Chinese Journal of Electronics
Vol.32, No.4, July 2023

Self-Adaptive Discrete Cuckoo Search Algorithm for the Service Routing Problem with Time Windows and Stochastic Service Time

ZHANG Guoyun, WU Meng, LI Wujing, OU Xianfeng, and XIE Wenwu

(*School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang 414006, China*)

Abstract — Making house calls is very crucial to deal with the competitive pressures of the service business and to improve service quality. We design a model called service routing problem with time windows and stochastic service time (SRPTW-SST) that is based on vehicle routing problem with time windows. A self-adaptive discrete cuckoo search algorithm with genetic mechanism (sDCS-GM) is proposed for the model SRPTW-SST. Moreover, we design a selection mechanism to improve the logicality of the algorithm based on the strong randomness of the Lévy flight. We introduce a genetic mechanism and design a neighborhood search mechanism for improving the robustness of the algorithm. In addition, an adaptive parameter adjustment method is designed to eliminate the impact of fixed parameters. The experimental results show that the sDCS-GM algorithm is more robust and effective than the state-of-the-art methods.

Key words — Service routing problem with time window, Stochastic service time, Cuckoo search, Vehicle routing problem with time window, Lévy flight.

I. Introduction

The vehicle routing problem with time windows (VRPTW) is a classic discrete combinatorial optimization problem [1]. The key challenges remain lowering transportation costs and increasing customer satisfaction. Therefore, this work proposes a model of the service routing problem with time windows and stochastic service time (SRPTW-SST) based on VRPTW. With heuristic algorithms being the most popular due to their unique qualities and benefits, the focus of this work was on heuristic approaches for SRPTW-SST, such as genetic algorithm (GA) [2], flower pollination algorithm (FPA) [3], invasive weed optimization (IWO) [4], etc. Yang *et al.* [5] unveiled the cuckoo search algorithm (CS), a new swarn intelligence algorithm inspired by the social behavior of cuckoo birds.

In this paper, a self-adaptive discrete cuckoo search algorithm with genetic mechanism (sDCS-GM) is proposed for VRPTW and SRPTW-SST, in which a Lévy flight-based selection criterion and a deletion and supplementation-based neighborhood search strategy are designed to improve the search performance of the algorithm. Combining the discovery probability Pa and 2-opt, the search capability of the algorithm for local exploitation is improved. The experimental results show that the sDCS-GM algorithm is more robust and effective than the multi-adaptive particle swarm optimization (MAPSO) [6], genetic algorithm with adaptive simulated annealing mutation (GASA) [7], and other state-of-the-art methods. The main contributions are as follows:

1) A novel realistic model named SRPTW-SST based on VRPTW is designed.

2) sDCS-GM is presented for the VRPTW and SRPTW-SST problems.

3) Two selection criteria are designed based on Lévy flight and discovery probability Pa, respectively.

4) A parameter adaptive strategy based on the fitness and the iterations is designed to assign Pa values.

II. The Problem of SRPTW-SST

As a typical NP-hard problem, VRPTW has attracted the attention of more and more researchers. In order to obtain higher economic benefits, the company

Manuscript Received Apr. 4, 2022; Accepted Sept. 26, 2022. This work was supported by the Scientific Research Fund of Education Department of Hunan Province (22A0471), the Hunan Provincial Natural Science Foundation (2020JJ5218), the Engineering Research Center on 3D Reconstruction and Intelligent Application Technology of Hunan Province (2019-430602-73-03-006049), and the Science and Technology Program of Hunan Province (2019TP1014).

 DOI:10.23919/cje.2022.00.072

that provides the service needs to send fewer technicians to complete more work in a day. A typical example of SRPTW-SST is shown in Fig.1.

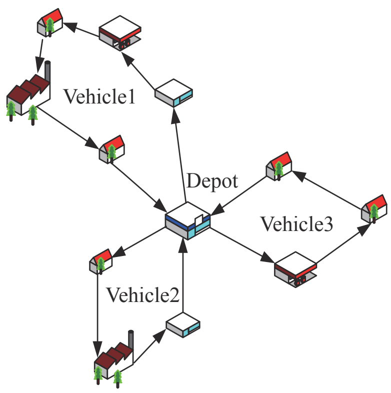

Fig. 1. A classic example of VRPTW.

Table 1 lists the symbols and their meanings used in the VRPTW and SRPTW-SST models.

Table 1. Basic Notions

Symbol	Meanings	Symbol	Meanings
V	Set of all vehicles	q_i	Demand of customer C_i
C	Set of all customers	lt_i	Latest arrival time at C_i
N	Number of customers	st_i	Service time at C_i
Q	Capacity of vehicle	at_i	Arrival time at C_i
TD_{ij}	Distance from C_i to C_j	wt_i	Wait time at C_i
tt_{ij}	Time from C_i to C_j	sst_i	Stochastic service time
et_i	Earliest arrival time	–	–

The formulation of the VRPTW model is shown as

$$\min f(x) = \sum_{k\in V}\sum_{i\in C}\sum_{j\in C} TD_{ij}X_{ijk} \tag{1}$$

where $f(x)$ represents the shortest driving distance of all vehicles. The $X_{ijk}=1$ means that the vehicle k can pass through the road between customers C_i and C_j.

$$\sum_{i\in C} X_{i0k} = \sum_{j\in C} X_{0jk} = 1 \ \ (\forall k \in V) \tag{2}$$

$$\sum_{j\in C, j\neq i} X_{ijk} = \sum_{j\in C, j\neq i} X_{jik} \le 1 \ \ (\forall i \in C, \forall k \in V) \tag{3}$$

$$\sum_{k\in V}\sum_{i\in C, i\neq j} X_{ijk} = 1 \ \ (\forall j \in C) \tag{4}$$

$$\sum_{k\in V}\sum_{j\in C, j\neq i} X_{ijk} = 1 \ \ (\forall i \in C) \tag{5}$$

Equations (2)–(5) are defined to satisfy the restriction conditions that only K vehicles are used to serve customers and each customer is served by one vehicle once. The maximum capacity of each vehicle is specified by

$$\sum_{i\in C} q_i \sum_{j\in C, j\neq i} X_{ijk} \le Q \ \ (\forall k \in V) \tag{6}$$

The time windows constraints are defined by (7)–(9) as following:

$$wt_j = \max\{et_i - lt_i - tt_{ij}, 0\} \ \ (\forall i, j \in C, i\neq j) \tag{7}$$

$$lt_i + st_i + tt_{ij} + wt_i \le lt_j \ \ (\forall i, j \in V, i \neq j) \tag{8}$$

$$et_i \le lt_i + wt_i \le lt_j \ \ (\forall i \in C) \tag{9}$$

According to the actual service time required by customers, st_i is improved to

$$sst_i = st_i + \mathrm{randperm}\,[-a, a] \tag{10}$$

where sst_i is the actual service time, $a \in [-10, 10]$, the formula (8) is adjusted to

$$lt_i + sst_i + tt_{ij} + wt_i \le lt_j \ \ (\forall i, j \in V, i \neq j) \tag{11}$$

III. Proposed Method for SRPTW-SST

1. Genetic mechanism

This mechanism mainly involves selection operator, crossover operator, and mutation operator shown in Algorithms 1–3. There are K vehicles and N customers, the coding length is $K+N-1$. Two random integers r_1 and r_2, $r_1, r_2 \in [1, K+N-1]$. No longer sets the crossover and mutation probability to reduce the number of parameters.

Algorithm 1 Selection operator

$cumfit = cumsum\,(fit)$;
$Nsel = \max\,(floor\,(N \times Gap + 0.5)\,, 2)$;
$tr = \frac{cumfit}{Nsel} \times \left(rand + (0 : Nsel - 1)'\right)$;
Obtain Mf and Mt based on $Nsel$ and N;
Sort index to a random order;
Get a selected list;

Algorithm 2 Crossover operator

for $i = 1 : Nse1$
  $r_1, r_2 \in [1, num\_c]$;
  $tempA = A\,(r_1, r_2)$; $tempB = B\,(r_1, r_2)$;
  $A\,(r_1, r_2) = tempB$; $B\,(r_1, r_2) = tempA$;
  Adjust the same indexes in A and B respectively;
end for

Algorithm 3 Mutation operator

for $i = 1 : Nse1$
  $r_1, r_2 \in [1, num\_c]$;
  $x\,(i, r_1 : r_2) = x\,(i, r_2 : r_1)$;
end for

2. DCS Algorithm

1) Local search based on 2-opt

As shown in Algorithm 4, 40% of $(K+N-1)$ nodes is randomly selected to perform breakpoint operations.

