$

To keep the diversity and variety of the non-dominated solution set, choose the individual with the maximum extrusion coefficient and delete it form the non-dominated solution set.

### 3.3. Representation

Quantum evolutionary algorithm (QEA) [37] is based on the concept and principles of quantum computing such as a quantum bit and superposition of states. Instead of binary and numeric representation, QEA uses a Q-bit chromosome as a probabilistic representation and a Q-bits individual is modeled by a string of Q-bits.

The smallest unit of information stored in two-state quantum computer is called a Q-bit, which may be in the "1" state, or in the "0" state, or in any superposition of the two. The state of a Q-bit can be represented as follows

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (3.4)$$

where  $\alpha$  and  $\beta$  are complex numbers.  $|\alpha|^2$  and  $|\beta|^2$  donate the probabilities that the Q-bit will be found in the "0" state and in the "1" state, respectively. Normalization of the state to unity is used to meet  $|\alpha|^2 + |\beta|^2 = 1$ .

So a Q-bit individual with a string of  $m$  Q-bits can be expressed as follows

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{bmatrix}, \quad (3.5)$$

where  $|\alpha_i|^2 + |\beta_i|^2 = 1$ ,  $i = 1, 2, \dots, m$ .

The main advantage of the representation is that any linear superposition of solutions can be represented. For example, a three-Q-bit system can contain the information of eight states. QEA with Q-bit representation has a better characteristic of population diversity than other representations, as it can represent linear superposition of state's probabilities.

In this paper, a method of converting integer representation to Q-bit representation is designed. For the MVRPCS with  $L$  customers, the representation of customers route is described as the permutation of  $1 \sim L$ . Note the permutation of  $1 \sim L$  is  $(g_1, g_2, \dots, g_L)$ , and represent each gene  $g_j (j = 1, 2, \dots, L)$  as a string of  $r$ -Q-bit, then  $L$  groups  $n$ -Q-bit are obtained. So, a quantum individual is described as a  $rL \times 2$  Q-bit matrix. Here  $r = \{m\}$ , where  $\{m\}$  is the smallest integer not less than  $m$  and  $2^m \geq L$ , thus we can get  $m \geq \log_2 L$ .

### 3.4. Decoding Method

The "Customers permutation Route First, Vehicles distribution Cluster Second" rule is adopted for decoding.

**Table 1:** Lookup table of rotation angle.

$r_i$	$y_i$	$f(y) > f(r)$	$\Delta\theta_i$	$s(\alpha_i, \beta_i)$			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	False	0	0	0	0	0
0	0	True	$\theta$	+1	-1	0	$\pm 1$
0	1	False	0	0	0	0	0
0	1	True	$\theta$	-1	+1	$\pm 1$	0
1	0	False	$\theta$	-1	+1	$\pm 1$	0
1	0	True	$\theta$	-1	+1	0	$\pm 1$
1	1	False	0	0	0	0	0
1	1	True	$\theta$	-1	+1	0	$\pm 1$

- (1) Firstly, get the customers permutation route. The solution of MVRPCS is a permutation of all customers and Q-bit representation cannot be evaluated directly. So it should be converted to permutation for evaluation.

The Q-bit string is firstly converted to binary string  $\gamma$ . Specifically, a random number  $\eta$  between  $[0, 1]$  is generated, if the  $i$ th bit  $\alpha_i$  of Q-bit string satisfied  $|\alpha_i|^2 > \eta$ , then let the corresponding bit  $\gamma_i$  of the binary string  $\gamma$  be 1, otherwise let it be 0. Then the binary representation is converted to integer representation, which is viewed as random key representation [38], and customer permutation is constructed based on the generated random key. If two random key values are different, the smaller random key denotes the customer with smaller number; otherwise, let the one that first appears denote the customer with smaller number.

- (2) Secondly, distribute the vehicles and get the subroute. A vehicle is dispatched to service customers according to the customers' permutation route, if the vehicle cannot serve the next customer when it cannot meet the time window or loading capacity constraints, a new vehicle will be dispatched. For example, the customers' permutation route is [8 5 9 3 4 1 2 6 7], the 3 depots notes [10 11 12], use this decoding method, the subroute is: Route 1: 11-8-5-9; Route 2: 10-3-4-1; Route 3: 12-2-6-7.

