A Hybrid BSO-ACS Algorithm for Vehicle Routing Problem with Time Windows on Road Networks

Mingde Liu*, Yang Shen†, Qi Zhao‡ and Yuhui Shi§
*†‡§*Department of Computer Science and Engineering*
Southern University of Science and Technology, Shenzhen, China 518055
Email: *11849245@mail.sustech.edu.cn, †sheny3@mail.sustech.edu.cn,‡zhaoq@sustech.edu.cn, §shiyh@sustech.edu.cn*

***Abstract*—The Vehicle Routing Problem with Time Windows (VRPTW) is NP-hard which has many real-world applications in logistics and transportation. The traditional VRPTW is defined on a complete graph with customers as nodes, but in the real-world, VRPTWs are more based on road networks. To better simulate the real-world scenarios, this paper studies the VRPTW on road networks. Most researchers solve the VRPTW on road networks by utilizing exact algorithms which can not deal with large size problem. In this paper, a hybrid BSO-ACS algorithm, which combines Brain Storm Optimization (BSO), Ant Colony System (ACS) and Local Search (LS), is proposed to solve the VRPTW on road networks. A set of instances based on the road network of southwest Shenzhen, China are generated as benchmark problems. The computational experiments demonstrate the effectiveness of the proposed algorithm.**

***Index Terms*—vehicle routing problem, road network, brain storm optimization, ant colony system, local search**

1. Introduction

The Vehicle Routing Problem (VRP) is an important combinatorial optimization problem that has attracted wide publicity for the last decades. It has many applications in transportation, logistics, and distribution. The VRP is a generic name given to a whole class of problems whose objective is to find optimal routes to service a given set of customers by a fleet of vehicles [1]. There are several variations of traditional VRP, such as Capacitated Vehicle Routing Problem (CVRP) [2], Vehicle Routing Problem with Pickup and Delivery (VRPPD) [3], Vehicle Routing Problem with Time Windows (VRPTW) [4], and Asymmetric cost matrix Vehicle Routing Problem (AVRP) [5], etc. Traditional VRPs are defined on a complete graph that only contains customers and a depot. The customers are usually connected with a straight line, and the distance between two customers is measured by the Euclidean distance. However, in real-world scenarios, VRPs often relies on road networks, which contain not only customers and a depot, but also nodes of the real road. Unlike traditional VRPs, customers are connected by road paths instead of a straight line. Recently, there are many researchers attempt to solve the VRP on road networks [6]–[10]. Most of them use exact algorithms, which can not solve large-scale problems in an acceptable time.

Solving the VRP on road networks includes two phases: path selection and vehicle routing [9]. In the path selection, the Dijkstra's algorithm is usually used to find the path among customers. While in the vehicle routing, like traditional VRP algorithms, there are two main categories of approaches: exact algorithms and heuristic ones. Exact algorithms include: Column Generation (CG) [7], Mixed Integer Linear Programming (MILP) [9], Branch-and-Price (BP) [11], etc. These methods can get an optimal solution only on small-sized problems. With the increase in the size of the problem, the computational time of exact algorithms grows exponentially. To get a satisfying solution within an acceptable computational time for large-scale problems, heuristic algorithms are usually employed. VRP heuristic algorithms include two main categories, i.e., constructive heuristics and improvement heuristics. Constructive heuristics construct a feasible solution as much as possible. The most popular ones include saving algorithm [12], cluster-first route-second algorithm [13], nearest neighbor algorithm [14], etc. The improvement heuristics enhance the current solution by modifying solutions. It includes 2-opt [15], 3-opt [16], chain-exchange [17], cross and relocate [18], λ-exchange [19], etc. There are also many metaheuristics that are used to solve VRPs, like population based metaheuristics, Genetic Algorithms (GA) [20] and Ant Colony Systems (ACS) [21]. Other techniques are based on local search approaches, such as Tabu Search (TS) [6], Simulated Annealing (SA) [22], Adaptive Large Neighborhood Search (ALNS) [10], etc.

There are many comparison studies [23]–[25] that have analyzed the performance of different heuristic and metaheuristic algorithms for VRP. Their results showed that no single heuristic or metaheuristic could

Corresponding author: Yuhui Shi (e-mail: shiyh@sustech.edu.cn).