Algorithm 4 Local search based on 2-opt

for $i = 1 : N$
  Select m point (Id);
  for $j = 1 : m$
    $newX(i, (j) : (j+1)) = reverse(X(i, (j) : (j+1)))$;
  end for
end for

2) 3-opt method

As shown in Fig.2, after the variant of 3-opt method, there is the possibility of obtaining multiple offspring. Although keeping individuals with better fitness values is currently the best choice, it has to be considered that near the individuals with better fitness is the local optimal route. This can increase the probability that the algorithm will fall into a local optimum. On the contrary, there will be the following situation: individuals with poor fitness will develop into individuals with better fitness in the subsequent optimization process. Therefore, this paper devises a technique: If $r > sp$, then X_i is an individual with optimal fitness, otherwise it is a suboptimal individual.

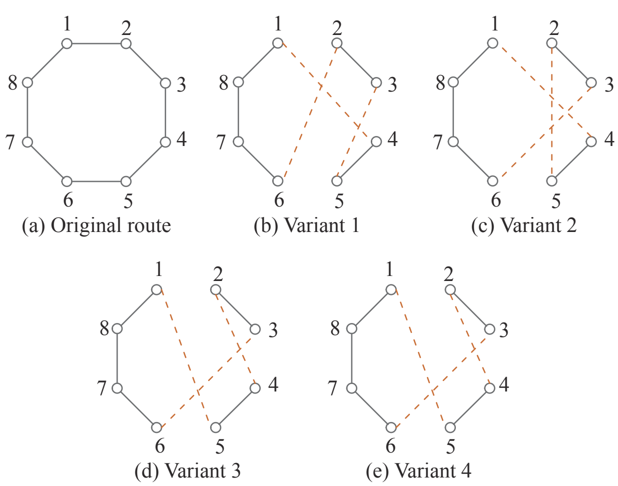

Fig. 2. The variant of 3-opt.

3) Lévy flight

Lévy flight will generate short step lengths in most of the time of the random process and will be accompanied by occasional long step lengths. It can be clearly found that a short step size can help the algorithm to improve the calculation accuracy during local exploitation, and a long step size can help the algorithm get rid of the problem of local optimization in the global exploration stage. CS algorithm relies on the advantages of the combination of long and short step length of Lévy flight to achieve ideal performance for solving continuous optimization problems in (12).

$$LFvalue = u \times |v|^{-\beta^{-1}} \tag{12}$$

where $\beta = 1.5$, $u \sim N\left(0, \sigma^2\right)$ and $v \sim N(0, 1)$. The σ is expressed as

$$\sigma = \left\{ \frac{\Gamma(1+\beta)\sin(\beta \times \pi/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right\}^{\frac{1}{\beta}} \tag{13}$$

where Γ is the gamma function.

3. Self-adaptive mechanism

This paper designs a parameter adaptive mechanism for the probability of discovery to minimize or even eliminate the above-mentioned influences and drawbacks. Only the change of Pa with the change of the number of iterations is considered [8]. This mechanism is not conducive to the balance of search capabilities between global exploration and local search. Therefore, while considering that the parameter Pa changes with the number of iterations, this paper also takes into account the changes in fitness, as shown in (14).

$$Pa' = \left[1.1 - \sin\left(\frac{\max Iter - iter}{\max Iter} \times \frac{\pi}{2}\right)^{\theta}\right] \times f(fit) \tag{14}$$

where $Iter$ represents the current number of iterations, $\max Iter$ represents the maximum number of iterations, $\theta = 0.6$. $f(fit) = \exp\left(-(fit_{iter}/fit_{iter-1})\right)$.

4. Neighborhood search

How to better apply this part to solving discrete combinatorial optimization problems becomes particularly essential. In order to achieve the goal of improving the algorithm's optimization ability as much as possible, this paper designs a neighborhood search mechanism based on random removal and supplement in Algorithms 5 and 6.

Algorithm 5 Neighborhood search

for $i = 1 : N$
  Decode;
  Implement remove operator;
  for $j = 1 : length(removed)$
    Obtain the optimal insertion point;
    Insert into the original route;
  end for
  Adjust the new route;
  Evaluate the fitness of the new population;
  Record optimal results;
end for

Algorithm 6 Remove operator

Choose one customer randomly;
for $i = 1 : num\_re$

```
    Calculate relevance δ between a removed customer
    and all other customers;
    Sort the remaining customer by relevance δ;
    Get the removed list;
  end while
  n = size(Nvc);
  for i = 1 : n
    R = Nvc_i;
    for j = 1 : num_re
      if R == removed_i
        The corresponding R is left blank;
      end if
    end for
    Adjust the new route;
  end for
```

The design of the relevance δ between customer nodes is inspired by the Euclidean distance, the relevanc $\delta = 1/(\varepsilon + \varphi)$, and ε and φ can be stated as

$$\varepsilon = \frac{D\left(de_{i+1}, re_{j+1}\right)}{\max\left(D\left(de_{i+1}, 2 : end\right)\right)} \tag{15}$$

$$\varphi = \begin{cases} 0, & \text{if } i \in VC \text{ and } j \in VC \\ 1, & \text{otherwise} \end{cases} \tag{16}$$

where D represents the distance matrix, de and re represent the matrix of removed nodes and remaining nodes, respectively. VC represents the list of customers served by the current vehicle.

5. sDCS-GM algorithm

Through the analysis of the CS algorithm, as well as the elaboration and demonstration of the improvement ideas mentioned above, this paper proposes an sDCS-GM algorithm for solving VRPTW and SRPTW-SST problems in Algorithm 7.

Algorithm 7 The framework of sDCS-GM

```
  Require: maxIter: the maximum number of iterations;
           N: the number of population; Pa: the discov-
           ery probability.
  Ensure: x: an optimal solution; fit: the optimal fitness.
    Initialize the population;
    Evaluate the fitness of the initial population;
    Encode;
    while iter < maxIter
      Execute the selection operator;
      Update LF with equation (12);
      if LF < cA
        Implement the variant of 3-opt;
        Implement crossover operator;
        Implement muttation operator;
      else
        Implement the local search based on 2-opt;
        Implement muttation operator;
        Implement crossover operator;
      end if
      Evaluate the fitness of the new population;
      for i = 1 : N
        Update Pa' with equation (16);
        if rand < Pa'
          Implement the local search based on 2-opt;
        else
          Implement neighborhood search;
        end if
      end for
      Evaluate the fitness of the new population;
      Decode;
      Record optimal results;
    end while
```

IV. Experiment and Results

In this section, the proposed sDCS-GM algorithm is applied to solve the VRPTW and SRPTW-SST problems, and relevant experimental data are collected andanalyzed.

1. Experiment setting

In order to highlight the effectiveness of the sDCS-GM algorithm, it is chosen to compare with the multi-adaptive particle swarm optimization (MAPSO) [6], genetic algorithm with adaptive simulated annealing mutation (GASA) [7] and discrete cuckoo search algorithm (DCS). The relevant parameters of each algorithm are shown in Table 2. The MAPSO, GASA, DCS, and sDCS-GM algorithms were independently run 30 times for 42 VRP-TW instances with 100 customers. In addition, the efficiency of sDCS-GM is compared to that of the bees algorithm (BA) [9], hybrid shuffled frog leaping algorithm (HSFLA) [10], hybrid ant colony algorithm and brain storm optimization (HACS-BSO) [11], and multi-objective discreet learnable evolution model (MODLEM) [12].