### 3.5. Strategy of Updating by Q-Gate

In the MOQEA, a Q-gate is an evolution operator which is the same as the QEA in [39]. A rotation gate is often used to update a Q-bit individual, as shown in (3.6)

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = U(\theta_i) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}, \quad (3.6)$$

where  $[\alpha_i, \beta_i]^T$  is the  $i$ th Q-bit and  $\theta_i$  is the rotation angle of each Q-bit.  $\theta_i = s(\alpha_i, \beta_i)\Delta\theta_i$ .

The lookup table of  $\theta_i$  is shown in Table 1. In this paper, a non-dominated solution is randomly selected as the current objective solution from the non-dominated solution set. In the multiobjective optimization, it is unable to find the optimal value with all objectives met. So we need to choose an objective solution  $y$  for each individual  $r$  only to find the non-dominated solution set.

**Table 2:** Optimization results of coefficient  $k$  ( $s = 0.2$ ).

Problem	pr02 (96, 4)			pr07 (72, 6)		
	VN	CS	Cost	VN	CS	Cost
0.08	16	0.402	4539	15	0.492	4116
0.06	15	0.378	4045	14	0.424	3904
0.05	14	0.355	3906	13	0.399	3331
0.02	15	0.364	3998	14	0.431	3969

**Table 3:** Optimization results of constant  $s$  ( $k = 0.06$ ).

Problem	pr02 (96, 4)			pr07 (72, 6)		
	VN	CS	Cost	VN	CS	Cost
0.1	15	0.381	4013	14	0.435	3391
0.2	14	0.357	3902	13	0.389	3367
0.3	15	0.361	3926	15	0.486	4124
0.4	16	0.411	4625	15	0.498	4102

In the above table,  $\Delta\theta_i$  is the magnitude of rotation angle.  $s(\alpha_i, \beta_i)$  is the sign of  $\theta_i$  that determines the direction.  $r_i$  and  $y_i$  are the  $i$ th Q-bit of the binary solution in individual  $r$  and the objective solution  $y$  respectively.  $f(\bullet)$  is the objective.  $\theta$  is the rotation angle of size, which affects the convergence speed and search capability. In this paper, a method of dynamically adjusting the rotation angle is proposed, that is;  $\theta$  will be changed with the extrusion coefficient as discussed in Section 3:

$$\theta = \frac{0.5s}{n}\pi, \quad (3.7)$$

where  $n$  is the extrusion coefficient.  $s$  is a constant in the range  $[0, 1]$ . From (3.7), we can see when the extrusion coefficient  $n$  is small, the rotation angle  $\theta$  is big to accelerate the convergence speed; when the Extrusion Coefficient  $n$  is big, and the search step size will be reduced to enhance the solution diversity.

### 3.6. Procedure of MOQEA

The flow chart of the MOQEA for this problem is illustrated in Figure 6.

The detailed procedure of the MOQEA is as follows.

*Step 1.* Let  $t = 0$  and randomly generate an initial population  $Q(t) = \{q_1^t, q_2^t \cdots q_n^t\} = \left\{ \begin{matrix} \alpha_1^t & \alpha_2^t & \cdots & \alpha_n^t \\ \beta_1^t & \beta_2^t & \cdots & \beta_n^t \end{matrix} \right\}$ , that is, randomly generate any value in  $[0, 1]$  for  $\alpha_i$  and  $\beta_i$ , where  $q_j^t$  denotes the  $j$ th individual in the  $t$ th generation. At the same time, construct initially empty external set  $O(t)$  with size of  $m$ .

*Step 2.* Convert  $Q(t)$  to binary population  $R(t)$ , then convert it to integer population  $P(t)$ .