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

be better than others at any time, but the hybridization of different algorithms can overcome the shortcoming of a single method. In this paper, a hybrid BSO-ACS algorithm to solve the VRPTW on road networks is proposed. At first, Dijkstra's algorithm is employed in path selection to obtain the shortest path between two customers. Then we use a saving heuristic algorithm to initialize solutions. The core of the proposed algorithm is BSO, which contains three phases: 1) Clustering; 2) Generating new solutions; 3) Disrupting solutions. In phase 2), the ACS is performed to generate a new solution from parents like the combination operation in the memetic algorithm. In phase 3), a local search is used to improve individuals like local improvement in the memetic algorithm. There are some advantages in this hybrid algorithm: 1) The clustering operator of BSO can help ACS learn more similarity properties in one cluster, which accelerates the exploitation of ACS. 2) Since the ACS is easy to fall into local optimum by using deposited pheromones, a set of local search heuristics is used to disrupt the solution to avoid local optimum.

The main contributions of this paper are:

- A hybrid BSO-ACS algorithm is proposed, in which ACS is used to help BSO to combine advantages of solutions in one cluster, and local search is used to disrupt solutions to better explore the search space and avoid local optimum;
- 30 instances of VRPTW on real road networks from Shenzhen, China are adopted for experimental study, with the results demonstrate the better performance of the proposed algorithm than the compared approaches.

The remainder of this paper is organized as follows. Section 2 presents the definition of the problem. Section 3 firstly describes BSO, ACS, and Local Search which includes 2-opt, exchange and relocate neighborhoods, then proposes a hybrid BSO-ACS algorithm. Section 4 evaluates the performance of the proposed algorithm. Conclusions are provided in Section 5.

2. Problem description

2.1. Standard VRPTW

VRPTW is an important generalization of the VRP. It is a combinatorial optimization problem in logistics. The VRPTW can be formally defined as follows. Firstly, there is a directed complete graph $G(V, E)$, where $V = \{v_0, v_1, \ldots, v_n\}$ is the vertex set to represent the depot v_0 and customers $\{v_1, v_2, \ldots, v_n\}$, and each vertices v_i located at coordinates (x_i, y_i); $E = \{(v_i, v_j)|v_i, v_j \in V, i \neq j\}$ is the edge set, the edge is normally just a straight line between two vertices, and the cost $cost_{ij}$ in edge (v_i, v_j) equals to the Euclidean distance between v_i and v_j. The speed of the vehicle is often assumed to be one unit without loss of generality, so the time cost equals to the Euclidean distance between v_i and v_j. Each customer has a demand q_i, a time window $[e_i, l_i]$, and each vehicle has a capacity Q. The vehicle should arrive at customer i within the time window $[e_i, l_i]$, and the vehicles' capacity should be satisfied. The objective of VRPTW is to minimize the total distance cost of all vehicles.

2.2. VRPTW on road networks

The standard VRPTW assumes customers on a complete graph and sets the cost to euclidean distance. However, in the real world, most VRPTWs take place on road networks, and customers are connected by road paths instead of a straight line. Some researchers [7], [9], [11] focus on the VRPTW on road networks to be more suitable for practical application. The VRPTW on road networks can be formally defined as follows. First, there is a road network graph $G(V, C, E)$, where $V = \{v_0, v_1, \ldots, v_m\}$ is the vertex set to represent the depot v_0 and road nodes $\{v_1, v_2, \ldots, v_m\}$, the customer set $C = \{c_1, c_2, \ldots, c_n\}$ is the subset of the vertex set V. $E = \{(i,j)^p; i,j \in V, i \neq j, p = 1, \ldots, |E_{(i,j)}|\}$ is the path set representing the alternative paths between vertex i and j. The distance between the two vertices is the path length, and the cost time equals the distance divided by vehicle's speed, The mathematical model of VRPTW on road networks is defined as follows [11].

Parameters description:

K	the set of all vehicles
V	the set of all vertices
C	the set of all customers
E	the set of all paths
N	total number of customers
Q	maximum vehicle capacity
$cost_{ij}$	cost from vertex i to vertex j
t_{ij}	travel time from vertex i to vertex j
q_i	demand of customer c_i
e_i	earliest arrival of the time window of customer c_i
l_i	latest arrival of the time window of customer c_i
s_i	service time of customer c_i
t_i	arrival time of vertex v_i
t'_i	leave time of vertex v_i
w_i	wait time of customer c_i