Table 2. Simulation parameters for each algorithm

	Symbol	Value	Meanings
Common	max$Iter$	500	Maximum iteration
	N	50	Population size
sDCS-GM	num_Re	5	Number of removed
	Pa	0.25	Intial discovery probability
	sp	0.2	Selection probability
GASA	pc	0.8	Crossover probability
	pm	0.8	Mutation probability
	P_c	0.9	Selection parameter
	α_1, α_2	0.8, 2	Cooling coefficient
	L_1, L_2	10, 8	Disturbance rounds
	T_0, T_1	2000, 1	Initial and final temperature
MAPSO	V	0	Initial velocity
	c_1, c_2	1.2, 1.2	Acceleration coefficients

2. VRPTW

1) Comparison with the different CS

To demonstrate the effectiveness of the proposed genetic mechanism and neighborhood search, this section compares the proposed sDCS-GM approach to several versions of DCS algorithms, such as discrete cuckoo search algorithm (DCS), cuckoo search with genetic mechanism (CS-GM), and cuckoo search with neighborhood search (CS-NS). The experimental results of the four algorithms are shown in Table 3, where *BKS* denotes the best-known solution.

Table 3. Comparison of results of sDCS-GM with CS-GM, CS-NS, and DCS

Sets	*BKS*		DCS	CS-GM	CS-NS	sDCS-GM	Sets	*BKS*		DCS	CS-GM	CS-NS	sDCS-GM
C101	828.94	Ave	4300.53	4099.02	834.84	**828.94**	R101	1642.88	Ave	3538.35	3387.13	**1671.49**	1674.18
		Min	3933.74	3895.44	**816.18**	828.94			Min	3372.48	3225.29	**1649.30**	1654.07
C102	828.94	Ave	4281.46	4127.73	832.39	**828.49**	R102	1472.62	Ave	3523.90	3375.88	1514.03	**1501.99**
		Min	4034.10	3883.67	816.26	**815.44**			Min	3329.43	3237.67	1482.80	**1482.18**
C103	828.06	Ave	4304.63	4066.94	827.13	**815.44**	R103	1213.62	Ave	3495.43	3364.93	1258.31	**1256.61**
		Min	4093.34	3932.27	**812.99**	815.44			Min	3321.20	3262.02	**1221.54**	1234.36
C104	824.78	Ave	3649.15	3430.70	821.36	**807.04**	R104	976.61	Ave	3449.41	3225.91	1039.32	**1030.47**
		Min	3345.98	3181.74	804.82	**804.28**			Min	3057.53	3035.16	1016.06	**1005.39**
C105	828.94	Ave	4292.12	4082.19	840.41	**822.85**	R105	1360.78	Ave	3526.64	3397.67	1430.35	**1427.65**
		Min	3966.25	3900.48	**819.56**	822.85			Min	3240.17	3255.21	1391.46	**1385.99**
C106	828.91	Ave	4295.41	4071.47	849.52	**820.61**	R106	1240.47	Ave	3499.25	3336.51	1296.61	**1289.92**
		Min	4035.97	3912.41	826.85	**820.61**			Min	3256.69	3249.38	**1245.99**	1259.83
C107	828.94	Ave	4264.37	4109.16	835.51	**820.61**	R107	1073.34	Ave	3475.71	3351.47	1140.85	**1128.06**
		Min	3875.93	3699.07	**818.66**	820.61			Min	3331.70	3190.48	1105.62	**1086.24**
C201	591.56	Ave	4432.22	4139.89	652.75	620.99	R201	1147.80	Ave	3708.95	3532.91	**1164.27**	1184.34
		Min	4179.41	3893.82	623.39	**591.56**			Min	3455.58	3406.73	**1122.43**	1159.34
C202	591.56	Ave	4364.28	4143.50	666.45	**629.82**	R202	1034.35	Ave	3674.80	3538.32	1088.35	**1073.39**
		Min	4090.10	3929.68	628.87	**591.56**			Min	3476.72	3422.93	**1055.17**	1057.23
C203	591.17	Ave	4240.01	3840.69	670.08	**638.52**	R203	874.87	Ave	3360.43	3066.29	927.34	**916.36**
		Min	3776.74	3683.64	614.64	**603.37**			Min	3004.84	2891.31	901.19	**893.82**
C204	590.60	Ave	3348.99	3157.62	662.70	**647.61**	R204	735.80	Ave	2859.85	2611.37	797.80	**775.27**
		Min	3157.15	2981.10	618.66	**590.60**			Min	2676.66	2534.28	772.17	**758.97**
C205	588.88	Ave	4412.73	4231.88	658.04	**602.85**	R205	954.16	Ave	3751.91	3471.35	**1006.02**	1006.66
		Min	4105.78	4074.76	621.74	**588.88**			Min	3366.56	3309.15	**976.16**	981.84
C206	588.49	Ave	4387.80	4159.86	661.38	**636.70**	R206	879.89	Ave	3459.02	3151.74	**930.93**	931.59
		Min	4145.66	3866.51	616.14	**588.49**			Min	3306.71	2918.82	**867.67**	901.53
C207	588.29	Ave	4350.04	4142.61	**619.01**	629.19	R207	799.86	Ave	3096.57	2811.81	865.30	**856.08**
		Min	3906.01	3857.93	**582.63**	588.29			Min	2851.37	2648.41	827.18	**824.53**
RC101	1623.58	Ave	4511.48	4461.17	1709.10	**1699.46**	RC201	1265.56	Ave	4861.12	4623.45	**1297.85**	1305.87
		Min	4179.15	4276.62	1664.53	**1648.57**			Min	4487.61	4472.81	1268.90	**1267.97**
RC102	1461.23	Ave	4593.83	4351.02	1545.67	**1507.27**	RC202	1095.64	Ave	4775.28	4630.20	1141.98	**1134.53**
		Min	4153.73	4103.42	1506.84	**1471.16**			Min	4549.74	4440.15	1110.45	**1103.56**
RC103	1261.67	Ave	4529.02	4344.18	1348.97	**1345.87**	RC203	928.51	Ave	4437.70	4086.72	985.33	**970.59**
		Min	4326.67	4096.27	**1268.41**	1280.69			Min	4219.00	3844.75	951.93	**945.39**
RC104	1135.48	Ave	4503.85	4260.71	1211.91	**1183.75**	RC204	786.38	Ave	3737.60	3447.50	848.83	**835.67**
		Min	4123.74	3724.73	1153.28	**1127.65**			Min	3522.40	3308.22	830.60	**805.42**
RC105	1518.58	Ave	4532.46	4275.61	1604.19	**1595.88**	RC205	1157.55	Ave	4844.63	4474.70	1200.59	**1191.64**
		Min	4248.80	4028.78	1559.27	**1557.99**			Min	4443.93	4237.60	1171.66	**1164.85**
RC106	1371.69	Ave	4451.94	4325.05	1453.66	**1439.13**	RC206	1054.61	Ave	4773.55	4510.86	**1096.29**	1104.22
		Min	4129.44	3657.25	**1366.39**	1408.39			Min	4392.61	4313.23	**1049.72**	1072.32
RC107	1212.83	Ave	4407.98	4151.09	1311.16	**1299.88**	RC207	966.08	Ave	4668.09	4132.40	**1000.01**	1009.65
		Min	4058.11	4009.58	**1190.20**	1241.39			Min	4173.77	3947.37	**949.97**	966.08

Table 3 shows that the genetic mechanism can significantly improve the performance of the DCS algorithm. Neighborhood search can significantly reduce the vehicle travel distance, and the CS-NS method shows strong competitiveness and superiority compared with the DCS and CS-GM methods. Even compared with the sDCS-GM method, more desirable results can be obtained for a few instances. However, overall the sDCS-GM method exhibits more satisfactory stability along with better performance of the search for superiority. Therefore, the experimental results demonstrate that GM and NS have different degrees of performance improvement for the DCS method. Compared with the sDCS-GM method, it is also shown that the GM can effectively improve the stability of the NS strategy.

2) Comparison with the MAPSO and GASA

The experimental results obtained by MAPSO, DCS, GASA, and the proposed sDCS-GM method are shown in Table 4, including the minimum and average values, for 42 VRPTW instances.

In Table 4, the proposed sDCS-GM method is able to obtain shorter driving routes than MAPSO, GASA and DCS method paths for most of the instances. For C101–C107, the distances obtained by the proposed method are all less than or equal to the best-known solution BKS, for example, C101, C104, which is a performance that the comparison method does not achieve. For C201–C207, although both the proposed algorithm and MAPSO algorithm are able to obtain BKS for some instances, MAPSO is more stable. In addition, for C205 and C206, the proposed algorithm is able to obtain more optimal driving routes. For R101–R207, although the proposed algorithm does not obtain BKS values, the difference with BKS is small. In addition, only the MAPSO algorithm performs better than the proposed algorithm for R104, R106 and R205 instances. And for RC101–RC207, the MAPSO algorithm only performs better than the proposed algorithm at RC103.

