

Research Article

Multiobjective Vehicle Routing Problem with Route Balance Based on Genetic Algorithm

Wei Zhou,¹ Tingxin Song,¹ Fei He,² and Xi Liu¹

¹ Institute of Mechanical Engineering, Hubei University of Technology, Wuhan, Hubei 430068, China

² Nanjing University of Information Science and Technology, No. 219, Ningliu Road, Nanjing, Jiangsu 210094, China

Correspondence should be addressed to Fei He; hfeijust@163.com

Received 2 July 2013; Revised 1 October 2013; Accepted 13 November 2013

Academic Editor: Tinggui Chen

Copyright © 2013 Wei Zhou 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.

This study proposes a genetic algorithm to solve the biobjective vehicle routing problem with time windows simultaneously considering total distance and distance balance of active vehicle fleet. A new complex chromosome is used to present the active vehicle route. Through tournament selection, one-point crossover, and migrating mutation operator, the solution of the problem is solved. In experiment on Solomon's benchmark problems, considering the total distance and distance balance, the results are improved in all classes of problems. According to the experimental results, the suggested approach is sufficient and the average GA performance is good.

1. Introduction

The vehicle routing problem (VRP) is one of the most attractive topics in operation research, logistics, and supply chain management. VRP deals with minimizing the total cost of logistics systems. VRPs are well-known combinatorial optimization problems arising in transportation logistic that usually involve scheduling in constrained environments. In transportation management, there is a requirement to provide services from a supply point (depot) to various geographically dispersed points (customers) with significant economic implications. Because of VRP's important applications, many researchers have developed solution approaches for those problems.

Vehicle routing problem with time windows (VRPTW) is a variant of VRP with adding time windows constraints to that model. In VRPTW, a set of vehicles with limited capacity is to be routed from a central depot to a set of geographically dispersed customers with known demands and predefined time windows in order that fleet size of vehicles and total traveling distance are minimized and capacity and time windows constraints are satisfied. Due to its inherent complexities and usefulness in real life, the VRPTW continues to draw the attention of researchers and has become a well-known problem in network optimization,

so many authors developed different solution approaches based on exact and heuristics methods.

Many exact optimization approaches have been used to solve the VRPTW which is a well-known NP-hard problem [1]. An exact algorithm [2] of branch and cut techniques is presented. For its complexity, only small scale models can be solved [3] and such methods are inefficient in general [4]. By far Kohl's work [5] is one of the most efficient exact methods for solving 100-customer-size instance. As a result, many researchers have investigated the VRPTW using heuristics and metaheuristics approaches.

In recent years, approximate approaches are used in VRPTW instead of exact methods considering latter's intolerably high cost. Various heuristic methods may be found in literature in [6, 10]. These methods, including simulated annealing [7], and tabu-search [8], were proposed in literature. Genetic algorithm for VRPTW [6, 9–11] maybe the most widely used solution because of its efficiency. Thangiah [12] presents a hybrid using genetic algorithm and local search optimization. Different performance of genetic algorithm, tabu-search, and simulated annealing is studied in [6, 10].

These above pieces of literature focus on the single objective problems of the VRPTW by far. In fact multiobjective problems attract many researchers' attention since

the multiobjective is closer to real environments in these years. Some multiobject VRPs are formulated as a single function using weight parameters determined only experientially. Pareto-based approach is good to solve such problem since the managers can make their own decisions from the Paretooptimal output [13]. A specialized genetic operators and variable-length chromosome representation was used in VRPTW and produced very good result on Solomon's 56 benchmark problems [14].

Different objectives were classified in [15] according to different factors, that is, the tour, the resources, and the node activity. On tour, minimizing distance travelled (or time required) was the most common objective, while reducing the imbalance (or disparity) in the workload of vehicles was studied in [16]. Minimizing the number of vehicles is one of objectives related to resources. Ghoseiri and Ghannadpour [17] studied the multiobjective problem of minimizing the number of vehicles and the travelling distance. However, sometimes in real life the vehicles are often employed by the company and the cost is fixed. That means it is impossible to reduce such cost by reducing the numbers of vehicles. Whereas the total travelling distance is an important economic variable which is related to fuel consumption [18]. Furthermore, the workload balance of vehicles is another important variable because of management requirements.