Objective function:

$$\min \sum_{k \in K} \sum_{(i,j)^p \in E} cost_{ij} x_{(i,j)^p k} \tag{1}$$

subject to:

$$x_{(i,j)^p k} = \begin{cases} 1 & \text{if vehicle } k \text{ uses path } (i,j)^p \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\sum_{i \in V} \sum_{p \in E_{(i,0)}} x_{(i,0)^p k} = \sum_{j \in V} \sum_{p \in E_{(0,j)}} x_{(0,j)^p k} = 1 \quad (\forall k \in K) \tag{3}$$

$$\sum_{k\in K}\sum_{i\in C} x_{(i,j)^p k} = 1 \quad (\forall j \in C) \tag{4}$$

$$\sum_{i\in C} q_i \sum_{j\in C, j\neq i} x_{(i,j)^p k} \leq Q \quad (\forall k \in K) \tag{5}$$

$$t_i' = t_i + w_i + s_i \tag{6}$$

$$w_j = \begin{cases} \max\{e_j - t_i' - t_{ij},\ 0\} & \text{if } j \in C \\ 0 & \text{if } j \notin C \end{cases} \quad (\forall i, j \in V) \tag{7}$$

$$t_i + s_i + t_{ij} + w_i \leq t_j \quad (\forall i \in V, j \in C, i \neq j) \tag{8}$$

$$e_i \leq t_i + w_i \leq l_i \quad (\forall i \in C) \tag{9}$$

The objective function (1) minimizes the total cost of all routes. Equation (2) shows a binary variable to represent whether a vehicle uses the path $(i,j)^p$. Equation (3) means the vehicle should start from and end in the depot. Equation (4) represents that each customers should only be serviced once by vehicles. Constraint (5) implies that the vehicle capacity should be satisfied. Constraint (8) is the travel time relationship between vertex v_i and its successor c_j, and Inequality (9) means the vehicle should arrive at customer c_i within time window $[e_i, l_i]$.

3. Hybrid BSO-ACS Algorithm

This section describes the proposed algorithm for the VRPTW on road networks. Firstly we briefly introduce the BSO algorithm, local search heuristics used in the proposed algorithm, and the ACS algorithm. Then, a hybrid BSO-ACS algorithm is proposed to solve the VRPTW on road networks.

3.1. Brain Storm Optimization

Brain Storm Optimization (BSO) [26] is firstly proposed by Shi, which belongs to swarm intelligence algorithms [27]. It is based on the brainstorming process and has been used to solve traditional VRPs [28], [29]. The main idea of this algorithm is to divide solutions into clusters that the same cluster has similar properties. In the new solutions generation procedure, the cluster combines with each other to enhance the solution quality, while the disruption of one cluster's solutions help the algorithm search more space to avoid stuck into local optimum. The procedure of BSO is shown in Fig. 1.

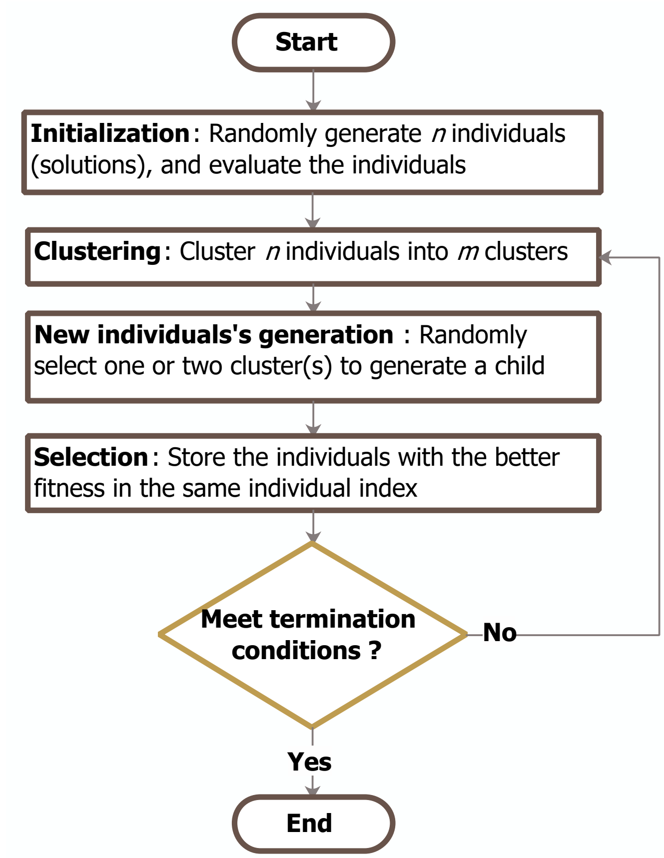


Figure 1. BSO Procedure

3.2. Local Search

Local search is a widely used method to solve VRPs. It starts from the current solution s, and moves to another solution in its neighborhoods $N(s)$. This section will introduce some classical neighborhood strategies.