Table 4. Comparison of results of sDCS-GM with MAPSO, GAASAM, and DCS for VRPTW

Sets	BKS		MAPSO	GASA	DCS	sDCS-GM	Sets	BKS		MAPSO	GASA	DCS	sDCS-GM
C101	828.94	Ave	924.58	842.69	4300.53	**828.94**	R101	1642.88	Ave	1714.25	1750.61	3538.35	**1674.18**
		Min	846.48	832.16	3933.74	**828.94**			Min	1692.89	1643.27	3372.48	**1654.07**
C102	828.94	Ave	886.62	854.22	4281.46	**828.49**	R102	1472.62	Ave	1541.01	1607.78	3523.90	**1501.99**
		Min	847.26	829.11	4034.10	**815.44**			Min	1529.47	1504.80	3329.43	**1482.18**
C103	828.06	Ave	888.99	928.73	4304.63	**815.44**	R103	1213.62	Ave	1271.38	1328.16	3495.44	**1256.61**
		Min	853.22	832.10	4093.34	**815.44**			Min	1267.35	1218.80	3321.21	**1234.36**
C104	824.78	Ave	857.50	837.71	3649.15	**807.04**	R104	976.61	Ave	**979.44**	1229.94	3449.41	1030.47
		Min	831.97	828.72	3345.98	**804.28**			Min	**977.93**	1083.48	3057.53	1005.39
C105	828.94	Ave	863.94	871.82	4292.12	**822.85**	R105	1360.78	Ave	1437.12	1448.02	3526.64	**1427.65**
		Min	828.37	838.52	3966.25	**822.85**			Min	1423.12	1361.76	3240.17	**1385.99**
C106	828.91	Ave	853.50	867.96	4295.41	**820.61**	R106	1240.47	Ave	**1279.85**	1329.54	3499.25	1289.92
		Min	825.16	838.33	4035.97	**820.61**			Min	**1255.31**	1272.64	3256.69	1259.83
C107	828.94	Ave	853.51	846.36	4264.37	**820.61**	R107	1073.34	Ave	1137.47	1145.77	3475.71	**1128.06**
		Min	825.28	828.94	3875.93	**820.61**			Min	1132.12	1119.93	3331.70	**1086.24**
C201	591.56	Ave	**591.91**	747.60	4432.22	620.99	R201	1147.80	Ave	1263.12	1211.69	3708.95	**1184.34**
		Min	**591.56**	607.69	4179.41	**591.56**			Min	1203.16	1159.32	3455.58	**1159.34**
C202	591.56	Ave	**597.63**	662.71	4364.28	629.82	R202	1034.35	Ave	1146.68	1124.65	3674.80	**1073.39**
		Min	592.63	602.65	4090.10	**591.56**			Min	1057.35	1063.47	3476.72	**1057.23**
C203	591.17	Ave	**591.56**	676.62	4240.01	638.52	R203	874.87	Ave	899.87	996.65	3360.43	**916.36**
		Min	**591.56**	594.78	3776.74	603.37			Min	895.35	912.29	3004.84	**893.82**
C204	590.60	Ave	**596.16**	705.05	3348.99	647.61	R204	735.80	Ave	811.36	826.45	2859.85	**775.27**
		Min	**590.60**	596.09	3157.15	**590.60**			Min	762.67	768.66	2676.66	**758.97**
C205	588.88	Ave	655.91	712.55	4412.73	**602.85**	R205	954.16	Ave	**980.43**	1068.17	3751.91	1006.66
		Min	645.79	615.07	4105.78	**588.88**			Min	**897.57**	1015.30	3366.56	981.84
C206	588.49	Ave	**593.83**	661.55	4387.80	636.70	R206	879.89	Ave	968.32	1041.82	3459.02	**931.59**
		Min	590.28	606.19	4145.66	**588.49**			Min	925.64	971.61	3306.71	**901.53**
C207	588.29	Ave	**593.04**	773.52	4350.04	629.19	R207	799.86	Ave	866.24	904.83	3096.57	**856.08**
		Min	**583.88**	616.64	3906.01	588.29			Min	**821.04**	848.25	2851.37	824.53
RC101	1623.58	Ave	1703.33	1819.46	4511.48	**1699.46**	RC201	1265.56	Ave	1310.82	1430.10	4861.12	**1305.87**
		Min	1691.57	1703.74	4179.15	**1648.57**			Min	1272.88	1258.19	4487.61	**1267.97**
RC102	1461.23	Ave	1508.98	1636.85	4593.83	**1507.27**	RC202	1095.64	Ave	1141.84	1228.03	4775.28	**1134.53**
		Min	1497.62	1505.92	4153.73	**1471.16**			Min	1108.27	1130.22	4549.74	**1103.56**
RC103	1261.67	Ave	**1306.31**	1397.66	4529.03	1345.87	RC203	928.51	Ave	986.01	1049.99	4437.70	**970.59**
		Min	**1269.61**	1327.77	4326.67	1280.69			Min	949.75	952.80	4219.01	**945.39**
RC104	1135.48	Ave	1209.05	1394.59	4503.85	**1183.75**	RC204	786.38	Ave	843.80	1099.19	3737.60	**835.67**
		Min	1173.88	1213.62	4123.75	**1127.65**			Min	816.32	847.61	3522.40	**805.42**
RC105	1518.58	Ave	1602.12	1708.65	4532.46	**1595.88**	RC205	1157.55	Ave	1196.40	1356.11	4844.63	**1191.64**
		Min	1558.60	1618.97	4248.80	**1557.99**			Min	1160.23	1238.91	4443.93	**1164.85**
RC106	1371.69	Ave	1450.13	1486.58	4451.94	**1439.13**	RC206	1054.61	Ave	1181.03	1416.09	4773.55	**1104.22**
		Min	1419.03	1245.20	4129.44	**1408.39**			Min	1151.88	1263.83	4392.61	**1072.32**
RC107	1212.83	Ave	1722.93	1402.08	4407.98	**1299.88**	RC207	966.08	Ave	1045.08	1489.65	4668.09	**1009.65**
		Min	1681.47	1306.54	4058.11	**1241.39**			Min	984.08	1422.69	4173.77	**966.08**

$$Dis\_Gap = \frac{Dis_{\text{others}} - Dis_{\text{ours}}}{Dis_{\text{ours}}} \times 100\% \tag{17}$$

$$Robust = \frac{Best - Average}{Average} \times 100\% \tag{18}$$

In Table 5, the values calculated from (22) show that the proposed sDCS-GM algorithm and MAPSO algorithm perform more consistently, for example, sDCS-GM (C1, R2 and RC2), MAPSO (C2, R1 and RC1).

In Table 6, Dis_Gap between GASA and the proposed sDCS-GM algorithm are all greater than 0. Dis_Gap between both DCS and the proposed sDCS-GM algorithm is greater than 110. Overall, the sDCS-GM exhibits much better performance in terms of stability than the comparative algorithms such as MAPSO and GASA. To highlight the superiority and competitiveness of the proposed sDCS-GM algorithm in terms of convergence speed, we make convergence curves for some of the instances, as shown in Fig.3. We make a roadmap of the shortest driving path obtained by the proposed sDCS-GM algorithm for some instances, as shown in Fig.4.