*Step 3.* According to the decoding method to get the subroute, evaluate the objectives to get the MVRPCS solution set  $M(t) = \{(f_{11}^t, f_{21}^t) \cdots (f_{1n}^t, f_{2n}^t)\}$ , where  $(f_{1j}^t, f_{2j}^t)$  is the  $j$ th value of the two objectives in the  $t$ th generation. And let  $M(t)$  be the construction set.

**Table 4:** The distance and demand of each client.

CN	Coordinate/km		ST	De/ <i>t</i>	TW
1	-29.730	64.136	2	12	[399 525]
2	-30.664	5.463	7	8	[121 299]
3	51.642	5.469	21	16	[389 483]
4	-13.171	69.336	24	5	[204 304]
5	-67.413	68.323	1	12	[317 458]
6	48.907	6.274	17	5	[160 257]
7	5.243	22.260	6	13	[170 287]
8	-65.002	77.234	5	20	[215 321]
9	-4.175	-1.569	7	13	[80 233]
10	23.029	11.639	1	18	[90 206]
11	25.482	6.287	4	7	[397 525]
12	-42.615	-26.392	10	6	[271 420]
13	-76.672	99.341	2	9	[108 266]
14	-20.673	57.892	16	9	[340 462]
15	-52.039	6.567	23	4	[226 377]
16	-41.376	50.824	18	25	[446 604]
17	-91.943	27.588	3	5	[444 566]
18	-65.118	30.212	15	17	[434 557]
19	18.597	96.716	13	3	[319 460]
20	-40.942	83.209	10	16	[192 312]
21	-37.756	-33.325	4	25	[414 572]
22	23.767	29.083	23	21	[371 462]
23	-43.030	20.453	20	14	[378 472]
24	-35.297	-24.896	10	19	[308 477]
25	-54.755	14.368	4	14	[329 444]
26	-49.329	33.374	2	6	[269 377]
27	57.404	23.822	23	16	[398 494]
28	-22.754	55.408	6	9	[257 416]
29	-56.622	73.340	8	20	[198 294]
30	-38.562	-3.705	10	13	[375 467]
31	-16.779	19.537	7	10	[200 338]
32	-11.560	11.615	1	16	[456 632]
33	-46.545	97.974	21	19	[72 179]
34	16.229	9.320	6	22	[182 282]

Table 4: Continued.

CN	Coordinate/km		ST	De/ $t$	TW
35	1.294	7.349	4	14	[159 306]
36	-26.404	29.529	13	10	[321 500]
37	4.352	14.685	9	11	[322 430]
38	-50.665	-23.126	22	15	[443 564]
39	-22.833	-9.814	22	13	[207 348]
40	-71.100	-18.616	18	15	[457 588]
41	-7.849	32.074	10	8	[203 382]
42	11.877	-24.933	25	22	[75 167]
43	-18.927	-23.730	23	24	[459 598]
44	-11.920	11.755	4	3	[174 332]
45	29.840	11.633	9	25	[130 225]
46	12.268	-55.811	17	19	[169 283]
47	-37.933	-21.613	10	21	[115 232]
48	42.883	-2.966	17	10	[414 531]

*Step 4.* Use the formulas (2.4) and (2.5) to evaluate the domination of each  $(f_{1j}^t, f_{2j}^t)$  in construction set  $M(t)$  and use the method of Challenge Cup rules to construct non-dominated solution set  $NDset(t)$ .

*Step 5.* When  $t = 0$ , reproduce  $NDset(0)$  to the external set  $O(t)$ . When  $t > 0$ , if a certain individual in  $NDset(t)$  dominates one solution in  $O(t)$  delete the solution and join the individual into the  $O(t)$ , and if a solution in  $O(t)$  dominates a certain individual in  $NDset(t)$ , the solutions in  $O(t)$  does not change; otherwise, join the individual of  $NDset(t)$  into the  $O(t)$ .

*Step 6.* Adjust the size of  $O(t)$  to the number of  $m$  and satisfy the distribution of the non-dominated solution set. The method is discussed in detail as follows. If the size of  $O(t)$  is less than  $m$ , randomly select individual of  $NDset(t)$  into the  $O(t)$  until the size of  $O(t)$  is  $m$ ; otherwise, use the self-adaptive grid method to keep the diversity and variety of  $O(t)$ , divide the individual space of  $O(t)$  into several small grids, choose the grid with the maximum extrusion coefficient, and randomly delete a solution set from it.