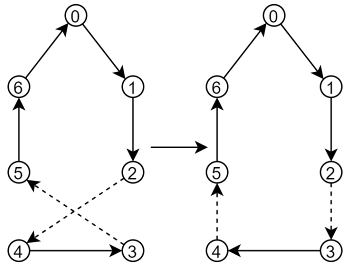
3.2.1. 2-opt. 2-opt [15] is a widely used neighborhood approach first proposed by Croes for solving TSP (Travelling Salesman Problem). The main idea is to remove the intersecting edges in the routes and reverse the nodes between these edges. In VRPs, the 2-opt has two different operations for the single route or two routes, these two operations are shown in Fig. 2.

3.2.2. Exchange and Relocate. The exchange and relocated neighborhoods are firstly proposed by Savelsobergh [18]. In these neighborhoods, the sub-routes between routes may exchange and relocate, which are shown in Fig. 3.

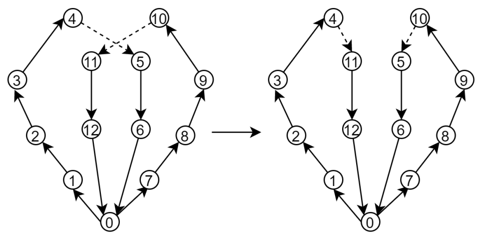
3.3. Ant Colony System

Ant Colony Optimization (ACO) [30] is a swarm intelligence algorithm. It is inspired by the behavior of ants searching the shortest path to get food by pheromones. When searching food, firstly ants wander randomly and leave pheromones when a route is found. Shorter routes have more pheromones and are more attractive for ants to select since ants travel more frequently between food and nest.

When applying ACO to solve VRRs, ants select the next customer one by one from the depot until it can

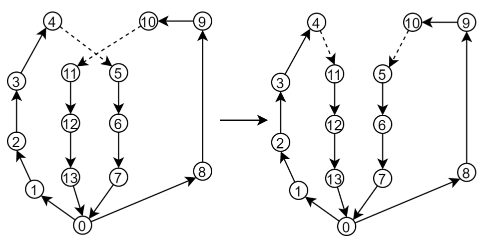


(a) Intra-route 2-opt

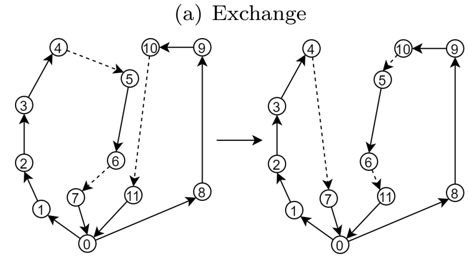


(b) Inter-route 2-opt

Figure 2. 2-opt



(a) Exchange



(b) Relocate

Figure 3. Exchange and Relocate

not select a feasible customer. Then, the ACO generates a route and restarts from the depot to generate a new route for another vehicle [31]. For an ant located in the node of customer i, its probability of selecting customer j is Eq. (10):

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij}^{\alpha})\cdot(\eta_{ij}^{\beta})}{\sum_{k\in J_k(i)}(\tau_{ik}^{\alpha})\cdot(\eta_{ik}^{\beta})} & \text{if } j \in J_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where τ_{ij} is the amount of pheromone deposited on the path from customer i to j, $0 \leq \alpha$ is a parameter which controls the relative influence of τ_{ij}, η_{ij} is the desirability of the state transition (typically is $1/d_{ij}$, i.e., the reciprocal of the distance between i and j), $\beta \geq 1$ is a parameter to control the influence of η_{ij}, and $J_K(i)$ is a customer set which can be traveled by ant k from customer i.

When all ants have found their solutions, pheromones along the edges are updated by the performance of solutions.

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_k \Delta\tau_{ij}^k \tag{11}$$

where $0 \leq \rho \leq 1$ is the pheromone evaporation coefficient, and $\Delta\tau_{ij}^k$ is the amount of pheromone deposited by ant k, which is defined by

$$\Delta\tau_{ij}^k = \begin{cases} 1/L_k & \text{if ant } k \text{ uses edge } (i,j) \text{ in its solution} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where L_k is the cost (usually distance cost) of the solution found by ant k.