This paper compares the sDCS-GM algorithm against other state-of-the-art algorithms such as the HSFLA, BA, ACS-BSO, and MODLEM in order to demonstrate its competitiveness and superiority. The

Table 5. Comparison results of the robust (%) for VRPTW

Sets	MAPSO	GASA	DCS	sDCS-GM	Sets	MAPSO	GASA	DCS	sDCS-GM	Sets	MAPSO	GASA	DCS	sDCS-GM
C101	−8.45	−1.25	−8.53	0.00	R101	−1.25	−6.13	−4.69	−1.20	RC101	−0.69	−6.36	−7.37	−2.99
C102	−4.44	−2.94	−5.78	−1.57	R102	−0.75	−6.40	−5.52	−1.32	RC102	−0.75	−8.00	−9.58	−2.39
C103	−4.02	−10.40	−4.91	0.00	R103	−0.32	−8.23	−4.98	−1.77	RC103	−2.81	−5.00	−4.47	−4.84
C104	−2.98	−1.07	−8.31	−0.34	R104	−0.15	−11.91	−11.36	−2.43	RC104	−2.91	−12.98	−8.44	−4.74
C105	−4.12	−3.82	−7.59	0.00	R105	−0.97	−5.96	−8.12	−2.92	RC105	−2.72	−5.25	−6.26	−2.37
C106	−3.32	−3.41	−6.04	0.00	R106	−1.92	−4.28	−6.93	−2.33	RC106	−2.14	−16.24	−7.24	−2.14
C107	−3.31	−2.06	−9.11	0.00	R107	−0.47	−2.25	−4.14	−3.71	RC107	−2.41	−6.81	−7.94	−4.50
Average	−4.38	−3.57	−7.18	−0.27	Average	−0.83	−6.45	−6.53	−2.24	Average	−2.06	−8.66	−7.33	−3.42
C201	−0.06	−18.71	−5.70	−4.74	R201	−4.75	−4.32	−6.83	−2.11	RC201	−2.89	−12.02	−7.68	−2.90
C202	−0.84	−9.06	−6.28	−6.08	R202	−7.79	−5.44	−5.39	−1.51	RC202	−2.94	−7.96	−4.72	−2.73
C203	0.00	−12.10	−10.93	−5.51	R203	−0.50	−8.46	−10.58	−2.46	RC203	−3.68	−9.26	−4.93	−2.60
C204	−0.93	−15.45	−5.73	−8.80	R204	−6.00	−6.99	−6.41	−2.10	RC204	−3.26	−22.89	−5.76	−3.62
C205	−1.54	−13.68	−6.96	−2.32	R205	−8.45	−4.95	−10.27	−2.47	RC205	−3.02	−8.64	−8.27	−2.25
C206	−0.60	−8.37	−5.52	−7.57	R206	−4.41	−6.74	−4.40	−3.23	RC206	−2.47	−10.75	−7.98	−2.89
C207	−1.54	−20.28	−10.21	−6.50	R207	−5.22	−6.25	−7.92	−3.69	RC207	−5.84	−4.50	−10.59	−4.32
Average	−0.79	−13.95	−7.33	−5.93	Average	−5.30	−6.17	−7.40	−2.51	Average	−3.44	−10.86	−7.13	−3.04

Table 6. Comparison results of the cost gap (%)

Sets	MAPSO	GASA	DCS	Sets	MAPSO	GASA	DCS
C101	11.54	1.66	418.80	R101	2.39	4.57	111.35
C102	7.02	3.11	416.78	R102	2.60	7.04	134.62
C103	9.02	13.89	427.89	R103	1.18	5.69	178.16
C104	6.25	3.80	352.16	R104	−4.95	19.36	234.74
C105	4.99	5.95	421.62	R105	0.66	1.43	147.02
C106	4.01	5.77	423.44	R106	−0.78	3.07	171.28
C107	4.01	3.14	419.66	R107	0.83	1.57	208.11
C201	−4.68	20.39	613.73	R201	6.65	2.31	213.17
C202	−5.11	5.22	592.94	R202	6.83	4.78	242.35
C203	−7.36	5.97	564.03	R203	−1.80	8.76	266.72
C204	−7.94	8.87	416.77	R204	4.65	6.60	268.89
C205	8.80	18.20	631.98	R205	−2.61	6.11	272.71
C206	−6.73	3.90	589.15	R206	3.94	11.83	271.30
C207	−5.75	22.94	591.37	R207	1.19	5.70	261.72
RC101	0.23	7.06	165.47	RC201	0.38	9.51	272.25
RC102	0.11	8.60	204.78	RC202	0.65	8.24	320.91
RC103	−2.94	3.85	236.51	RC203	1.59	8.18	357.22
RC104	2.14	17.81	280.47	RC204	0.97	31.53	347.26
RC105	0.39	7.07	184.01	RC205	0.40	13.80	306.55
RC106	0.76	3.30	209.35	RC206	6.96	28.24	332.30
RC107	2.14	17.81	239.11	RC207	3.51	47.54	362.35

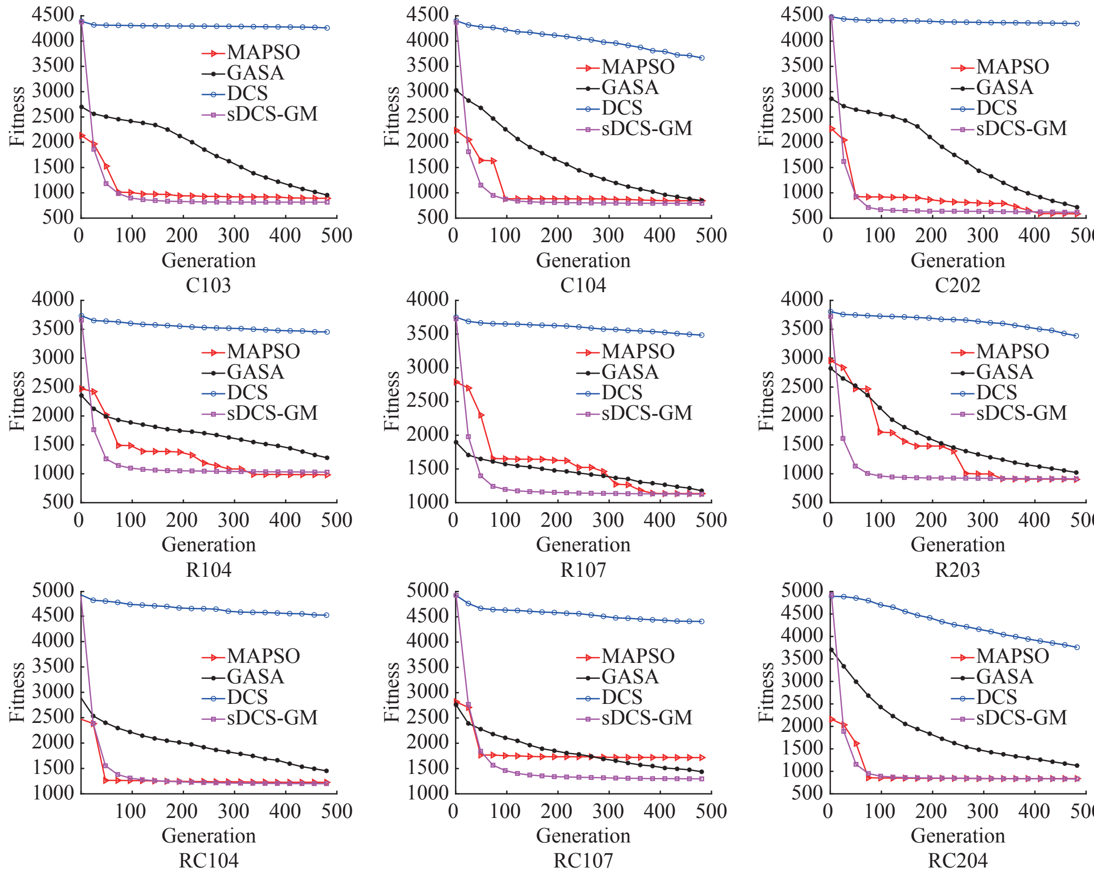

Fig. 3. Comparison of convergence performance of C103, C104, C202, R104, R107, R203, RC104, RC107, and RC204.

