Differential Evolution Algorithm Based on Sort Mutation Operation for Vehicle Routing Problem with Time Windows

Xiaoyu Song, Yan Wu* and Ming Zhao
Control Engineering Faculty, Shenyang Jianzhu University, Shenyang, Liaoning, 110168, China

Abstract. In order to solve vehicle routing problem with time windows, a new sort-based differential evolution algorithm is proposed in this paper. First, Pareto dominates the idea to perform dual-objective optimization and natural number encoding to reflect the order of solutions; at the same time, in order to balance the global exploration and local development capabilities of the algorithm, a new sort-based mutation operation DE /sort/1 is proposed. And adopts an innovative dual-strategy evolutionary method based on Pareto frontier selection, and chooses different crossover operations during the crossover process. Finally, this paper uses the standard Solomon test set. Experimental results show that the algorithm is an effective method for solving the vehicle routing problem with time windows.

Keywords: Vehicle routing problem with time windows; Sort-Based differential evolution algorithm; Two-objective optimization problem.

1. Introduction
In recent years, the logistics industry has developed rapidly, and research on reducing logistics costs has received more and more attention. Therefore, the problem of vehicle routing has become a hot issue. VRP was first proposed by Dantzig and Ramser [1] in 1959. Solomon [2] first introduced the time window to the vehicle routing problem in 1979, and proposed the vehicle routing problem with the time window. Because the vehicle routing problem with time windows is an NP-hard problem [3], it is difficult to solve it with an accurate algorithm, so the research mainly uses a heuristic algorithm. Among them, the differential evolution algorithm is an excellent group-based heuristic algorithm. DE retains the global search strategy of the population, using real number coding, differential mutation operation, and parent-child competition selection strategy. However, DE itself has the disadvantage of low solution quality. Therefore, how to make DE have good global exploration ability and local development ability when solving multi-objective problems has become a research focus. The sort-based mutation operation can solve this problem well, and the better individuals in the population are given greater probability as the participation vector in the mutation formula [4].
The vehicle routing problem with time windows considered in this paper can be described as: on the premise of a single distribution center, with the goal of using the least number of vehicles and driving distance to provide distribution services to multiple customer points. During the distribution process, the vehicle must meet the on-board capacity not exceeding its limited nuclear capacity, and the service must be completed within the customer's acceptance time, and each customer point can only be served once.

In this regard, this paper uses Pareto domination to perform bi-objective optimization, proposes an innovative algorithm, uses Pareto frontiers to determine mutation strategies, combines ranking and random mutation-based operations, and combines binomial and exponential crossing. Experiments on
Solomon examples of different scales show that the hybrid differential evolution algorithm proposed in this paper is an effective method for solving vehicle routing problems with time windows.

2. Hybrid Differential Evolution Algorithm Based on Sort Mutation Operation for VRPTW

2.1. Chromosome Encoding and Decoding Methods.

Because VRPTW is an order-based optimization problem in VRP, and binary coding cannot express this order relationship well, natural number coding is generally used in the solution [5]. The natural number coding method used in this article is: use the natural numbers from 1 to n to represent customers, and arrange the n natural numbers in full. One arrangement represents a solution to the problem, and the order of natural numbers between gene positions represents the order of vehicles reaching the customer point. Because there is no genetic position of the path demarcation point in the chromosome, each natural number of customers in the solution is sequentially included in the vehicle path in accordance with the constraint conditions, that is, the vehicle capacity and the time windows. When a customer point violates the time windows or load capacity constraint, reopen a new path and incorporate this point into the path.

2.2. Pareto Domination

Because the vehicle routing problem with time windows is a multi-objective optimization problem, common processing methods include multi-objective weighting and Pareto-based domination. The multi-objective weighting method is difficult to determine the exact weight coefficient, resulting in the loss of the optimal boundary solution. Compared with this, Pareto domination selection strategy has more advantages. Among them, the concept of congestion distance is adopted in this paper, and it has the advantages of better ability and lower complexity to preserve good individuals. Among them, individuals in different non-dominated layers, non-dominated layers. The smaller the rank number is, the higher the individual fitness value is; the larger the crowded distance is, the higher the fitness value is for individuals in the same non-dominated layer.