*Step 7.* If the stopping condition is satisfied, then output the Pareto set; otherwise, go on to the following steps.

*Step 8.* Randomly select some individuals from the Q-bit  $Q(t)$ , which is instated of the individuals from  $Q(t)$  corresponding to  $O(t)$ .

*Step 9.* Use (3.6) to perform rotation operation for  $Q(t)$  to generate  $Q(t + 1)$ .

*Step 10.* Let  $t = t + 1$  and go back to Step 2.

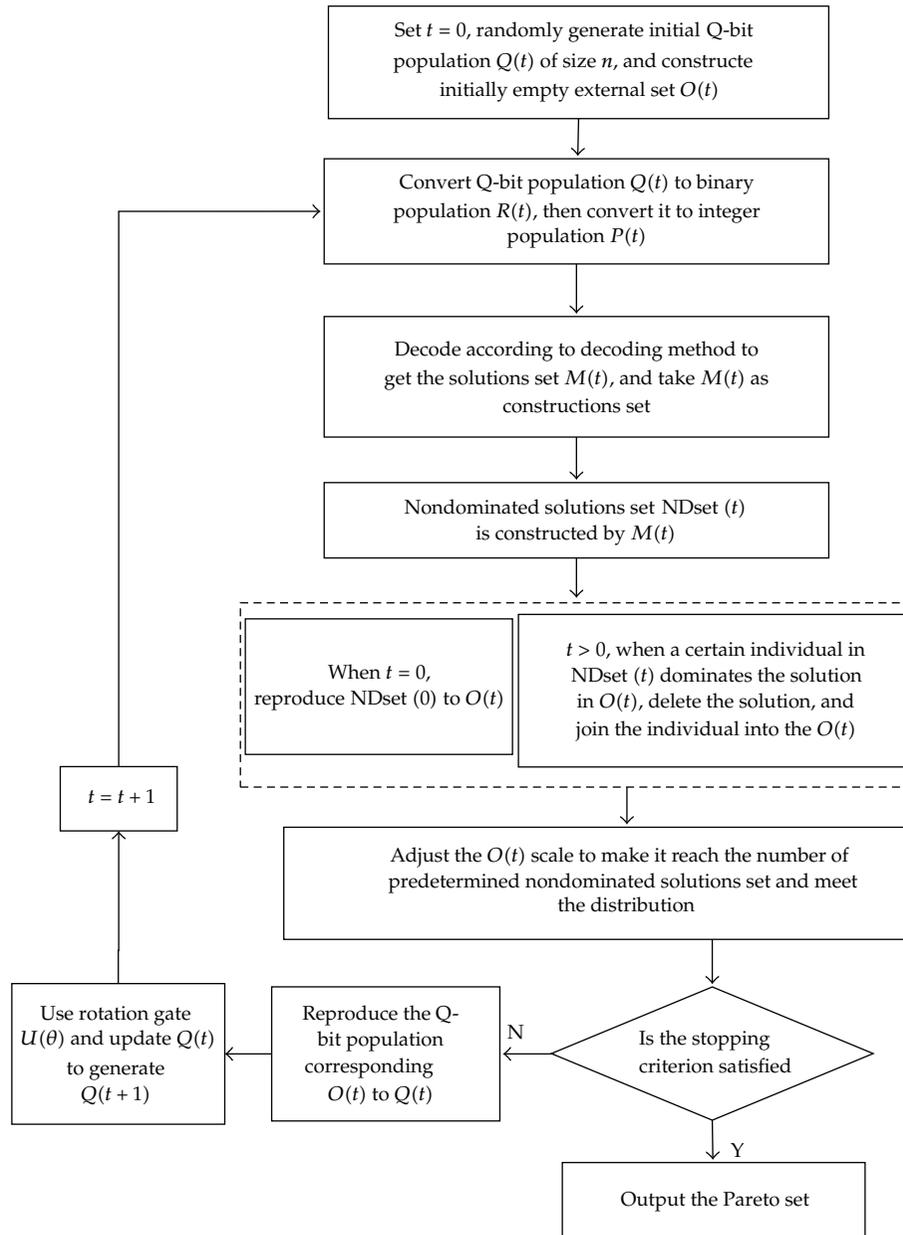


Figure 6: Flow chart of MOQEA.