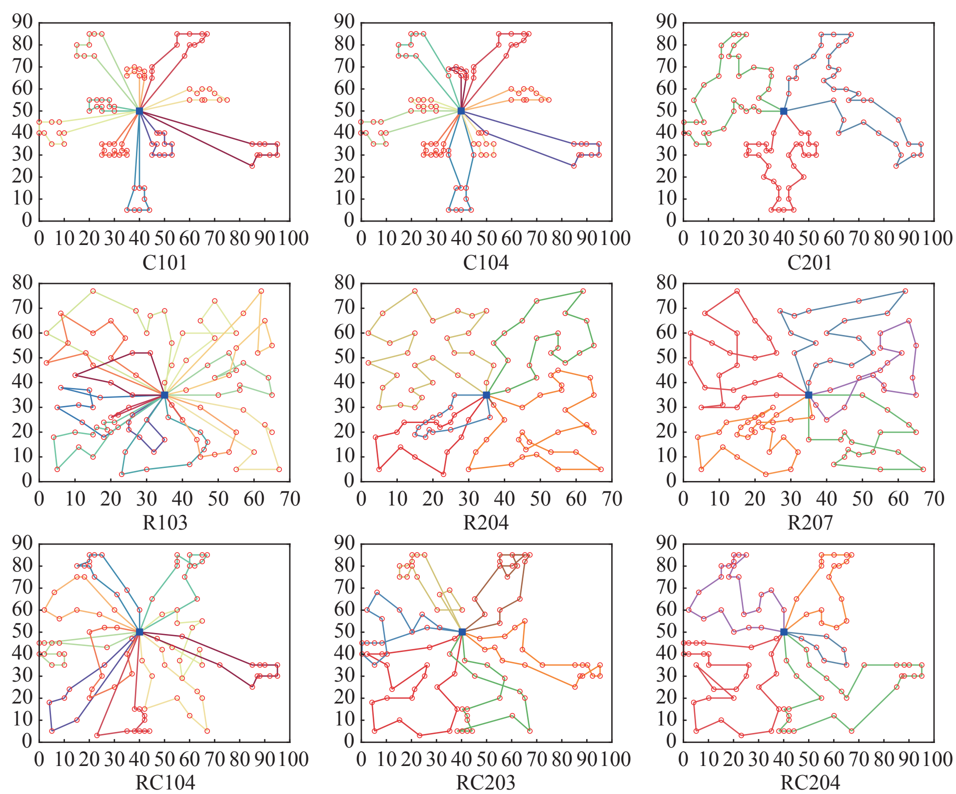

Fig. 4. Optimal route of several C, R and RC instances by sDCS-GM (C101, C104, C201, R103, R204, R207, RC104, RC203, and RC204.

average values of the five algorithms' results are shown in Table 7. This demonstrates that the sDCS-GM algorithm has a significant competitiveness and superiority.

Table 7. Comparison of results of sDCS-GM with other state-of-the-art methods

Sets	HSFLA	BA	ACS-BSO	MODLEM	sDCS-GM	Sets	HSFLA	BA	ACS-BSO	MODLEM	sDCS-GM
C101	**828.94**	**828.94**	**828.94**	829.04	**828.94**	R201	1252.88	1185.57	1336.05	1252.47	**1184.34**
C102	828.94	828.94	828.94	829.04	**828.49**	R202	1192.27	1103.15	1128.05	1191.80	**1073.39**
C103	828.06	828.94	828.06	828.17	**815.44**	R203	939.95	958.94	1020.10	939.60	**916.36**
C104	824.78	858.90	828.78	828.88	**807.04**	R204	826.31	818.44	834.92	**731.40**	775.27
C105	828..94	828.94	824.94	829.04	**822.85**	R205	994.80	1020.53	1105.38	**964.20**	1006.66
C106	828.94	828.94	828.94	829.04	**820.61**	R206	906.59	960.29	949.11	**887.70**	931.59
C107	828.94	828.94	828.94	829.04	**820.61**	R207	891.14	905.70	812.35	**807.10**	856.08
RC201	1407.22	1308.76	1514.41	1407.04	**1305.87**	RC205	1298.26	1210.68	1360.91	1297.75	**1191.64**
RC202	1365.96	1167.00	1326.71	1365.75	**1134.53**	RC206	1146.87	1112.38	1237.21	1146.42	**1104.22**
RC203	1050.14	1014.79	1166.91	1049.72	**970.59**	RC207	1061.50	1059.62	1039.59	**759.33**	1009.65
Sum	1	1	1	0	10	Sum	0	0	0	5	5

3. SRPTW-SST

In this paper, an SRPTW-SST based on VRPTW is designed for realistic requirements. The instances is also transformed into SC, SR and SRC according to (10). The experimental results obtained by the proposed sDCS-GM method with MAPSO, GASA, and DCS, including the average and optimum values, are shown in Table 8. Tables 9 and 10 show the robustness of the sDCS-GM method and the gap between the sDCS-GM method and other competitors, respectively. To highlight the superiority and competitiveness of the proposed sDCS-GM algorithm, we make convergence curves for some of the instances, as shown in Fig.5.

Combined with the analysis of the experimental results, the sDCS-GM algorithm is in the absolute lead compared with the DCS algorithm. For SC101–SC207,

Table 8. Comparison of results of sDCS-GM with MAPSO, GASA, and DCS for SRPTW-SST

Sets		MAPSO	GASA	DCS	sDCS-GM	Sets		MAPSO	GASA	DCS	sDCS-GM
SC101	Ave	867.38	847.51	4318.31	**842.09**	SR101	Ave	**1676.05**	1774.57	3475.13	**1696.26**
	Min	837.27	830.10	4110.27	**820.73**		Min	1599.07	1739.21	3200.92	**1589.34**
SC102	Ave	861.52	843.31	4332.27	**841.96**	SR102	Ave	1608.71	1580.98	3501.11	**1513.61**
	Min	821.73	827.38	4080.82	**820.73**		Min	1478.11	1565.26	3330.09	**1467.33**
SC103	Ave	842.58	840.97	4175.51	**818.18**	SR103	Ave	1349.35	1338.19	3432.01	**1258.26**
	Min	822.67	827.43	3901.45	**815.44**		Min	1234.84	1310.39	3137.54	**1201.93**
SC104	Ave	821.22	868.70	3619.72	**808.55**	SR104	Ave	**1032.11**	1132.56	3450.96	**1054.90**
	Min	809.54	847.97	3236.52	**803.54**		Min	**985.57**	1105.83	3281.07	**991.21**
SC105	Ave	843.35	895.26	4338.09	**831.77**	SR105	Ave	1446.36	1506.07	3438.92	**1428.58**
	Min	824.09	869.17	4081.95	**820.61**		Min	1395.70	1465.70	3174.24	**1370.21**
SC106	Ave	853.92	921.30	4305.16	**850.70**	SR106	Ave	1312.15	1367.00	3453.23	**1295.83**
	Min	825.23	885.93	3960.34	**821.38**		Min	1264.53	1335.30	3221.55	**1254.75**
SC107	Ave	838.21	899.89	4281.93	**823.84**	SR107	Ave	1174.05	1209.46	3490.89	**1123.69**
	Min	823.50	872.68	3986.99	**820.61**		Min	1092.61	1160.08	3232.06	**1086.62**
SC201	Ave	**618.30**	713.55	4414.03	636.03	SR201	Ave	1222.14	1252.84	3674.73	**1180.61**
	Min	**591.56**	679.65	4153.47	**591.56**		Min	1151.13	1221.24	3541.25	**1149.78**
SC202	Ave	**639.65**	703.86	4309.56	**639.07**	SR202	Ave	1149.31	1139.18	3745.02	**1071.10**
	Min	631.59	678.73	4016.91	**591.56**		Min	1040.51	1107.80	3606.84	**1029.73**
SC203	Ave	**630.72**	715.68	4131.18	636.19	SR203	Ave	1030.61	993.30	3350.41	**915.11**
	Min	606.66	688.34	3503.74	**588.49**		Min	907.77	959.62	3076.64	**886.93**
SC204	Ave	**641.03**	727.25	3346.12	649.88	SR204	Ave	781.21	852.04	2847.29	**778.49**
	Min	620.35	703.09	3159.13	**599.29**		Min	755.90	818.15	2666.31	**742.71**
SC205	Ave	645.11	700.87	4390.64	**634.17**	SR205	Ave	1011.44	1083.73	3772.80	**1008.68**
	Min	628.14	673.53	4119.04	**588.88**		Min	989.49	1042.19	3428.79	**973.63**
SC206	Ave	**622.30**	716.09	4346.67	639.52	SR206	Ave	935.76	1004.58	3480.67	**934.68**
	Min	597.35	690.36	4119.77	**588.49**		Min	909.59	974.99	3233.91	**903.39**
SC207	Ave	**619.25**	704.04	4399.85	625.70	SR207	Ave	**835.92**	927.90	3041.98	859.64
	Min	590.39	676.16	4109.49	**588.29**		Min	819.19	898.82	2842.69	**816.33**