This paper studies a biobjective VRPTW considering simultaneous minimization of the total traveling distance and workload imbalance of vehicles. Generally, the workload imbalance includes the distance imbalance and the load imbalance. However, in some real life environment, that is, fresh food delivery, the weight of good can be ignored because these orders are not heavy and make no influence on the workload cost. In other words, this paper will consider the multiobjective of the total travelling distance and the distance imbalance of active vehicle fleet. Section 2 describes the formulation of the VRPTW problem. Section 3 discusses the process of genetic algorithm to solve this problem. The experiments and results are analyzed in Section 4. Finally, Section 5 provides the conclusions to this work.

2. Model Formulation

The VRP problem was introduced by [19] and became one of the most widely analyzed NP-hard problems. The single objective VRPTW aims to determine which customers are visited by each vehicle and the route each vehicle follows to serve the assigned customers, while the distances travelled by the vehicles are minimized and the capacity and time windows constraints are satisfied. The VRPTW has been widely studied because it remains one of the most difficult problems in combinational optimization and has a considerable economic impact on all the logistic system [11], especially due to the importance of supply chain operations [20]. Some VRPTW problems were discussed with exact methods, such as Lagrangian relaxation-based methods, column generation, and dynamic programming. However, these exact methods often perform poorly for some intermediate and large problems. In this case, some heuristic and meta-heuristic methods have been proposed to solve these problems. And

the results show that these methods obtain feasible solutions in acceptable times.

2.1. Formulation for VRPTW. A nondirected complete graph $G(V, E)$ can be used to model the VRPTW. The vertices $V = \{1, \dots, N\}$ denote the depot and the customers, and edges $e \in E\{(i, j), i, j \in V\}$ correspond to the links between them.

The VRPTW can be formulated as follows.

Notation

a_j : is the earliest time for customer j to allow the service.

b_j : is the latest time for customer to allow the service.

C_{ij} : is the cost for travelling from node i to node j . It is considered as the distance or time required for travelling from node i to node j .

d_j : is the demand at customer j .

K : is the maximum number of vehicles that can be used.

N : is the number of customers plus the depot. The depot is denoted with number 1, and the customers are denoted as $(2, \dots, N)$.

Q : is the loading capacity of each vehicle.

S_{kj} : is the corresponding time at which vehicle k starts to service customer j .

L : is a given large value.

X_{ij}^k : is the decision variable. It is equal to 1 if vehicle k travels from node i to node j and is equal to 0 otherwise.

$$\text{Minimize TD} = \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N X_{ij}^k C_{ij} \quad (1)$$

subject to

$$X_{ii}^k = 0 \quad (\forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}), \quad (2)$$

$$X_{ij}^k \in \{0, 1\} \quad (\forall i, j \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}), \quad (3)$$

$$\sum_{k=1}^K \sum_{i=1, i \neq j}^N X_{ij}^k = 1 \quad (\forall j \in \{2, \dots, N\}), \quad (4)$$

$$\sum_{i=2}^N d_j \sum_{j=2, i \neq j}^N X_{ij}^k \leq Q \quad (\forall k \in \{1, \dots, K\}), \quad (5)$$

$$\sum_{k=1}^K \sum_{j=2}^N X_{1j}^k \leq K, \quad (6)$$

$$\sum_{j=2}^N X_{1j}^k - \sum_{j=2}^N X_{j1}^k = 0 \quad (\forall k \in \{1, \dots, K\}), \quad (7)$$

$$a_j \leq s_{kj} \leq b_j \quad (\forall i, j \in \{2, \dots, N\}, \forall k \in \{1, \dots, K\}), \quad (8)$$

$$s_{ki} + C_{ij} - L(1 - X_{ij}^k) \leq s_{kj} \quad (\forall i, j \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}). \quad (9)$$

Equation (1) is the objective function of the single objective problem. Equation (2) denotes that a vehicle must travel from one node to a different one. Equation (3) indicates that variable X_{ij}^k is equal to 1 if vehicle k goes from node i to node j and is equal to 0 otherwise. Equation (4) states that a customer is serviced only once by exactly one vehicle. By specifying the constraint of (5), it is taken into account that the load for a given vehicle k cannot exceed its capacity Q . Equation (6) specifies that there are up to K routes going out of the delivery depot. Equation (7) guarantees that each vehicle departs from and returns to the depot. Equation (8) ensures that time windows are observed. Given a large value, L , the inequality represented in (9) specifies that, if vehicle k is travelling from customer i to customer j , the vehicle cannot arrive at customer j before $s_{ki} + C_{ij}$. The variable s_{kj} corresponds to the time at which vehicle k starts to service customer j . If the vehicle k does not service j , s_{kj} is not calculated.