Since ACO always uses the probability of p_{ij}, it can not balance the exploration and exploitation. To handles this issue, the Ant Colony System (ACS) [32] was proposed by Dorigo. Its state transition rule is defined

$$s = \begin{cases} \arg\max_{j\in J_k(i)} \tau_{ij}\cdot\eta_{ij}^{\beta} & \text{if } q \leq q_0 \\ \frac{\tau_{ij}\cdot(\eta_{ij}^{\beta})}{\sum_{k\in J_k(i)}(\tau_{ik}^{\alpha})\cdot(\eta_{ik}^{\beta})} & \text{otherwise} \end{cases} \tag{13}$$

where $0 \leq q \leq 1$ is a random number, $0 \leq q_0$ is a self-defined parameter which balance exploration and exploitation.

ACS includes local and global pheromones. When a solution is found by an ant, it is updated by the local update Eq. (14) to refine the edges' pheromone; when a global best solution is found by ants, the τ_{ij} of edge (i,j) which is used by the best solution will be updated by Eq. (15).

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij} \tag{14}$$

$$\tau_{ij} \leftarrow (1-\alpha)\cdot\tau_{ij} + \alpha\cdot\ L_{ij}^* \tag{15}$$

where α is the pheromone evaporation rate, and L_{ij}^* is the length of the best solution.

3.4. Proposed Hybrid BSO-ACS Algorithm

To solve VRPTWs on road networks, we propose a hybrid BSO-ACS algorithm. In this algorithm, the BSO clustering procedure can help the ACS combine similarity properties which improves exploitation, and a local search expands more search space to avoid stuck into local optimal. In the proposed hybrid BSO-ACS algorithm, we first use the Savings method [12] to generate initial solutions. In the clustering stage of BSO, to make the process more efficient, we conduct the clustering operation in the object space [33] with the number of clusters set to two, i.e., better individuals, "elitists" and worse individuals "normal". In the new solution generation stage, the ACS is used to generate a new individual from two original clusters, and the local search is used to generate a new individual from one original cluster. More details are given in Alg. 1.

Algorithm 1 Hybrid BSO-ACS Algorithm

Input: $G(V, C, E)$
Output: s_{best}, best solution
use the saving method to initialize n solutions
while not termination **do**
  evaluate solutions S and sort solutions by fitness
  divide solutions into two clusters by fitness, set the top $perc_e$ percentage as elitists and remaining as normals
  if $rand(0,1) < p_e$ **then**
    if $rand(0,1) < p_{one}$ **then**
      randomly pick a solution s_i from elitists
      $s' \leftarrow$ local_search(s_i)
    else
      randomly pick two solutions s_i and s_j from elitists
      $s' \leftarrow$ ACS(s_i, s_j)
    end if
  else
    if $rand(0,1) < p_{one}$ **then**
      randomly pick a solution s_i from normals
      $s' \leftarrow$ local_search(s_i)
    else
      randomly pick two solutions s_i and s_j from normals
      $s' \leftarrow$ ACS(s_i, s_j)
    end if
  end if
  if s' is better than the selected solution(s) s **then**
    $s \leftarrow s'$
  end if
end while
return s_{best} in solutions S

4. Experiments and Discussions

We use the road network in Nanshan district, southwest of Shenzhen, China from OpenStreetMap. The road network graph has 8214 nodes and 18558 road arcs. We generate 30 instances like the Solomon benchmark, which contains C1, C2, R1, R2, RC1, RC2, and all instances have 100 customers, respectively. Customers in "C" problems are clustered, and customers in "R" problems are random. In "RC" problems customers are partially clustered and partially randomly distributed. In problem sets 1 (i.e., R1, C1, and RC1), the scheduling horizon is short because of the low capacity of vehicles. In contrast, problem sets R2, C2 and RC2 have longer scheduling horizons. More instances can be obtained at https://github.com/lmingde/VRPTW_on_Road_Network_Benchmark.

To evaluate the performance of the proposed algorithm, we compare the proposed algorithm with ACS and ALNS in the total distances. The ALNS is a popular algorithm, and have shown its promise on various problems [10], [34]–[37].

4.1. Experiment Setup

The proposed algorithm is implemented in the Python programming language, and all experiments are conducted on an Intel Xeon E5-2650 CPU@2.30GHz PC with 16GB RAM.