2.3. Mutation Process

2.3.1. Sort Operation. It can be known from the standard DE algorithm that the mutated individuals are the result of the interaction between 3 target individuals randomly selected in the target population. \( X_i^G = [x_{i,1}^G, x_{i,2}^G, ..., x_{i,n}^G] \), \( V_i^G = [v_{i,1}^G, v_{i,2}^G, ..., v_{i,n}^G] \), \( U_i^G = [u_{i,1}^G, u_{i,2}^G, ..., u_{i,n}^G] \) are the i-th individual of the G-generation target population, the mutant population, and the experimental population, respectively. Because the mutation operation of the differential evolution algorithm is to randomly select individuals to optimize the solution, there is the disadvantage of slower convergence speed. In order to solve this problem, this paper uses a ranking-based mutation process, that is, by assigning a greater probability of improvement to the ranking of good individuals to enhance their local exploration capabilities, the specific operations are as follows:

In this article, Pareto domination is used to stratify the target individuals, each individual is assigned a hierarchical serial number \( R_i^G \). At this time, the value of the hierarchical serial number of the excellent individual is small and needs to be processed. Then, the hierarchical individual is assigned a serial number using the formula \( F_i^G \).

\[
F_i^G = R_i^G - R_{\text{max}}^G + 1
\] (1)

Among them, \( R_{\text{max}}^G \) is the largest number of contemporary levels. After treatment, individuals with higher levels in the population at this time have obtained larger serial numbers \( F_i^G \). Next, the selection probability of each individual is calculated according to formula (2). In this way, all individuals in the population have completed the selection of the selection probability. Individuals
with the same selection probability at each layer and higher-level individuals have a greater selection probability.

\[ P_i^G = \frac{F_i^G}{\sum_{k=1}^{G_{max}} k}; k = 1, 2, ..., R_{max} \]  

(2)

Next, mutate all individuals in the population: For the selection of \( G_{r1}X \) in the mutation formula, first randomly select an individual different from \( G_i^X \) from the population, and then generate a \((0,1)\) random number. If the probability value \( P_{r1}^G \) is not less than a random value, the selection is successful, otherwise the selection is made again; the selection of \( G_{r2}X \) and \( G_{r3}X \) is the same as \( G_{r1}X \), and it is guaranteed that \( G_{r2}X, G_{r3}X \) and \( G_{r1}X \) are different from each other.

2.3.2. Mutation Operation. This paper proposes a new mutation strategy, \( DE/SORT/1 \), which is redefined as:

\[ V_i^G = X_{r1}^G \oplus [F \otimes (X_{SORT}^G \oplus X_{r2}^G)] \]  

(3)

Among them, \( r1, r2 \) are mutually different integers in the interval \([1, n]\); \( X_{SORT}^G \) is an individual selection operation based on ranking; \( F \) is the scaling factor, and \( F \in [0,1] \).

The calculation process of \( \otimes \) is taken as an example as follows:

\[ F \otimes (X_{r1}^G \oplus X_{r2}^G) = \begin{cases} X_{r1}^G \oplus X_{r2}^G, & \text{rand()} < F \\ X_{r1}^G, & \text{rand()} \geq F \end{cases} \]  

(4)

If \( \text{rand}() < F \), \( \otimes \) calculation is performed. The implementation process is as follows: Take \( X_{r1}^G \oplus X_{r2}^G \) as an example, randomly select a starting point \( r \) and a random array length from \( X_{r2}^G \), place it in front of \( X_{r1}^G \), and then delete duplicate genes in \( X_{r1}^G \).

It can be seen that \( DE/RAND/1 \) can be redefined as:

\[ V_i^G = X_{r1}^G \oplus [F \otimes (X_{r2}^G \oplus X_{r3}^G)] \]  

(5)

The specific operation is similar to the \( DE/SORT/1 \) operation.

