

A Dynamic Space Reduction Ant Colony Optimization for Capacitated Vehicle Routing Problem

Jinsi Cai

Northwestern Polytechnical University

Peng Wang (✉ wangpeng305@nwpu.edu.cn)

Northwestern Polytechnical University <https://orcid.org/0000-0002-8745-320X>

Siqing Sun

Huazhong University of Science and Technology

Huachao Dong

Northwestern Polytechnical University

Research Article

Keywords: Ant colony optimization, capacitated vehicle routing problem, discrete combinatorial optimization problem, evolutionary algorithm

Posted Date: February 15th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1344589/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A Dynamic Space Reduction Ant Colony Optimization for Capacitated Vehicle Routing Problem

Jinsi Cai ^a, Peng Wang ^{a,*}, Siqing Sun ^b, and Huachao Dong ^a

^a School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China

^b School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China
(Corresponding author: Peng Wang, E-mail: wangpeng305@nwpu.edu.cn).

Abstract

As a typical meta-heuristic algorithm, ant colony optimization (ACO) has achieved good results in solving discrete combinatorial optimization problems. However, it suffers from poor solutions and the drawback of easily being trapped in local optima. This paper presents a new type of ACO called "dynamic space reduction ant colony optimization" (DSRACO) to solve the capacitated vehicle routing problem, which is a typical nondeterministic polynomial-hard optimization problem. In DSRACO, ACO is integrated with a unique dynamic space reduction method, an elite enhanced mechanism, and large-scale neighborhood search methods to improve the quality of the solution. The performance of DSRACO is evaluated using 73 well-known benchmark instances in comparison to ACO and three other cutting-edge algorithms. The experimental results show that DSRACO can solve CVRP with a satisfactory result.

Keywords Ant colony optimization, capacitated vehicle routing problem, discrete combinatorial optimization problem, evolutionary algorithm.

1. Introduction

Modern supply chains require the transportation of commodities in a variety of settings, including inside a city, region, country, or even throughout the world. Realizing the importance of this respect, a lot of researchers devote themselves to finding optimal and novel solutions for more efficient transportation and vehicle routing problems are put forward under such circumstances.

Vehicle Routing Problem (VRP) is one of the fundamental problems in operational research and physical distribution. The typical VRP and its variants, including Capacitated VRP, Multi-depot VRP, Dynamic VRP, and VRP with Time Window are extensively studied and arouse great interest among researchers. VRP is a typical Nondeterministic Polynomial-hard (NP-hard) optimization problem [1] and the study of VRP effectively promotes the development of evolutionary computation and discrete combinatorial optimization.

The formulation of capacitated vehicle routing problem (CVRP) was proposed by Dantzig and Ramser [2]. CVRP entails meeting the needs of " n " geographically dispersed customers from a central depot using a fleet of available vehicles with identical capacity at the lowest possible cost. The main goal of CVRP is to minimize the total distance (TD). The constraints commonly considered for CVRP are: (i) total demand of all the customers served on each route must be less than or equal to the total capacity of each vehicle, (ii) each customer is served by a single vehicle only once, and (iii) all the vehicles must start from and return to the same depot.

Common methods for solving VRP are exact methods and meta-heuristic algorithms. Exact methods, such as the branch and cut algorithm, which uses cutting planes to tighten linear programming relaxations, are only appropriate for small-scale VRP problems [3][4]. As the size of the problem grows larger, exact methods take an excessive amount of time to find the best solution. On the contrary, meta-heuristic algorithms are effective to solve various discrete combinatorial optimization problems in a reasonable time limits [5][6][7][8][9].