There are some parameters in the hybrid algorithm. The parameters of BSO are $per_e,\ p_e,\ p_{one}$. A small per_e value with a large p_e value, conduct the algorithm to search in the neighborhood of elitists. This operation can facilitate exploitation, but may result in falling into local optimum. The p_{one} is used to control the strength of disruption, which enhances the exploration. The parameters of ACS are $M,\ \beta,\ \rho,\ q_0$. Large β and q_0 indicate greed strategy. When ρ is small, the exploration of the algorithm will be reduced. When ρ is a large value, the exploitation of the algorithm will be enhanced. The parameter setting is given in Table 1. All parameters are selected by tuning to balance the trade-off between computational cost and solution quality.

TABLE 1. Parameters of the Hybrid Algorithm

parameter	value	parameter	value
M	20	$perc_e$	0.1
β	2	p_e	0.2
q_0	0.1	p_{one}	0.6
ρ	0.1	max_iter	150

4.2. Result Analysis

The comparison results for all instances are given in Table 2, where "NV" represents the number of the used vehicles, "TD" represents the total distances. We use the gap metric to represent the cost reduction between our method and others, the gap is computed according to Eq. (16).

$$Gap = \frac{TD_{ours} - TD_{others}}{TD_{ours}} \tag{16}$$

As it can be observed from Table 2, the average number of vehicles obtained by the proposed algorithm is 10.33, which is better than ACS (10.83) and ALNS (10.37). The average total distance obtained by the proposed algorithm is 223648.04, which is superior to the ACS (249310.02) and ALNS (228705.44). For all instances, the total distance obtained by the hybrid BSO-ACS algorithm is smaller than ACS, and the gap ranges from -18.36% to -2.23%. The total distances obtained by the hybrid BSO-ACS algorithm are reduced by -9.13% to 1.13% comparing to ALNS. Counting the number of the best solutions in the three methods, the hybrid-BSO-ACS gets best in 25 out of 30 cases, while the ALNS obtains the best in the remaining five instances. There are some results which attained by hybrid algorithm are shown in Fig. 4, where the black point is the depot and the grey points are customers.

(a) C

(b) R

(c) RC

Figure 4. Results of the Hybrid Algorithm

Table. 3 summarizes the average cost and gap in each problem set. The experiments were run 10 times in each instance. In Table 3, compared with ACS, the hybrid BSO-ACS obtains solution results smaller -3.68%, -12.49%, -12.88%, -13.94%, -9.57% and -15.17% for C1, C2, R1, R2, RC1 and RC2. Gaps between hybrid BSO-ACS and ALNS are -0.91%, -3.77%, -2.24%, -4.06%, -0.52% and -3.39% for C1, C2, R1, R2, RC1, and RC2. It can be observed that gaps in C2, R2, and RC2 are bigger than that in C1, R1, and RC1. The reason is that the scheduling horizon is longer in C2, R2, RC2, i.e., the average length of routes are longer since the capacity of vehicles is larger. Local search in the proposed hybrid BSO-ACS algorithm generates a new solution from one single solution that is effective to deal with such long route cases.

5. Conclusions

This paper proposed a hybrid BSO-ACS algorithm to solve the VRPTW on road networks. In the proposed algorithm, the BSO uses Local Search and ACS to generate new solutions, and the solutions are improved iteratively. 30 instances based on the road network on the southwest of Shenzhen, China with 100 customers are generated. Among 30 instances, the proposed algorithm achieved 25 out of 30 better solutions than classic ACS and ALNS algorithm. The experimental results demonstrated the effectiveness of the proposed algorithm.

Acknowledgment

This work is partially supported by National Science Foundation of China under grant number