2.2. Multiobjective VRPTW with Distance Balance. The paper aims to solve the vehicle routing problem with hard time windows and route balance as a multiobject problem, where both the total travelling distance and route imbalance are minimized. The route balance often was related to the following factors:

- (1) balancing the number of customers visited by each active vehicle,
- (2) balancing the quantity or weight of the good delivered by each active vehicle, sometimes balancing the load rate (BLR), denoted as (10), where LV_i is the exact load of vehicle i and Q_i is the capacity of vehicle i
- (3) balancing the time required of the route,
- (4) balancing the waiting time required of the route,
- (5) balancing the delayed time of the route,
- (6) balancing the distance of the route travelled by active vehicles.

In this paper, we consider the imbalance of the distances of the route travelled, which is defined as (11). And (12) is the mean of all distances. In order to describe the balance more clearly, we use the balance factor to represent the degree in (13). Consider the following:

$$BL_d = \max \left(\sum_{i=1}^N \sum_{j=2}^N X_{ij}^k C_{ij} \right) - \min \left(\sum_{i=1}^N \sum_{j=2}^N X_{ij}^k C_{ij} \right) \quad (11)$$

($\forall k \in K$),

$$BL_m = \frac{1}{K} \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N X_{ij}^k C_{ij}, \quad (12)$$

$$BF = \frac{BL_d}{BL_m}. \quad (13)$$

Thus, from (1) and (11), the new multiobjective problem is defined as follows.

Minimize

$$f_{td+bl} = f \left(\left(\sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N X_{ij}^k C_{ij} \right), \left(\frac{BL_d}{BL_m} \right) \right), \quad (14)$$

while the constraints described above ((2)–(9)) are satisfied.

3. Multiobjective Genetic Algorithm

Various heuristic and meta-heuristic approaches have been proposed for solving the VRPTW. GA, compared with other heuristics [21–23], has been widely used to solve this problem, because of its efficiency and flexibility. Three main GA algorithmic operators, namely, selection, crossover, and mutation, can be configured in different ways, resulting in various GA combinations. Thangiah et al. [24] were the first to apply GA to VRPTW. During the past few years, numerous studies have been devoted to developing GAs for solving VRPTW. GA is an adaptive heuristic search method based on population genetics. The genetic algorithm represents the solution space using genetic coding of a feasible solution as a chromosome that defines an individual member of a population. While binary strings have been commonly used in the literature to code chromosomes, we adopt integer strings in the proposed GA, where each gene in a chromosome represents a customer (or a node). In a single objective problem of genetic algorithm, special fitness function is often defined but in MOP application of genetic algorithm the Pareto ranking scheme has often been used [25]. The Pareto ranking process tries to rank the solutions to find the nondominated solutions. Therefore, according to this process each solution gives a rank value in respect of different objective values that shows the quality of the solution compared to the other solutions. It is easily incorporated into the fitness evaluation process within a genetic algorithm by replacing the raw fitness scores with Pareto ranks. These ranks, to be defined later, stratify the population into preference categories. With it, lower ranks are preferable and the individuals within rank 1 are the best in the current population. The idea of Pareto ranking is to preserve the independence of individual objectives. This is done by treating the current candidate solutions as stratified sets or ranks of possible solutions. The individuals in each rank set represent solutions that are in some sense incomparable with one another. Pareto ranking will only differentiate individuals that are clearly superior to others in all dimensions of the problem. This contrasts with a pure genetic algorithms attempt to assign a single fitness score to a MOP, perhaps as a weighted sum. Doing so essentially recasts the MOP as a single objective problem. The difficulty with this is that the weighted sum necessitates the introduction of bias into both search performance and quality of solutions obtained. For many MOP's, finding an effective weighting for the multiple dimensions is difficult and ad hoc and often results in unsatisfactory performance and solutions.