2.3.3. Co-evolution of Double Mutation Strategies. This article adopts an innovative method that uses the number of individuals in the Pareto front to determine the mutation strategy, that is, to use the \( DE/SORT/1 \) mutation operation on the excellent solution on the Pareto front to better retain the excellent genes and improve the ability of algorithm development; Using the \( DE/RAND/1 \) mutation operation, the disturbance is greater, the algorithm exploration ability is improved, and the optimal solution is easier to access. However, as the algorithm is in progress, there may be too many and too few Pareto fronts. In order to further balance the algorithm’s exploration and development capabilities, the number of individuals in the Pareto front should be between 20% and 80% of the population. Accepted, rationalize the situation where the number of individuals at the Pareto front is not in this range:

For the case where the number of individuals at the Pareto front is less than 20% of the number of populations, a 0.2\( NP \) excellent solution is selected. First determine that 0.2\( NP \) individuals fall in the first few layers: If it falls in the second layer, randomly select a solution from the second layer and form 0.2\( NP \) excellent solutions together with the Pareto front to perform \( DE/SORT/1 \) operation and the remaining individuals to perform \( DE/RAND/1 \) operation; If it falls in levels 3 and 3, select the
solutions of all levels before this level as excellent solutions, and then randomly select solutions in this level and the previously selected solutions to form 0.2NP excellent solutions, and perform DE/SORT/1 operation, the remaining individuals perform DE/RAND/1 operation.

For the case where the number of individuals at the Pareto front is greater than 80% of the number of populations, a 0.8NP excellent solution is selected. In the Pareto front, 0.8NP solutions are randomly selected for DE/SORT/1 operation, and the remaining individuals are subjected to DE/RAND/1 operation.

2.4. Crossover Process
In order to balance the performance of algorithm exploration and development, this paper adopts different crossing methods for different mutation operations, adopts an improved binomial crossing method for DE/SORT/1, and uses an improved exponential crossing for DE/RAND/1. Binomial intersection: In order to make the mutated superior individuals produce less disturbances and legalize operations, crossovers are used to perform crossovers. The specific process: \( G_{i}^{X} \) will be assigned to \( U_{i}^{G} \), and then \( U_{i}^{G} \) will be improved. When \( \text{rand()} < CR \), the gene \( V_{i,j}^{G} \) in the current dimension of \( V_{i}^{G} \) will be replaced by \( X_{i,j}^{G} \) in the corresponding position, locate the corresponding \( j^{*} \) in the position of \( V_{i,j}^{G} \) in \( U_{i}^{G} \), and replace \( X_{i,j}^{G} \) with \( X_{i,j}^{G} \), in order to less disturb the excellent solution and better retain the excellent gene.

Exponential crossover: In order to make the inferior solution have a larger evolution radius to jump out of the local optimum and shorten the convergence speed, an improved exponential crossover method is adopted:

\[
U_{i}^{G} = CR \otimes (V_{i}^{G} \oplus X_{i}^{G}) = \begin{cases} V_{i}^{G} \oplus X_{i}^{G}, & \text{rand()} < CR \\ V_{i}^{G}, & \text{rand()} \geq CR \end{cases}
\]

(6)

Where \( \otimes \) and \( \oplus \) are calculated as above.

2.5. Selection Process
Since this paper uses the Pareto algorithm to solve the multi-objective problem, the dominance relationship between \( X_{i}^{G} \) and \( U_{i}^{G} \) is used for selection, if \( U_{i}^{G} \) dominates \( X_{i}^{G} \), choose \( U_{i}^{G} \), otherwise, choose \( X_{i}^{G} \).