TABLE 2. EXPERIMENTAL RESULTS FOR 30 INSTANCES

Instance	ACS		ALNS		Ours		ACS-Gap (%)	ALNS-Gap (%)
	NV	TD	NV	TD	NV	TD		
C101	10	178642.29	10	174643.13	10	**169510.09**	-5.39	-3.03
C102	10	164509.60	10	161987.45	10	**160922.73**	-2.23	-0.66
C103	9	159449.60	9	153394.83	9	**152799.33**	-4.35	-0.39
C104	11	199893.27	11	**193817.15**	11	194192.82	-2.94	0.19
C105	10	176443.25	10	171630.46	10	**170328.77**	-3.59	-0.76
C201	4	86760.81	4	79319.48	4	**76069.36**	-14.05	-4.27
C202	4	92290.71	4	83867.42	4	**80522.11**	-14.62	-4.15
C203	4	100014.34	4	91633.68	4	**91436.24**	-9.38	-0.22
C204	5	102152.46	5	96807.01	5	**88708.24**	-15.16	-9.13
C205	4	99525.90	4	91871.27	4	**90636.40**	-9.81	-1.36
R101	19	350604.44	17	330939.79	17	**317895.53**	-10.29	-4.10
R102	19	357345.61	17	320417.29	17	**315818.89**	-13.15	-1.46
R103	18	362358.72	17	329838.13	17	**319660.57**	-13.36	-3.18
R104	18	350986.04	16	316346.54	16	**308107.30**	-13.92	-2.67
R105	20	370838.08	18	**325680.74**	18	326149.74	-13.70	0.14
R201	7	250917.82	7	225081.07	7	**218971.18**	-14.59	-2.79
R202	8	279107.42	8	256822.71	8	**250436.08**	-11.45	-2.55
R203	7	241906.20	7	227153.38	7	**214950.93**	-12.54	-5.68
R204	7	251382.56	7	224213.37	7	**217326.73**	-15.67	-3.17
R205	7	254304.98	7	233498.18	7	**219593.73**	-15.81	-6.33
RC101	19	360650.38	17	320068.20	17	**317129.43**	-13.72	-0.93
RC102	19	410942.97	18	376564.69	18	**370533.78**	-10.91	-1.63
RC103	17	340871.83	16	**314626.59**	16	318156.02	-7.14	1.11
RC104	16	329499.95	15	**311177.73**	15	314723.80	-4.69	1.13
RC105	15	297460.63	15	273315.12	14	**267015.35**	-11.40	-2.36
RC201	8	279643.79	8	243028.61	8	**240960.97**	-16.05	-0.86
RC202	7	266227.40	7	247579.37	7	**237674.49**	-12.01	-4.17
RC203	7	235814.73	7	207468.23	7	**204255.57**	-15.45	-1.57
RC204	8	254073.69	8	230622.72	8	**214666.03**	-18.36	-7.43
RC205	8	274681.04	8	247748.97	8	**240289.13**	-14.31	-3.10
Avg.	10.83	249310.02	10.37	228705.44	10.33	223648.04	-11.33	-2.51

TABLE 3. AVERAGE TOTAL DISTANCES

Problem Set	ACS	ALNS	Ours	ACS-Gap(%)	ALNS-Gap(%)
C1	175787.6	171094.6	169550.8	-3.68	-0.91
C2	96148.84	88699.77	85474.47	-12.49	-3.77
R1	358426.6	324644.5	317526.4	-12.88	-2.24
R2	255523.8	233353.7	224255.7	-13.94	-4.06
RC1	347885.2	319150.5	317511.7	-9.57	-0.52
RC2	262088.1	235289.6	227569.2	-15.17	-3.39

61761136008, Shenzhen Peacock Plan under Grant No. KQTD2016112514355531, Program for Guangdong Introducing Innovative and Entrepreneurial Teams under grant number 2017ZT07X386.

References

[1] P. Toth and D. Vigo, *Vehicle routing: problems, methods, and applications.* SIAM, 2014.

[2] F. Li, B. Golden, and E. Wasil, "Very large-scale vehicle routing: new test problems, algorithms, and results," *Computers & Operations Research*, vol. 32, no. 5, pp. 1165–1179, 2005.

[3] Y. Dumas, J. Desrosiers, and F. Soumis, "The pickup and delivery problem with time windows," *European Journal of Operational Research*, vol. 54, no. 1, pp. 7–22, 1991.

[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] P. Toth and D. Vigo, "Models, relaxations and exact approaches for the capacitated vehicle routing problem," *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 487–512, 2002.

[6] R. Eglese, W. Maden, and A. Slater, "A road timetabletm to aid vehicle routing and scheduling," *Computers & Operations Research*, vol. 33, no. 12, pp. 3508–3519, 2006.

[7] A. N. Letchford, S. D. Nasiri, and A. Oukil, "Pricing routines for vehicle routing with time windows on road networkss," *Computers & Operations Research*, vol. 51, pp. 331–337, 2014.

[8] D. S. Lai, O. C. Demirag, and J. M. Leung, "A tabu search heuristic for the heterogeneous vehicle routing problem on a multigraph," *Transportation Research Part E: Logistics and Transportation Review*, vol. 86, pp. 32–52, 2016.

[9] Y. Huang, L. Zhao, T. Van Woensel, and J.-P. Gross, "Time-dependent vehicle routing problem with path flexibility," *Transportation Research Part B: Methodological*, vol. 95, pp. 169–195, 2017.