## 4. Experiment Results and Comparisons

### 4.1. Experimental Data

There are few studies on multi-VRP taking into account customers' satisfaction. Among the studies that have taken into account customer satisfaction, most of them are evaluated using randomly generated test cases. Therefore, there is no standard test cases library. The

**Table 5:** The distance of each depot.

Depot no.	Coordinate/km		Service time	Demand/ <i>t</i>	Time windows
49	4.163	13.559	0	0	[0 1000]
50	21.387	17.105	0	0	[0 1000]
51	-36.118	49.097	0	0	[0 1000]
52	-31.201	0.235	0	0	[0 1000]

**Table 6:** Optimization results of pr01.

Pareto set	The solution of maximum satisfaction (472, 4532)	The solution of minimizing travel cost, fixed cost, and waiting cost (630, 3523)
VN	9	11
Route	52 21 36 31 27 6 24	52 43 21 14 13
	51 28 13 44 25	52 30 45 44 38
	51 5 39 17	50 6 40 48 47 17
	49 32 19 20 18 42	49 7 37 11 32 26 16 36 42
	51 4 11 41 26 1 29 22	52 12 41 15 2 4
	51 14 12 43 16 33	51 33 18 3
	50 3 45 8 40 10 37	51 1 35 20 39
	52 23 34 46 38 30 15 47	51 8 29 9 22
	49 7 9 35 48 2	51 5 34 24 23
		51 19 27 28 10 31 25
	49 46	

tests data used in this research is from the benchmark problems in the standard example library of MDVRPTW (multiple depot vehicle routing problem with time windows), and all examples can be downloaded from <http://neo.lcc.uma.es/radi-aeb/WebVRP/>.  $[E_i, L_i]$  is the time windows in initial data, as the tolerable time in this paper.  $[a_i, b_i]$  is the desirable service time, which can be computed using the following formula [27]:

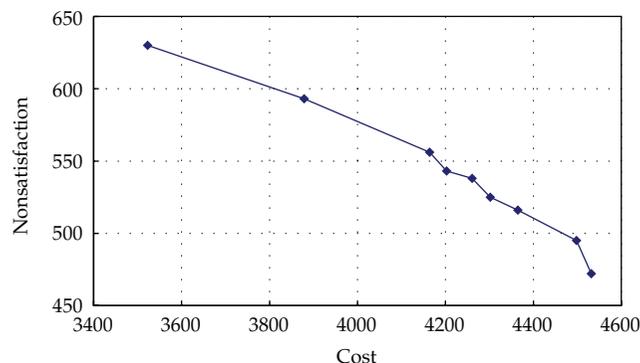
$$\begin{aligned}
 a_i &= E_i + \text{rand}() * 0.5 * (L_i - E_i), \\
 b_i &= L_i - \text{rand}() * 0.5 * (L_i - E_i).
 \end{aligned}
 \tag{4.1}$$

#### 4.2. Parameters Discussion of MOQEA

The parameters involved in the MOQEA include coefficient  $k$  in formula (2.1) and constant  $s$  in formula (3.7). The proposed MOQEA for different parameters was discussed and analyzed results are shown in Tables 2 and 3. VN is the vehicle numbers. CS is the customer nonsatisfaction. These tables only list the solution that the total cost is the least. From the tables we can see when  $s = 0.2$  and  $k = 0.05$ , the resultant vehicle number is the smallest.