3.1. Chromosome Representation. This paper uses a complex two-part chromosome to represent the solution of VRPTW.

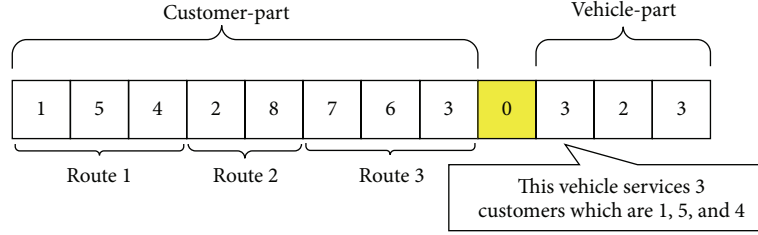


FIGURE 1: An example of a chromosome.

The chromosome is separated into two parts by a zero number decorated by yellow. The first part of the chromosome is a chain of integers and each of the integers represents a customer. We also call this part customer-part. The customers on it are separated to several routes, each of them representing a sequence of delivers that must be covered by a vehicle. The second part of the chromosome contains vehicles information. We also can call this part vehicle-part. In the vehicle-part the quantity of genes equals the quantity of routes in the customer-part. The number on each of the genes represents the length of its corresponding route. The sum of these numbers in vehicle-part must be equal to the quantity of customers. For example, Figure 1 shows a representation of a possible solution with 8 customers and 3 vehicles. There are 3 genes in vehicle-part which means that those 8 customers are separated to 3 routes. The 3 on the first gene in vehicle-part represents route 1 that services 3 customers which are 1, 5, and 4. The 2 on the second gene means that route 2 services 2 and 8. The 3 on the third gene means that route 3 services 7, 6, and 3.

This design is different from the classical approach, in which the route information is mixed with the customer sequences in a single chromosome. Storing the route information and customer sequences separately can represent the solution more clearly and facilitate the implementation of the algorithm, but its effect on the computational efficiency would not be significant. Without loss of generality, we consider the following implementations of the three operators.

3.2. Selection. There are several commonly used selection operators used in GA selection process. Roulette wheel selection (RWS) is to stochastically select from one generation to create the basis of the next generation. RWS enables the fittest individuals to have a greater chance of survival than weaker ones. This replicates nature in that fitter individuals will tend to have a better probability of survival and will go forward to form the mating pool for the next generation. Weaker individuals are not without a chance. In nature such individuals may have genetic coding that may be proven to be useful for future generations. Unlike RWS, uniform selection (US) assigns the same probability to each chromosome of the population. The US operation proceeds at random and is easy to implement. However, it has been criticized for lacking the spirit of natural evolution compared with RWS. Tournament selection (TS) is the most commonly used operation besides RWS. The TS operator involves running several “tournaments” among a set of chromosomes chosen

at random. The one with the largest fitness is selected for crossover in a pair of chromosomes. The tournament size can be used to adjust the selection pressure. If the tournament size is larger, weaker individuals will have a smaller chance to be selected. This process is repeated until the mating pool is full. Since some experiments indicate that the TS operator outperforms the RWS and US, we choose TS as the selection operator. A possible explanation is that TW always selects the best set of individuals to crossover, whereas RWS and TS are probabilistic and hence some good individuals may be lost in the evolutionary process [26].

3.3. Crossover. One-point crossover operator evolves selecting one point randomly which divides a parent into two parts. Each of these points is selected with equal probability. For example, the crossover point is selected at the third gene of parent 1 randomly. The offspring inherits the left side from parent 1 and other genes are inherited from parent 2. Another offspring is produced by exchanging the roles of two parents. Figure 2 illustrates the process of one-point crossover [27]. The cycle crossover can produce offspring through a cycle which is a sequent of the position of the first parents. The partially mapped crossover operator produces the offspring by randomly selecting two crossover points [28]. The linear-order crossover also selects two crossover points from one parent and produces a new offspring with another parent [29]. Some experiments illustrates that the one-point crossover is more efficient than the other tree operators [26]. Thus this paper chooses the one-point crossover operator.

3.4. Mutation. In mutation process, there are also several mutation operators in different literatures. Some of them are very complex. However, some different mutation operators were experimented that they did not make significant difference in GA efficiency. A possible explanation to that maybe the mutation rate is always very small, typically between 0.01 and 0.1. A migrating mutation is adopted to produce heterogeneous chromosomes in the pool to avoid early convergence of the algorithm. This mutation method is to randomly select a chromosome from the pool and then randomly choose a customer from one route. Then the selected customer is tried to be inserted into a new route. If the insertion results produce a feasible route, this mutation operator succeeds. Otherwise, this process is repeated until a feasible solution is achieved. Figure 3 illustrates the migrating mutation process.