3. Experimental Analysis
In this paper, the well-known Solomon example is used to test the effectiveness of the algorithm. This article uses VC++ 6.0 for programming. After several experiments, it is determined that the scaling factor \( F \) is 0.4, the binomial cross probability \( CR \) is 0.08, and the exponential cross probability \( CR \) is 0.08. For 25 customer sets, the population size is 40, evolution The generation number is 400; for a set of 50 customers, the population size is 100 and the number of evolutionary generations is 1000; for a set of 100 customers, the population size is 100 and the evolutionary number is 1000. Without loss of generality, 18 cases were selected from the 25 customer sets, 50 customer sets, and 100 customer sets of the Solomon study as experimental test data, and each study was solved 30 times. In order to verify the performance of the algorithm in this paper, the following three experimental operations are adopted: (1) The effectiveness of the sorting operation is verified by comparing the improved algorithm based on ranking with the improved algorithm based on randomness(rDE); (2) The stability of the algorithm is verified by recording the average and standard deviation; (3) The superiority of the algorithm is verified by comparing with the optimal solutions of ABC [6] and IDE-VND [7], and comparing the relative deviations with known optimal solutions. These three experimental results are shown in Table 1. In this table, E represents calculation examples, C represents the number of customers, OS represents an optimal solution, KOS represents a known optimal solution, A represents
an average value, SD represents a standard deviation, RD represents a relative deviation, V represents a vehicle, and D represents a distance.

Table 1. Experimental results of Sort-Based DE.

| E  | C   | CDE | ABC | IDE-VND | Ranking-Based DE | KOS | RD |
|----|-----|-----|-----|---------|------------------|-----|----|
|    |     | V   | D   | V   | D   | V   | D   | V   | D   | V   | D   | V   | D   | V   | D   | V   | D   |
| C104 | 25  | 3   | 187.5 | 3 | 187.5 | 3 | 187.5 | 2.6 | 225.73 | 0.489 | 25.309 | 3 | 186.9 | 0.0% | 0.3% |
| C201 | 25  | 2   | 215.5 | 2 | 215.5 | 2 | 215.6 | 2.2 | 273.699 | 0.678 | 44.569 | 2 | 214.7 | 0.0% | 0.4% |
| R103 | 25  | 5   | 455.7 | 5 | 455.7 | 5 | 454.9 | 5.1 | 470.272 | 0.435 | 25.422 | 5 | 454.6 | 0.0% | 0.1% |
| R201 | 25  | 4   | 464.4 | 4 | 464.4 | 4 | 464.1 | 3.3 | 496.543 | 0.641 | 25.209 | 4 | 463.3 | 0.0% | 0.1% |
| RC101 | 25  | 4   | 462.2 | 4 | 463.6 | 4 | 462.4 | 5.3 | 492.677 | 0.557 | 33.971 | 4 | 461.1 | 0.0% | 0.1% |
| RC208 | 25  | 2   | 279.6 | 2 | 269.6 | 2 | 269.6 | 2 | 297.996 | 0.40 | 414.2 | 2 | 269.1 | 0.0% | 0.1% |
| C103 | 50  | 5   | 365.2 | 5 | 392.2 | 5 | 364.2 | 4 | 385.5 | 4.25 | 437.774 | 0.622 | 58.779 | 5 | 361.4 | -20.0% | 6.7% |
| C201 | 50  | 3   | 365.8 | 3 | 373.8 | 3 | 367.8 | 3 | 363.5 | 2.85 | 383.556 | 0.726 | 59.069 | 3 | 360.2 | 33.3% | -7.4% |
| R101 | 50  | 12  | 1099.7 | 13 | 1049.5 | 12 | 1046.7 | 12 | 1046.3 | 12.35 | 1084.63 | 0.735 | 54.545 | 12 | 1044 | 0.0% | 0.1% |
| R201 | 50  | 6   | 794.3 | 7 | 829.8 | 6 | 793.7 | 5.8 | 812.213 | 0.927 | 72.141 | 6 | 791.9 | 0.0% | 0.1% |
| RC101 | 50  | 8   | 956.10 | 9 | 977.1 | 8 | 959.3 | 8 | 957.2 | 9.7 | 969.808 | 1.004 | 73.239 | 8 | 944 | 0.0% | 1.4% |
| RC204 | 50  | 3   | 459.2 | 3 | 459.4 | 3 | 449.2 | 3 | 451.423 | 2.05 | 502.838 | 0.497 | 74.409 | 3 | 444.2 | 0.0% | 1.6% |
| C101 | 100 | 10  | 838.9 | 10 | 828.9 | 10 | 828.9 | 9 | 970.361 | 11.35 | 893.502 | 1.195 | 78.729 | 10 | 827.3 | -10.0% | 17.3% |
| C201 | 100 | 3   | 596.6 | 4 | 618.6 | 3 | 595.6 | 4 | 545.381 | 4.7 | 650.33 | 0.865 | 61.527 | 3 | 589.1 | 33.3% | -7.4% |
| R103 | 100 | 15  | 1235.5 | 15 | 1264.5 | 15 | 1235.5 | 14 | 1284.121 | 14.75 | 1406.93 | 1.219 | 63.241 | 14 | 1208.7 | 0.0% | 6.2% |
| R201 | 100 | 9   | 1174.8 | 11 | 1227.2 | 9 | 1174.8 | 8 | 1211.435 | 8.65 | 1275.654 | 0.783 | 78.226 | 8 | 1143.2 | 0.0% | 6.0% |
| RC101 | 100 | 17  | 1675.1 | 18 | 1698.17 | 17 | 1675.1 | 16 | 1669.45 | 16.7 | 1703.65 | 0.456 | 96.153 | 15 | 1619.8 | 13.3% | 3.1% |
| RC205 | 100 | 8   | 1177.4 | 9 | 1221.6 | 8 | 1177.4 | 7 | 1350.958 | 7.95 | 1310.151 | 0.529 | 97.94 | 7 | 1154 | 0.0% | 17.1% |
|    | 5   | 1461 | 8 | 1249.7 | 5 | 1461 | 9 | 1197.9 | 28.6% | 3.8% |