**Table 7:** Comparisons of the MOQEA to the HMOEA in [34].

Algorithm	Project 1 MOQEA			Project 2 HMOEA in [34]		
	VN	CS	Cost	VN	CS	Cost
pr02 (96, 4)	14	0.355	3906	15	0.391	4005
pr07 (72, 6)	13	0.399	3331	13	0.402	3397

**Figure 7:** Pareto optimal solution set.

### 4.3. Simulation Experiments

All the programs in this research are developed using the JAVA language and run on a PC with Dual 2.8 GHz CPU and 1.0 GB of memory. A manufacturing company has 4 warehouses and provides goods to 48 vendors. The actual distribution process can be attributed to the open, capacity constraints, and multidepot VRP. The capacity is 20 t. The proposed MOQEA is used to solve this problem, and the distance and demand of each client and depot are shown in Tables 4 and 5. CN is the customer NO. ST is the service time. De is the demand. TW is the time windows.

The results obtained are shown in Table 6 and Figure 7. Specifically, the number of iterations is 2000, and the population size is 30. The coefficient  $k$  is set as 0.06. The constant  $s$  in formula (4.1) is 0.2. The Pareto optimal solution set is (630, 3523), (593, 3879), (556, 4164), (543, 4203), (538, 4261), (525, 4302), (516, 4365), (495, 4498), (472, 4532). The value of non-satisfaction magnified 1000 times. It can be seen from Figure 7 that the Pareto front that is close to the axis forms a more satisfactory solution set. Comparing the leftmost and the rightmost points, we can see that in this instance, it would be possible to decrease the total cost by 20% at the expense of an increase in the non-satisfaction which is about 25%.

### 4.4. Comparison and Discussion

In order to evaluate the performance of the algorithm, the proposed MOQEA is compared with the hybrid multiobjective evolutionary algorithm (HMOEA) developed in [34]. In the HMOEA, feasible individuals are constructed as the initial population by using the push-forward insertion heuristic (PFIH), and the GA is used to update these populations to obtain the new subpopulation and to improve the individuals of the subpopulation by the local

search method of  $\lambda$ -interchange with variable probability, then non-dominated solution set is constructed by using the Challenge Cup rule.

Table 7 shows the comparison of the results obtained from the two algorithms. Because the calculation result is a solution set, this table only lists the solutions in which the total cost is the least. VN is the vehicle numbers. CS is the customer non-satisfaction. From the table we can see for these two multidepot VRPs obtained from the proposed MOQEA the total cost is smaller than that from the HMOEA, and the customers' satisfaction obtained from the MOQEA is greater than that from the HMOEA.

## 5. Conclusions

This paper presents the modeling of vehicle scheduling problem that takes into account customer satisfaction and the development of the MVRPCS. Specifically, an improved trapezoidal fuzzy number is proposed to represent customers' satisfaction and the MOQEA for this problem is developed. The MOQEA can get multiple solutions, namely, the Pareto optimal solution set, according to his own expectations. These solutions will be used by the decision maker to choose the best one on the basis of different preferences on satisfaction maximization and travel costs minimization. In the MOQEA, the Challenge Cup rule is constructed for non-dominated solution set and a method for attaining keeping the variety of the solution set, is designed, based on self-adaptive grid. Simulation results and comparisons show that the MOQEA is an effective method. In our future work, we will focus on improving the algorithm and test it on other datasets.

## Acknowledgments

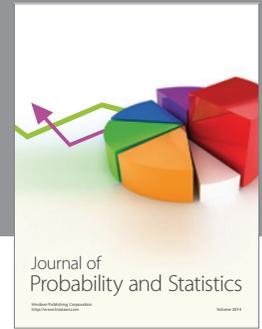
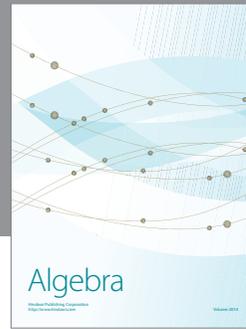
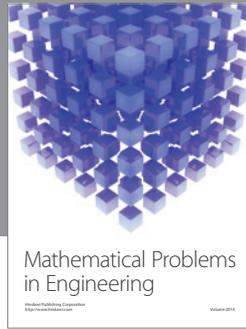
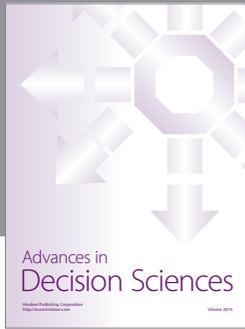
This paper is supported by the National Natural Science Foundation of China (Grant no. 60970021), the Postdoctoral Science Foundation of Zhejiang Province, and the Department of Education Foundation of Zhejiang Province (No. Y201225032). The authors are also most grateful for the constructive suggestions from anonymous reviewers which led to the making of several corrections and suggestions that have greatly aided in the presentation of this paper.