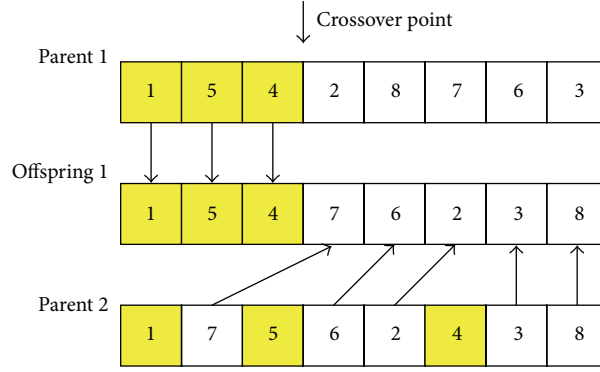


FIGURE 2: One-point crossover operator.

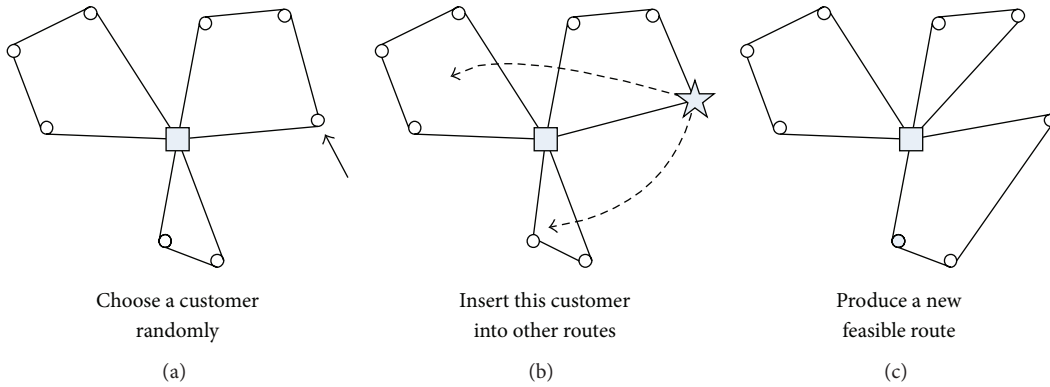


FIGURE 3: Migrating mutation process.

4. Experimental Results and Comparisons

The Solomon's problems consist of 56 data sets, which have been extensively used for benchmarking different algorithms for VRPTW in literature over the years, since they represent relatively well different kinds of routing scenarios [14]. These problems are different in fleet size, vehicle capacity, traveling time of vehicles, and so on. The customers' details include the sequence of customer index, location in x and y coordinates, the demand for load, the ready time, due date, and the service time required. All the test problems consist of 100 customers and these customers are generally adopted as the problem size for performance comparisons in VRPTW. The traveling time between customers is equal to the corresponding Euclidean distance. This problem data is clustered into six classes, named C1, C2, R1, R2, RC1, and RC2, respectively. And different categories indicate the different customer distribution and different time windows constraints. According to the customer location, problem category C has all customers clustered in groups, problem category R has all customers located randomly, and problem category RC has a mixture of random and clustered customers. In other words, customers are located closer to each other in problem category C than in R and customers in RC are in the middle. That also means that the R is more difficult to be solved. Moreover, the time windows of category 1 (C1, R1, and RC1) is smaller than that of category 2 (C2, R2, and RC2). Smaller time windows mean

that some candidate solutions are more likely to become unfeasible after a small change in the sequence of visited customers. Furthermore, for category 1, the time window is also narrow for the depot, which means that only a few customers can be served by one vehicle.

This section describes computational experiments carried out to investigate the efficiency of the proposed GA. The algorithm is coded in JAVA and run on a PC with 2.53 GHz CPU and 2000 MB memory. The standard Solomon's VRPTW benchmark problem instance is used as experimental data [30]. Empirically the computation is based on the following parameters:

Population size = 100,
 Generation number = 500,
 Crossover rate = 0.9,
 Mutation rate = 0.2.

To illustrate the influence of different models with routing balance, we have employed the best solution of the benchmark problems in Table 1, 2, and 3, for the data groups C, R, and RC, respectively. Each table includes the following:

in the first column, the benchmark problem instance according to [30],

in the second column, the best known solution for that problem in literatures,

TABLE 1: Results of C instances.