As can be seen from the table, there are 11 examples on the improved difference algorithm that are worse than the results of the proposed algorithm, it can be seen that the mutation operation based on sorting is effective. The relative deviation and standard deviation of the average of the algorithm are both small, which shows that the algorithm has strong stability. Compared with ABC, the proposed algorithm has obvious advantages: the results of 12 cases are better than ABC; the results of 3 cases are the same as the results of ABC; the results of 3 cases are not dominated by ABC results. Compared with IDE-VND, it also has certain advantages: 9 examples have better results than IDE-VND; 3 examples have the same results as IDE-VND; 4 examples have results that are independent of ABC results; only The results of two examples are worse than IDE-VND.
4. Summary
This paper optimizes the difference algorithm for the vehicle Routing Problem with Time windows in three aspects: the dual-objective optimization problem that uses PARETO dominance to conduct individual evaluation to solve the vehicle routing problem; uses the mutation operation based on ranking to improve the exploration ability and accelerate the convergence speed; Based on the Pareto frontier, the dual-strategy mutation evolution method balances the exploration and development capabilities with the corresponding cross-operations. In this paper, 18 Solomon examples of different scales are tested experimentally. It shows that the algorithm in this paper is an effective method to solve the vehicle Routing Problem with Time windows, and provides a new solution for the vehicle Routing Problem with Time windows. At the same time, it has certain significance for studying other types of vehicle routing problems. Use for reference.

Acknowledgement
This research was financially supported by the Liaoning Natural Science Foundation (2017054767).

References
[1] Dantzig G B, Ramster J. The truck dispatching problem. Management Science. 10 (1959) 80-91.
[2] Solomon M M. Algorithms for the vehicle routing and scheduling problems with time windows constraints. Operations Research. 35 (1987) 254-265.
[3] Setak M, Habibi M, Karimi H, et al. A Time-dependent Vehicle Routing Problem in Multigraph with FIFO Property. Journal of Manufacturing Systems. 5(2016)37-45.
[4] Lin Y, Liu Y, Chen W N, et al. A Hybrid Differential Evolution Algorithm for Mixed-variable Optimization Problems. Information Sciences. 466(2018)170-188.
[5] Chen R, Gen M. Vehicle routing problem with fuzzy due-time using genetic algorithms. Japanese Journal of Fuzzy Theory and Systems. 7(1995)1050-1061.
[6] Jadon S S, Tiwari R, Sharma H, et al. Hybrid Artificial Bee Colony Algorithm with Differential Evolution. Applied Soft Computing. 58(2017)11-24.
[7] Song Xiaoyu, Zhu Jiayuan, Sun Huanliang. A hybrid differential evolution algorithm for vehicle routing problems with time windows. Computer Science. 41 (2017) 220-225.