Ant colony optimization is a nature-inspired swarm intelligence algorithm and was first proposed by Dorigo and his colleagues in the early 1990's [10][11] to forage the behavior of real ants in search of food. When searching for food, ants release pheromones on their path and can communicate via them for decision making. Paths with higher pheromone content will be selected by more ants. Due to evaporation, the pheromone on the path will gradually disappear over time, which motivates ants to explore more paths. Unlike other meta-heuristic algorithms such as particle swarm optimization (PSO) and differential evolution algorithm (DE), the pheromone mechanism of ant colony algorithm (ACO) can more effectively combine probability construction with total distance, making it more suitable for solving VRP.

Wang et al. [12] propose a novel ACO algorithm to solve CVRP. The proposed algorithm allows ants to go in and out of the depots more than once until they have visited all customers, which simplifies the procedure of constructing feasible solutions. Zhang et al. [13] propose a solution strategy based on ant colony optimization and three mutation operators (2-Opt, shift, and inverse), which incorporates the concept of Pareto optimality for multi-objective optimization. Lin et al. [14] put forward an effective order-aware hybrid genetic algorithm for CVRP. The method is characterized by an improved initialization strategy and a problem-specific crossover operator. The former combines the sweep algorithm, while the latter integrates neighborhood search heuristics to find the offspring with the best fitness and check constraints simultaneously. Szeto et al. [15] present a variant of the artificial bee colony (ABC) algorithm for VRP. The proposed algorithm is integrated with 7 neighborhood search methods to improve the solution, but its neighborhood search mechanism is still relatively simple and the search range is small. Kao et al. [16] hybridize ACS and PSO to solve VRP. To avoid the stagnation of pheromone, they propose a pheromone disturbance approach.

Altabeeb et al. [17] propose an improved hybrid firefly algorithm for CVRP. The algorithm is integrated with types of local search (2-Opt and 2-h-Opt) and genetic operators to enhance the quality of the solution and accelerate the convergence. Goel and Maini [18] present a hybrid of the ant colony and firefly algorithms (HAFA) to solve CVRP and they test the algorithm both on CVRP and VRPTW. Akpinar [19] proposes a new hybrid algorithm that executes large neighborhood search (removal and insertion operators) methods in combination with the solution construction mechanism of the ant colony optimization for capacitated vehicle routing problem. Mahdi and Shokouhi [20] put forward a hybrid algorithm composed of adaptive large neighborhood search and variable neighborhood search. The proposed algorithm aims to solve the multi-depot multi-compartment vehicle routing problem and the neighborhood search methods include one-move, two-move, one-exchange, two-exchange, product spilt, and product aggregation. Juan et al. [21] propose a hybrid algorithm that combines a CVRP classical heuristic with Monte Carlo simulation using state-of-the-art random number generators, but the experimental results of the proposed algorithm are not good enough. With the goal of decreasing the overall distance, Amous et al. [22] provide a variable neighborhood search (VNS) algorithm for CVRP, however its neighborhood search mechanism only incorporates simple insertion and swaps.

To solve the drawbacks of poor solutions and local optima of ant colony optimization, this paper presents an improved algorithm, termed "dynamic space reduction ant colony optimization" (DSRACO). The main contributions of our work can be described as follows:

Dynamic space reduction method. In the beginning, the ant colony can freely explore the whole solution space to avoid local optima. With the iteration going on, the next destination of the ant colony is fixed on several cities nearest to improve the quality of the solution.

Elite enhanced mechanism. To the best of our knowledge, most of the elite methods applied to ant colony optimization are single-elite, and no one has studied the number of elites in detail. In this paper, a method of selecting several individuals as elites is proposed and a detailed corresponding parameter analysis is given.

Large-scale neighborhood search operators. As mentioned above, most neighborhood methods include only 2-Opt, 2-h-Opt, and so on. Its search range is relatively small and cannot effectively explore the solution space. This paper proposes a large-scale neighborhood search method to improve the quality of the solution.

This paper is organized as follows: In Section 2, a brief introduction to VRP is given and the mathematical model of CVRP is formulated. The proposed algorithm (DSRACO) is elaborated in Section 3. The detailed parameter settings and relevant analysis are given in Section 4. Section 5 details the performance of DSRACO and compares it to other algorithms. Section 6 discusses the paper's conclusion based on the current study.