No.	Instance	Best known		Suggestive algorithm			
		Total distance	Distance	Single objective Difference (%)	Balance	Biobjective Distance	Balance
1	C101	828.94	828.94	0.00	42.2%	839	10.3%
2	C102	828.94	828.94	0.00	42.2%	838	4.2%
3	C103	828.06	828.06	0.00	42.3%	1116	1.8%
4	C104	824.78	824.78	0.00	41.2%	1253	2.9%
5	C105	828.94	828.95	0.01	42.2%	969	4.2%
6	C106	828.94	828.95	0.01	42.2%	1098	6.0%
7	C107	828.94	828.95	0.01	42.2%	842	3.9%
8	C108	828.94	828.95	0.01	42.2%	1169	5.8%
9	C109	828.94	828.95	0.01	42.2%	1198	8.9%
10	C201	591.56	591.58	0.01	11.2%	565	2.7%
11	C202	591.56	591.58	0.01	11.2%	792	0.8%
12	C203	591.17	591.18	0.01	10.7%	716	0.8%
13	C204	590.60	590.62	0.01	11.2%	759	2.4%
14	C205	588.16	588.18	0.01	10.7%	784	1.7%
15	C206	588.49	588.51	0.01	10.7%	815	2.2%
16	C207	588.29	588.30	0.01	10.7%	693	3.9%
17	C208	588.32	588.32	0.00	10.7%	839	4.5%

TABLE 2: Results of R instances.

No.	Instance	Best known		Suggestive algorithm			
		Total distance	Distance	Single objective Difference (%)	Balance	Biobjective Distance	Balance
1	R101	1645.79	1650.8	0.30	64.6%	1673	8.1%
2	R102	1486.12	1486.12	0.00	62.9%	1746	4.5%
3	R103	1292.68	1292.68	0.00	49.3%	1471	5.1%
4	R104	1007.24	1007.24	0.00	4.5%	1141	3.2%
5	R105	1377.11	1377.11	0.00	50.8%	1332	4.7%
6	R106	1251.98	1251.98	0.00	51.8%	1505	4.5%
7	R107	1104.66	1104.66	0.00	4.5%	1303	1.5%
8	R108	960.88	960.88	0.00	0.9%	1164	0.5%
9	R109	1194.73	1194.73	0.00	30.4%	1518	3.9%
10	R110	1118.59	1118.59	0.00	5.4%	1090	3.7%
11	R111	1096.72	1096.72	0.00	7.3%	1335	1.7%
12	R112	982.14	987.24	0.52	3.7%	992	2.9%
13	R201	1252.37	1252.37	0.00	22.0%	1282	2.5%
14	R202	1191.70	1191.70	0.00	14.9%	1146	2.4%
15	R203	939.54	939.54	0.00	29.1%	1041	0.8%
16	R204	825.52	832.14	0.80	0.7%	847	3.2%
17	R205	994.42	994.42	0.00	4.2%	1138	1.1%
18	R206	906.14	906.14	0.00	7.6%	1054	4.7%
19	R207	890.61	896.88	0.70	0.4%	895	3.4%
20	R208	726.75	726.75	0.00	8.0%	891	0.1%
21	R209	909.16	909.16	0.00	20.8%	1171	5.6%
22	R210	939.34	939.37	0.00	7.7%	1046	2.4%
23	R211	892.71	904.78	1.33	0.7%	925	9.0%

TABLE 3: Results of RC instances.

No.	Instance	Best known		Suggestive algorithm			
		Total distance	Distance	Single objective Difference (%)	Balance	Biobjective Distance	Balance
1	RC101	1696.94	1696.95	0.00	51.2%	1963	4.5%
2	RC102	1554.75	1554.75	0.00	21.6%	1894	2.6%
3	RC103	1261.67	1261.67	0.00	28.8%	1669	5.3%
4	RC104	1135.48	1135.48	0.00	21.1%	1447	1.0%
5	RC105	1629.44	1629.44	0.00	30.3%	1956	3.5%
6	RC106	1424.73	1424.73	0.00	17.8%	1858	1.7%
7	RC107	1230.48	1230.48	0.00	22.3%	1764	2.9%
8	RC108	1139.82	1139.82	0.00	11.4%	1545	1.6%
9	RC201	1406.91	1406.94	0.00	16.8%	1646	4.3%
10	RC202	1365.65	1365.65	0.00	8.8%	1439	1.9%
11	RC203	1049.62	1058.33	0.82	10.0%	1243	0.4%
12	RC204	789.41	798.46	1.13	16.3%	936	9.6%
13	RC205	1297.19	1297.65	0.04	14.8%	1485	3.4%
14	RC206	1146.32	1146.32	0.00	17.3%	1187	5.1%
15	RC207	1061.14	1061.14	0.00	15.0%	1327	3.9%
16	RC208	828.14	828.71	0.07	25.0%	1046	8.1%

TABLE 4: Compared results of all instances.