Table 8 (Continued)

Sets		MAPSO	GASA	DCS	sDCS-GM	Sets		MAPSO	GASA	DCS	sDCS-GM
SRC101	Ave	1810.71	1785.80	4448.19	**1702.88**	SRC201	Ave	1333.58	1379.15	4824.91	**1306.32**
	Min	1680.21	1753.08	4104.93	**1604.73**		Min	1306.84	1354.06	4511.92	**1279.19**
SRC102	Ave	**1485.51**	1599.27	7553.72	1526.88	SRC202	Ave	1216.37	1213.60	4791.81	**1137.58**
	Min	1448.64	1566.39	4313.63	**1469.40**		Min	1110.84	1175.51	4529.40	**1101.73**
SRC103	Ave	**1314.60**	1416.40	4453.75	1334.29	SRC203	Ave	985.01	1055.87	4395.03	**974.50**
	Min	1252.97	1384.02	4191.75	**1245.15**		Min	956.11	1012.98	4104.63	**945.76**
SRC104	Ave	1181.77	1265.53	4363.48	**1180.34**	SRC204	Ave	843.38	894.30	3707.85	**827.98**
	Min	1111.99	1237.31	4129.67	**1108.75**		Min	807.46	866.16	3424.50	**801.54**
SRC105	Ave	1564.71	1625.21	4505.62	**1559.65**	SRC205	Ave	1220.95	1262.58	4795.70	**1188.42**
	Min	1448.30	1599.93	4325.25	**1443.79**		Min	1171.99	1225.74	4497.73	**1154.90**
SRC106	Ave	**1412.20**	1519.57	4464.37	**1437.99**	SRC206	Ave	1111.41	1178.57	4843.75	**1099.56**
	Min	1354.70	1485.17	4163.37	**1351.50**		Min	1086.34	1144.10	4654.30	**1064.02**
SRC107	Ave	**1291.59**	1365.29	4389.08	**1291.30**	SRC207	Ave	1033.69	1087.23	4665.67	**1006.00**
	Min	1229.23	1332.60	3951.22	**1228.92**		Min	996.56	1057.60	4415.26	**979.06**

Table 9. Comparison results of the robust (%) for SRPTW-SST

Sets	MAPSO	GASA	DCS	sDCS-GM	Sets	MAPSO	GASA	DCS	sDCS-GM	Sets	MAPSO	GASA	DCS	sDCS-GM
SC101	−3.47	−2.05	−4.82	−2.54	SR101	−4.59	−1.99	−7.89	−6.30	SRC101	−7.207	−1.83	−7.72	−5.76
SC102	−4.62	−1.89	−5.80	−2.52	SR102	−8.12	−0.99	−4.88	−3.06	SRC102	−2.482	−2.06	−42.89	−3.76
SC103	−2.36	−1.61	−6.56	−0.33	SR103	−8.49	−2.08	−8.58	−4.48	SRC103	−4.688	−2.29	−5.88	−6.68
SC104	−1.42	−2.39	−10.59	−0.62	SR104	−4.51	−2.36	−4.92	−6.04	SRC104	−5.905	−2.23	−5.36	−6.07
SC105	−2.28	−2.91	−5.90	−1.34	SR105	−3.50	−2.68	−7.70	−4.09	SRC105	−7.439	−1.56	−4.00	−7.43
SC106	−3.36	−3.84	−8.01	−3.45	SR106	−3.63	−2.32	−6.71	−3.17	SRC106	−4.072	−2.26	−6.74	−6.01
SC107	−1.76	−3.02	−6.89	−0.39	SR107	−6.94	−4.08	−7.41	−3.30	SRC107	−4.829	−2.39	−9.98	−4.83
Average	−2.75	−2.53	−6.94	−1.60	Average	−5.68	−2.36	−6.87	−4.35	Average	−5.23	−2.09	−11.80	−5.79
SC201	−4.33	−4.75	−5.90	−6.99	SR201	−5.81	−2.52	−3.63	−2.61	SRC201	−9.81	−1.82	−6.49	−2.08
SC202	−1.26	−3.57	−6.79	−7.43	SR202	−9.47	−2.75	−3.69	−3.86	SRC202	−8.68	−3.14	−5.48	−3.15
SC203	−3.82	−3.82	−15.19	−7.50	SR203	−11.92	−3.39	−8.17	−3.08	SRC203	−2.93	−4.06	−6.61	−2.95
SC204	−3.23	−3.32	−5.59	−7.78	SR204	−3.24	−3.98	−6.36	−4.60	SRC204	−4.26	−3.15	−7.64	−3.19
SC205	−2.63	−3.90	−6.19	−7.14	SR205	−2.17	−3.83	−9.12	−3.47	SRC205	−4.01	−2.92	−6.21	−2.82
SC206	−4.01	−3.59	−5.22	−7.98	SR206	−2.80	−2.95	−7.09	−3.35	SRC206	−2.26	−2.93	−3.91	−3.23
SC207	−4.66	−3.96	−6.60	−5.98	SR207	−2.00	−3.13	−6.55	−5.04	SRC207	−3.59	−2.73	−5.37	−2.68
Average	−3.42	−3.85	−7.35	−7.26	Average	−5.34	−3.22	−6.37	−3.72	Average	−5.08	−2.96	−5.96	−2.87

Table 10. Comparison results of the cost gap (%)

Sets	MAPSO	GASA	DCS	Sets	MAPSO	GASA	DCS
SC101	3.00	0.64	412.81	SR101	−1.19	4.62	104.87
SC102	2.32	0.16	414.55	SR102	6.28	4.45	131.31
SC103	2.98	2.78	410.34	SR103	7.24	6.35	172.76
SC104	1.57	7.44	347.68	SR104	−2.16	7.36	227.14
SC105	1.39	7.63	421.55	SR105	1.24	5.42	140.72
SC106	0.38	8.30	406.07	SR106	1.26	5.49	166.49
SC107	1.74	9.23	419.75	SR107	4.48	7.63	210.66
SC201	−2.79	12.19	593.99	SR201	3.52	6.12	211.26
SC202	0.09	10.14	574.35	SR202	7.30	6.36	249.64
SC203	−0.86	12.50	549.36	SR203	12.62	8.54	266.12
SC204	−1.36	11.90	414.88	SR204	0.35	9.45	265.75
SC205	1.73	10.52	592.34	SR205	0.27	7.44	274.03
SC206	−2.69	11.97	579.68	SR206	0.12	7.48	272.39
SC207	−1.03	12.52	603.19	SR207	−2.76	7.94	253.87
SRC101	6.33	4.87	314.60	SRC201	2.09	5.58	269.35
SRC102	−2.71	4.74	394.72	SRC202	6.93	6.68	321.23
SRC103	−1.48	6.15	233.79	SRC203	1.08	8.35	351.00
SRC104	0.12	7.22	269.68	SRC204	1.86	8.01	347.82
SRC105	0.32	4.20	188.89	SRC205	2.74	6.24	303.54
SRC106	−1.79	5.67	210.46	SRC206	1.08	7.19	340.52
SRC107	0.12	7.22	239.90	SRC207	2.75	8.07	363.78

the *Dis_Gap* values indicates that the proposed algorithm has 3–6 times the performance of the DCS algorithm.

In Fig.5, it is obvious that the DCS algorithm converges slowly during the iterative process and only shortens the distance partially than the initialized driving path. The MAPSO method outperforms the proposed sDCS-GM algorithm in terms of the average of 30 experimental data, but sDCS-GM can also get very good results in terms of the *Min*. The proposed sDCS-GM algorithm still performs well in terms of stability and robustness. Compared with the MAPSO algorithm, the sDCS-GM algorithm is superior in the SC1, SR1, SR2 and SRC2 series of instances. Compared with the GASA algorithm, the difference between the two is small, although it does not dominate across the board. Overall, the proposed sDCS-GM algorithm has more desirable robustness.