## References

- [1] P. Toth and D. Vigo, *The Vehicle Routing Problem*, vol. 9 of *SIAM Monographs on Discrete Mathematics and Applications*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2002.
- [2] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, pp. 80–91, 1959.
- [3] W. Huang and S. Chen, "Epidemic metapopulation model with traffic routing in scale-free networks," *Journal of Statistical Mechanics*, vol. 2011, no. 12, Article ID P12004, 19 pages, 2011.
- [4] M. Li, W. Zhao, and S. Chen, "MBm-based scalings of traffic propagated in internet," *Mathematical Problems in Engineering*, vol. 2011, Article ID 389803, 21 pages, 2011.
- [5] S. Y. Chen, H. Tong, Z. Wang, S. Liu, M. Li, and B. Zhang, "Improved generalized belief propagation for vision processing," *Mathematical Problems in Engineering*, vol. 2011, Article ID 416963, 12 pages, 2011.
- [6] C. H. Jiang, S. G. Dai, and Y. H. Hu, "Hybrid genetic algorithm for capacitated vehicle routing problem," *Computer Integrated Manufacturing Systems*, vol. 13, no. 10, pp. 2047–2052, 2007 (Chinese).
- [7] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 12, pp. 1985–2002, 2004.

- [8] P. Reiter and W. J. Gutjahr, "Exact hybrid algorithms for solving a bi-objective vehicle routing problem," *Central European Journal of Operations Research*, vol. 20, no. 1, pp. 19–43, 2012.
- [9] S. P. Anbuudayasankar, K. Ganesh, S. C. Lenny Koh, and Y. Ducq, "Modified savings heuristics and genetic algorithm for bi-objective vehicle routing problem with forced backhauls," *Expert Systems with Applications*, vol. 39, pp. 2296–2305, 2012.
- [10] S. C. Hong and Y. B. Park, "Heuristic for bi-objective vehicle routing with time window constraints," *International Journal of Production Economics*, vol. 62, no. 3, pp. 249–258, 1999.
- [11] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [12] A. Lim and F. Wang, "A smoothed dynamic tabu search embedded GRASP for m-VRPTW," in *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '04)*, pp. 704–708, November 2004.
- [13] K. C. Tan, Y. H. Chew, and L. H. Lee, "A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows," *Computational Optimization and Applications*, vol. 34, no. 1, pp. 115–151, 2006.
- [14] A. Garcia-Najera and J. A. Bullinaria, "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows," *Computers & Operations Research*, vol. 38, no. 1, pp. 287–300, 2011.
- [15] P. Badeau, F. Guertin, M. Gendreau, J. Y. Potvin, and E. Taillard, "A parallel tabu search heuristic for the vehicle routing problem with time windows," *Transportation Research C*, vol. 5, no. 2, pp. 109–122, 1997.
- [16] K. C. Tan, L. H. Lee, Q. L. Zhu, and K. Ou, "Heuristic methods for vehicle routing problem with time windows," *Artificial Intelligence in Engineering*, vol. 15, no. 3, pp. 281–295, 2001.
- [17] J. Berger and M. Barkaoui, "A parallel hybrid genetic algorithm for the vehicle routing problem with time windows," *Computers & Operations Research*, vol. 31, no. 12, pp. 2037–2053, 2004.
- [18] A. Le Bouthillier and T. G. Crainic, "A cooperative parallel meta-heuristic for the vehicle routing problem with time windows," *Computers and Operations Research*, vol. 32, no. 7, pp. 1685–1708, 2005.
- [19] G. B. Alvarenga, G. R. Mateus, and G. de Tomi, "A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows," *Computers and Operations Research*, vol. 34, no. 6, pp. 1561–1584, 2007.
- [20] R. Cheng and M. Gen, "Vehicle routing problem with fuzzy due-time using genetic algorithms," *Japanese Journal of Fuzzy Theory and Systems*, vol. 7, no. 5, pp. 1050–1061, 1995.