No.	Instance class	Best known			Suggestive approach			Improved
		B_{\min}	B_{\max}	B_{avg}	B_{\min}	B_{\max}	B_{avg}	B_{dec}
1	C1	42.3%	41.2%	42.1%	1.8%	8.9%	5.3%	87.4%
2	C2	11.2%	10.7%	10.9%	0.8%	5.5%	2.4%	78.3%
3	R1	64.6%	0.9%	28.0%	0.5%	8.1%	3.7%	86.8%
4	R2	29.1%	0.4%	10.6%	0.1%	9.1%	3.2%	69.7%
5	RC1	51.2%	11.4%	25.6%	1.0%	5.3%	2.9%	88.7%
6	RC2	25.0%	8.0%	15.5%	0.42%	8.08%	4.6%	73.9%

in the third column, the best solution found by the algorithm of this paper,

in the fourth column, the difference by percent between the best known and the best found,

in the third column from last, the balance rate of single objective search,

the second column from last and the final column, the new distance value and the balance rate of biobjective algorithm of this paper.

From the results of Tables 1 to 3 in single objective approach, it is found that the balance rate of C1, R1, and RC1 is bigger than that of C2, R2, and RC2 when the balance is not considered. In other words, category 1 with wider time windows has more space to improve the balance than category 2 with smaller time windows. For example, the balance rate in C101 is 42.2% but in C201 is 11.2%.

After considering the distance balance, the biobjective solution data illustrated much improvement on the balance rate without much influence on the distance cost. For example, in C101 when the balance decreases from 42.2% to 10.3%, the distance only reduces by about 10 (839 minus 828.94). Not only the instances with wide time windows get

a great improvement, but also the ones with the narrow time windows reduce the balance rates.

The comparative data shows that suggestive algorithm of this paper reaches better route balance without significant deterioration of the VRPTW solution, in terms of the active vehicle fleet. From Table 4, the last column (B_{dec}) presents the degree of balance improved.

5. Conclusion

The problem discussed in this paper is of significant practical importance in cases where employee's labor balance is key motivation of vehicle routing. In some situation, the weight is not very important compared with the distances of the active vehicle fleet when considering labor balance.

This paper proposed a genetic algorithm to solve the biobjective vehicle routing problem with time windows simultaneously considering total distance and distance balance of active vehicle fleet. We used a new complex chromosome to present the active vehicle route. We choose Tournament selection, one-point crossover, and migrating mutation operator to solve this GA. After iterator operation, the solution of the problem was solved. In experiment on Solomon's

benchmark problems, we found that this objective is close to best known value in literatures, even though it was not designed for the single objective problems. Considering the distance balance, those instances are imbalanced and have much space to improve. From the results, the distance balance was improved in all classes of problems in the biobjective problem. According to the produced results, the suggested approach was sufficient and the average GA performance was adequate.

Acknowledgments

The authors thank the anonymous reviewers for their useful suggestions and comments. This work was supported by the National Natural Science Funds of China no. 51105157.