[10] H. B. Ticha, N. Absi, D. Feillet, and A. Quilliot, "Multigraph modeling and adaptive large neighborhood search for the vehicle routing problem with time windows," *Computers & Operations Research*, vol. 104, pp. 113–126, 2019.

[11] ——, "Empirical analysis for the vrptw with a multigraph representation for the road network," *Computers & Operations Research*, vol. 88, pp. 103–116, 2017.

[12] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.

[13] M. L. Fisher and R. Jaikumar, "A generalized assignment heuristic for vehicle routing," *Networks*, vol. 11, no. 2, pp. 109–124, 1981.

[14] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, II, "An analysis of several heuristics for the traveling salesman problem," *SIAM Journal on Computing*, vol. 6, no. 3, pp. 563–581, 1977.

[15] G. A. Croes, "A method for solving traveling-salesman problems," *Operations Research*, vol. 6, no. 6, pp. 791–812, 1958.

[16] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[17] R. Fahrion and M. Wrede, "On a principle of chain-exchange for vehicle-routeing problems (1-vrp)," *Journal of the Operational Research Society*, vol. 41, no. 9, pp. 821–827, 1990.

[18] M. W. Savelsbergh, "The vehicle routing problem with time windows: Minimizing route duration," *ORSA Journal on Computing*, vol. 4, no. 2, pp. 146–154, 1992.

[19] I. H. Osman and N. Christofides, "Capacitated clustering problems by hybrid simulated annealing and tabu search," *International Transactions in Operational Research*, vol. 1, no. 3, pp. 317–336, 1994.

[20] E. Alba and B. Dorronsoro, "Solving the vehicle routing problem by using cellular genetic algorithms," in *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer, 2004, pp. 11–20.

[21] M. Reimann, K. Doerner, and R. F. Hartl, "D-ants: Savings based ants divide and conquer the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 4, pp. 563–591, 2004.

[22] W.-C. Chiang and R. A. Russell, "Simulated annealing metaheuristics for the vehicle routing problem with time windows," *Annals of Operations Research*, vol. 63, no. 1, pp. 3–27, 1996.

[23] G. Laporte, M. Gendreau, J.-Y. Potvin, and F. Semet, "Classical and modern heuristics for the vehicle routing problem," *International Transactions in Operational Research*, vol. 7, no. 4-5, pp. 285–300, 2000.

[24] A. Van Breedam, "Comparing descent heuristics and metaheuristics for the vehicle routing problem," *Computers & Operations Research*, vol. 28, no. 4, pp. 289–315, 2001.

[25] C. Gunes, W.-J. van Hoeve, and S. Tayur, "Vehicle routing for food rescue programs: A comparison of different approaches," in *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*. Springer, 2010, pp. 176–180.

[26] Y. Shi, "Brain storm optimization algorithm," in *International Conference in Swarm Intelligence*. Springer, 2011, pp. 303–309.

[27] S. Cheng, Q. Qin, J. Chen, and Y. Shi, "Brain storm optimization algorithm: a review," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 445–458, 2016.

[28] L. Wu, Z. He, Y. Chen, D. Wu, and J. Cui, "Brainstorming-based ant colony optimization for vehicle routing with soft time windows," *IEEE Access*, vol. 7, pp. 19 643–19 652, 2019.

[29] L. Ke, "A brain storm optimization approach for the cumulative capacitated vehicle routing problem," *Memetic Computing*, vol. 10, no. 4, pp. 411–421, 2018.

[30] M. Dorigo, "Optimization, learning and natural algorithms," *PhD Thesis, Politecnico di Milano*, 1992.

[31] M. Schyns, "An ant colony system for responsive dynamic vehicle routing," *European Journal of Operational Research*, vol. 245, no. 3, pp. 704–718, 2015.

[32] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[33] Y. Shi, "Brain storm optimization algorithm in objective space," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 1227–1234.

[34] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation Science*, vol. 40, no. 4, pp. 455–472, 2006.

[35] S. Chen, R. Chen, G.-G. Wang, J. Gao, and A. K. Sangaiah, "An adaptive large neighborhood search heuristic for dynamic vehicle routing problems," *Computers & Electrical Engineering*, vol. 67, pp. 596–607, 2018.

[36] G. M. Ribeiro and G. Laporte, "An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem," *Computers & Operations Research*, vol. 39, no. 3, pp. 728–735, 2012.

[37] I. Dayarian, T. G. Crainic, M. Gendreau, and W. Rei, "An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 95, pp. 95–123, 2016.