- [21] R. Cheng and M. Gen, "Fuzzy vehicle routing and scheduling problem using genetic algorithm," in *Genetic Algorithms and Soft Computing*, F. Herrera and J. Verdegay, Eds., pp. 683–709, Springer, 1996.
- [22] S. Wen, W. Zheng, J. Zhu, X. Li, and S. Chen, "Elman fuzzy adaptive control for obstacle avoidance of mobile robots using hybrid force/position incorporation," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 42, no. 4, pp. 603–608, 2012.
- [23] S. Y. Chen, J. Zhang, H. Zhang, N. M. Kwok, and Y. F. Li, "Intelligent lighting control for vision-based robotic manipulation," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3254–33263, 2012.
- [24] C. Cattani, "Shannon wavelets for the solution of integrodifferential equations," *Mathematical Problems in Engineering*, vol. 2010, Article ID 408418, 22 pages, 2010.
- [25] J. Y. Zhang, Y. H. Guo, and J. Li, "Research of multi-objective fuzzy vehicle scheduling problem based on satisfaction of customers," *Journal of the China Railway Society*, vol. 25, no. 2, pp. 15–17, 2003 (Chinese).
- [26] Y. J. Jia, *Optimal algorithm research of vehicle scheduling problem [Ph.D. thesis]*, Shanghai Jiaotong University, 2004.
- [27] B. Wu, *Particle swarm optimization for vehicle routing problem and its application [Ph.D. thesis]*, Zhejiang University of Technology, 2008.
- [28] J.-J. Lin, "A GA-based multi-objective decision making for optimal vehicle transportation," *Journal of Information Science and Engineering*, vol. 24, no. 1, pp. 237–260, 2008.
- [29] C. H. Wang and C. H. Li, "Optimization of an established multi-objective delivering problem by an improved hybrid algorithm," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4361–4367, 2011.
- [30] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 580–593, 2002.
- [31] M. Fonseca Carlos and J. Peter Fleming, "Genetic algorithm for multiobjective optimization: formulation, discussion and generalization," in *Proceeding of the 5th International Conference on Genetic Algorithm*, pp. 416–423, 1993.

- [32] S. Chen, Y. Zheng, C. Cattani, and W. Wang, "Modeling of biological intelligence for SCM system optimization," *Computational and Mathematical Methods in Medicine*, Article ID 769702, 10 pages, 2012.
- [33] S. Y. Chen and Y. F. Li, "Automatic Sensor Placement for Model-Based Robot Vision," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 34, no. 1, pp. 393–408, 2004.
- [34] Z. Jinhua, *Multi-Objective Evolutionary Algorithm and Its Application*, Science Press, Beijing, China, 2007.
- [35] J. Knowles and D. W. Corne, "The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation," in *Proceeding of the Congress of Evolutionary Computation*, pp. 98–105, 1999.
- [36] D. W. Corne, D. K. Joshua, and J. O. Martin, "The pareto envelope-based selection algorithm for multiobjective optimization," in *Proceeding of the Parallel Problem Solving from Nature VI Conference*, pp. 839–848, Springer, 2000.
- [37] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Proceedings of IEEE International Conference on Evolutionary Computation (ICEC '96)*, pp. 61–66, May 1996.
- [38] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA Journal on Computing*, vol. 6, pp. 154–160, 1994.
- [39] Z. Jingling, Z. Yanwei et al., "A hybrid quantum-inspired evolutionary algorithm for capacitated vehicle routing problem," in *Proceedings Advanced Intelligent Computing Theories and Applications*, vol. 5226, pp. 31–38, Springer, Berlin, Germany, 2008.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