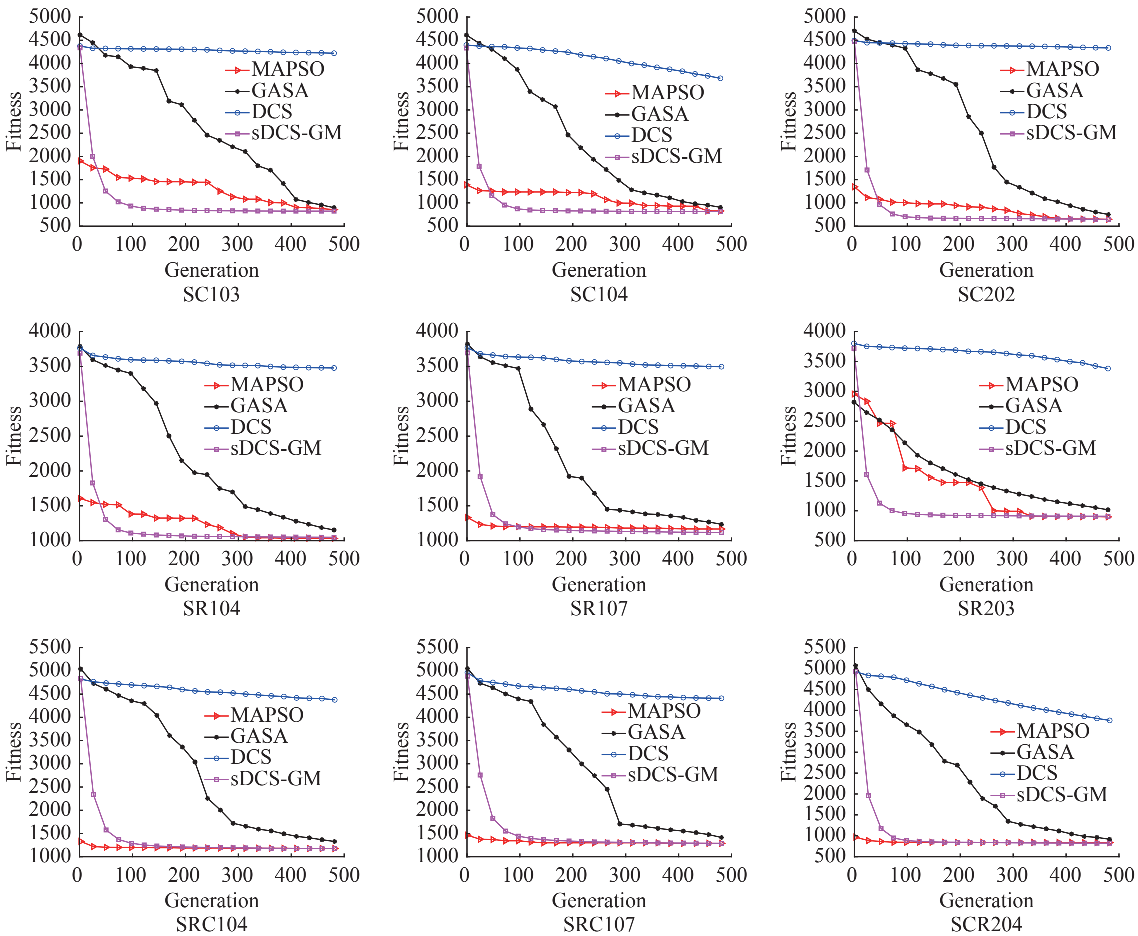

Fig. 5. Comparison of convergence performance of SC103, SC104, SC202, SR104, SR107, SR203, SRC104, SRC107, and SRC204.

In terms of convergence speed, the proposed sDCS-GM algorithm and the GASA algorithm are able to obtain a relatively more ideal driving path in the early iterative stage for the same initial travel, which means that the former has a faster convergence speed. The MAPSO algorithm has difficulty in obtaining a more optimal path in a shorter time despite the shorter initial driving path, which means that the MAPSO algorithm lacks the ability to get rid of the local optimum.

V. Conclusions

This paper designs an SRPTW-SST model to better meet the realistic requirements in real-world scenarios. To solve SRPTW-SST, an improved sDCS-GM algorithm is proposed. In simulation experiments, all of the following strategies were proved to be successful and reasonable. Comparisons with MAPSO and GASA algorithms are made to highlight the superiority and competitiveness of the sDCS-GM algorithm. In addition, comparisons are made with state-of-the-art algorithms such as HSFLA, BA, ACS-BSO, and MODLEM. The experimental results also demonstrate the effectiveness and competitiveness of the proposed sDCS-GM algorithm.

References

[1] Y. Y. Niu, Y. P. Zhang, Z. G. Cao, *et al.*, “MIMOA: A

membrane-inspired multi-objective algorithm for green vehicle routing problem with stochastic demands," *Swarm and Evolutionary Computation*, vol.60, article no.100767, 2021.
[2] Y. N. Sun, B. Xue, M. J. Zhang, *et al.*, "Automatically designing CNN architectures using the genetic algorithm for image classification," *IEEE Transactions on Cybernetics*, vol.50, no.9, pp.3840–3854, 2020.
[3] J. S. Liu, L. Liu, and Y. Li, "A differential evolution flower pollination algorithm with dynamic switch probability," *Chinese Journal of Electronics*, vol.28, no.4, pp.737–747, 2019.
[4] Y. R. Naidu and A. K. Ojha, "Solving multiobjective optimization problems using hybrid cooperative invasive weed optimization with multiple populations," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol.48, no.6, pp.821–832, 2018.
[5] H. N. Xuan, R. C. Zhang, and S. S. Shi, "An efficient cuckoo search algorithm for system-level fault diagnosis," *Chinese Journal of Electronics*, vol.25, no.6, pp.999–1004, 2016.
[6] Y. Marinakis, M. Marinaki, and A. Migdalas, "A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows," *Information Sciences*, vol.481, pp.311–329, 2019.
[7] D. L. Li, Q. Cao, M. Zuo, *et al.*, "Optimization of green fresh food logistics with heterogeneous fleet vehicle route problem by improved genetic algorithm," *Sustainability*, vol.12, no.5, article no.1946, 2020.
[8] M. Mareli and B. Twala, "An adaptive cuckoo search algorithm for optimisation," *Applied Computing and Informatics*, vol.14, no.2, pp.107–115, 2018.
[9] M. Alzaqebah, S. Jawarneh, H. M. Sarim, *et al.*, "Bees algorithm for vehicle routing problems with time windows," *International Journal of Machine Learning and Computing*, vol.8, no.3, pp.236–240, 2018.
[10] J. P. Luo, X. Li, M. R. Chen, *et al.*, "A novel hybrid shuffled frog leaping algorithm for vehicle routing problem with time windows," *Information Sciences*, vol.316, pp.266–292, 2015.
[11] Y. Shen, M. D. Liu, J. Yang, *et al.*, "A hybrid swarm intelligence algorithm for vehicle routing problem with time windows," *IEEE Access*, vol.8, pp.93882–93893, 2020.
[12] B. Moradi, "The new optimization algorithm for the vehicle routing problem with time windows using multi-objective discrete learnable evolution model," *Soft Computing*, vol.24, no.9, pp.6741–6769, 2020.

ZHANG Guoyun received the M.S. degree and Ph.D. degree in control theory and control engineering from Hunan University in 2002 and 2006, respectively. His research interests include image processing, and evolutionary computing. (Email: gyzhang@hnist.edu.cn)

WU Meng received the B.S. degree from the Zunyi Normal University in 2020. He is currently pursuing the M.S. degree from the Hunan Institute of Science and Technology. His research interests include evolutionary computing, and image processing.
(Email: 1794909836@qq.com)

LI Wujing received the B.S. degree in software engineering and Ph.D. degree in computer science and technology from Sichuan University, Chengdu, China, in 2007 and 2012 respectively. His research interests include image denoising, image restoration, image enhancement, and hyperspectral image processing. (Email: liwj@hnist.edu.cn)

OU Xianfeng (corresponding author) received the M.S. degre in communication and information system from Xinjiang University in 2009, the Ph.D. degree in communication and information system from Sichuan University, Chengdu, China, in 2015. His research interests include image processing and evolutionary computing.
(Email: ouxf@hnist.edu.cn)

XIE Wenwu received the B.S., M.S., and Ph.D. degrees in communication engineering from the Huazhong Normal University in 2004, 2007 and 2017, respectively. His research interests include communication algorithm, internet of thing, and evolutionary computing.
(Email: gavinxie@hnist.edu.cn)