References

- [1] M. W. P. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operations Research*, vol. 4, no. 1, pp. 285–305, 1985.
- [2] N. R. Achuthan, L. Caccetta, and S. P. Hill, "An improved branch-and-cut algorithm for the capacitated vehicle routing problem," *Transportation Science*, vol. 37, no. 2, pp. 153–169, 2003.
- [3] A. W. J. Kolen, A. H. G. Rinnooy Kan, and H. W. J. M. Trienekens, "Vehicle routing and scheduling with time windows," *Operations Research*, vol. 35, no. 2, pp. 266–273, 1987.
- [4] M. Desrochers, J. Desrosiers, and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Operations Research*, vol. 40, no. 2, pp. 342–354, 1992.
- [5] N. Kohl, *Exact methods for time constrained routing and related scheduling problems [Ph.D. thesis]*, Department of Mathematical Modeling, Technical University of Denmark, 1995.
- [6] 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.
- [7] W. Chiang and R. A. Russell, "Simulated annealing metaheuristics for the vehicle routing problem with time windows," *Annals of Operations Research*, vol. 63, pp. 3–27, 1996.
- [8] E. D. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Potvin, "A tabu search heuristic for the vehicle routing problem with soft time windows," *Transportation Science*, vol. 31, no. 2, pp. 170–186, 1997.
- [9] S. Thangiah, "Vehicle routing with time windows using genetic algorithms," in *Applications Handbook of Genetic Algorithms: New Frontiers, Volume II*, pp. 253–277, CRC Press, Boca Raton, Fla, USA, 1995.
- [10] K. C. Tan, L. H. Lee, and K. Ou, "Hybrid genetic algorithms in solving vehicle routing problems with time window constraints," *Asia-Pacific Journal of Operational Research*, vol. 18, no. 1, pp. 121–130, 2001.
- [11] 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.
- [12] S. R. Thangiah, "A hybrid genetic algorithms, simulated annealing and tabu search heuristic for vehicle routing problems with time windows," in *Practical Handbook of Genetic Algorithms Complex Structures, Volume 3. L. Chambers*, pp. 374–381, CRC Press, 1999.
- [13] D. E. Goldberg, *Genetic Algorithms Insearch, Optimization and Machine Learning*, New York, NY, USA, Addison-Wesley edition, 1989.
- [14] 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.
- [15] N. Jozefowicz, F. Semet, and E. Talbi, "Multi-objective vehicle routing problems," *European Journal of Operational Research*, vol. 189, no. 2, pp. 293–309, 2008.
- [16] N. Jozefowicz, F. Semet, and E. Talbi, "Target aiming Pareto search and its application to the vehicle routing problem with route balancing," *Journal of Heuristics*, vol. 13, no. 5, pp. 455–469, 2007.
- [17] K. Ghoseiri and S. F. Ghannadpour, "Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm," *Applied Soft Computing Journal*, vol. 10, no. 4, pp. 1096–1107, 2010.
- [18] Y. Kuo, "Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem," *Computers and Industrial Engineering*, vol. 59, no. 1, pp. 157–165, 2010.
- [19] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, pp. 80–91, 1959.
- [20] M. A. Figliozzi, "An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows," *Transportation Research C*, vol. 18, no. 5, pp. 668–679, 2010.
- [21] K. Pang, "An adaptive parallel route construction heuristic for the vehicle routing problem with time windows constraints," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11939–11946, 2011.
- [22] R. Baños, J. Ortega, C. Gil, A. L. Marquez, and F. de Toro, "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows," *Computers and Industrial Engineering*, vol. 65, no. 2, pp. 286–296, 2013.
- [23] J. O. Baños, C. Gil, A. Fernández, and F. de Toro, "A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows," *Expert Systems with Applications*, vol. 40, pp. 1696–1707, 2013.
- [24] S. R. Thangiah, K. E. Nygard, and P. L. Juell, "GIDEON: a genetic algorithm system for vehicle routing with time windows," in *Proceedings of the 7th IEEE Conference on Artificial Intelligence Applications*, pp. 322–328, Miami Beach, Fla, USA, February 1991.
- [25] B. Ombuki, B. J. Ross, and F. Hanshar, "Multi-objective genetic algorithm for vehicle routing problem with time windows," *Applied Intelligence*, vol. 24, pp. 17–33, 2006.
- [26] C. C. Lu and V. F. Yu, "Data envelopment analysis for evaluating the efficiency of genetic algorithms on solving the vehicle routing problem with soft time windows," *Computers & Industrial Engineering*, vol. 63, no. 2, pp. 520–529, 2012.
- [27] T. Murata and H. Ishibuchi, "Performance evaluation of genetic algorithms for flowshop scheduling problems," in *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, pp. 812–817, Orlando, Fla, USA, June 1994.
- [28] I. Oliver, D. Smith, and J. Holland, "A study of permutation crossover operators on the traveling salesman problem," in *Proceedings of the 2nd International Conference on Genetic Algorithms and Their Applications*, Lawrence Erlbaum Associates, Cambridge, Mass, USA, 1987.

- [29] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York, NY, USA, 2000.
- [30] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.