2. Mathematical modeling of capacitated vehicle routing problem

Capacitated Vehicle Routing Problem is one of the most widely studied typical NP-Hard combinatorial optimization problems, and it can be described as a graph $G = (N, E)$ with a set of nodes $N = \{0, 1, \dots, n\}$ where node 0 represents the depot and $\{1, 2, \dots, n\}$ represents the customers. Each edge $\{i, j\} \in E$ has a non-negative cost C_{ij} . Each customer $i \in V' = V \setminus \{0\}$ has a specific demand q_i ($i = 1, 2, \dots, n$). For a single depot 0, a set of homogeneous vehicles K with the same capacity limit Q . The goal of the CVRP is to determine the routes of the vehicles that can satisfy all the customers' needs while minimizing total distance. The mathematical representation of CVRP can be modeled as [23]:

$$\text{Minimize } \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K c_{ij} X_{ij}^k \quad (1)$$

Subject to

$$\sum_{k=1}^K \sum_{i=0}^N X_{ij}^k = 1 \quad j \in \{1, \dots, N\} : i \neq j \quad (2)$$

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

$$\sum_{i=0}^N \sum_{j=0}^N X_{ij}^k q_i \leq Q \quad k \in \{1, \dots, K\} \quad (4)$$

$$\sum_{j=1}^N X_{ij}^k = \sum_{j=1}^N X_{ji}^k \leq 1 \quad \text{for } i = 0 \text{ and } k \in \{1, \dots, K\} \quad (5)$$

Equation (1) is the objective function that minimizes the total distance traveled by the vehicles. If the vehicle k travels from customer i to customer j directly, $X_{ij}^k = 1$ otherwise, $X_{ij}^k = 0$. The remaining equations represent a series of constraints as follows: (i) Equation (2) and (3) make sure that each customer is served exactly by one vehicle; (ii) constraint (4) ensures that total demand in a route does not exceed the capacity of each vehicle Q ; and (iii) constraints (5) guarantees that each vehicle's route starts and ends at the same depot. A solution is feasible if no customer is served more than once and all routes meet the capacity constrain. As for the number of the vehicles k , some researchers consider it as a decision variable [22], however, some researchers consider it as a constrain [24].

3. The proposed dynamic space reduction ant colony optimization

A new variant of ACO, called DSRACO, is proposed. The main goal of DSRACO is to overcome the drawbacks of ACO that easily get stuck in local optima and poor solution. Dynamic space reduction method and elite enhanced mechanism to improve the quality of the solution. In addition, traditional neighborhood search methods cannot effectively avoid local optima, DSRACO integrates a unique large-scale neighborhood search strategy to guarantee the exploitation of the solution space. The pseudo-code of DSRACO is shown in Algorithm 1. In each step, DSRACO must satisfy the constraints of CVRP. The proposed DSRACO will be discussed in detail in this section:

Algorithm 1 DSRACO for CVRP

```

1 Input: objective function  $f(x)$ ;
2 Output: the best ant  $G_{best}$ ;
3 Initialize  $v=\{v_1, v_2, \dots, v_n\}$  is the set of  $n$  cities and  $v_0$  is the depot and  $G_{best}=Inf$ ;
  Initialize parameters  $PS, MI, Q, \alpha=1, \beta=5, \rho=0.2$ ;
4  $Iter = 1$ ;
5 Initialize ant population  $Pop: X=(x_1, x_2, \dots, x_{PS})$ .
6 For  $Iter = 1$  to  $MI$  do
7   For  $Pop = 1$  to  $PS$  do
8     Build each ant by applying  $n-1$  times the following steps
9     Generate the cities that have not been visited  $\Omega$  and sort it by distance
10    Determine the number of space reduction feasible cities by  $SR=floor(n \times e^{-10 \times Iter/MI} + 3)$ 
11    Choose top  $SR$  cities from  $\Omega$  to form  $F$ 
12    From city  $i$  choose the next city  $j$  with probability
      
$$p_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum [\tau_{ij}]^\alpha [\eta_{ij}]^\beta} \quad \text{if } v_j \in F$$

      else  $p_{ij} = 0$  and get Initial ant colony
13    Generate Neighborhood ant colony from Initial ant colony by Large-scale neighborhood search operators
14  End for
15  Merge colonies and calculate the total distance of each ant
16  Select the top  $N$  as the elites
17  For  $Elite = 1$  to  $N$ 
18    
$$\tau_{ij}^{new} = (1 - \rho) * \tau_{ij}^{old} + \sum_{l=1}^N \Delta \tau_{ij}^l$$

      
$$\Delta \tau_{ij}^l = \begin{cases} \frac{Q}{L_t}, & \text{if edge } (i, j) \text{ belongs to } EliteSolution \\ 0, & \text{otherwise} \end{cases}$$

      where
19  End for
20  If  $Iter_{best} < G_{best}$ 
21     $G_{best} = Iter_{best}$ 
22  End if
23 End for
24 Return the best ant  $G_{best}$ ;

```

3.1. Overall framework of DSRACO

The overall framework of DSRACO can be described as follows: (i) Initialization;(ii)Establish the feasible cities set by dynamic space reduction method;(iii) Build the ant colony(named initial ant colony) according to heuristic function and path pheromone content;(iv) Perform neighborhood search on initial ant colonies to generate neighborhood ant colony;(v) Merge initial ant colony with neighborhood ant colony and calculate the distance cost of each ant;(vi) Select several best elites to update the pheromone and update the current global optimal solution;(vii) Repeat step(ii) to step(vi) until the maximum iteration is met;(viii) Output the global optimal solution of each generation. The overall framework of the proposed DSRACO can be seen in Fig. 1.

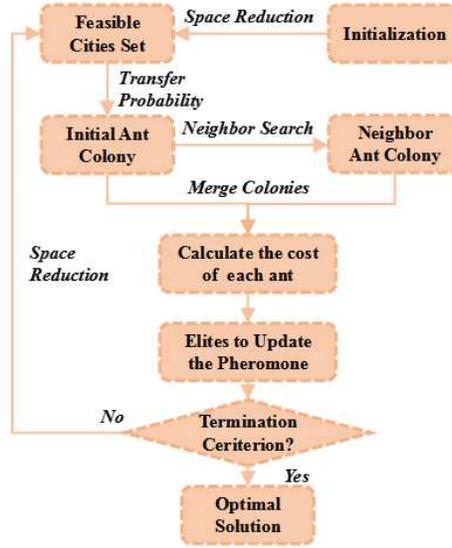


Fig. 1. Overall framework of DSRACO

3.2. Solution representations

In order to introduce DSRACO more concisely, we'd like to first explain the representations of the solution. DSRACO generates the solution to CVRP in two permutation representations. As shown in Fig. 2, CVRP solution representation is represented as an array list $\sum_{i=1}^K (S_i + 2)$, where K is the number of vehicles, S_i is the number of customers who are assigned to each route and 0 is the start and the endpoint (the depot) of each route i . TSP solution representation is represented as one-dimension array (N) , where N is the number of customers including the depot. The two forms of solution representations can make it simple to convert a TSP solution to a CVRP solution and vice versa. This strategy considerably simplifies the algorithm process and facilitates large-scale neighborhood search.

1	2	3	4	5	6	7	8	9	10	11	12	13
---	---	---	---	---	---	---	---	---	----	----	----	----

(a) TSP solution representation

0	1	2	3	4	0	
0	5	6	7	8	0	
0	9	10	11	12	13	0

(b) CVRP solution representation

Fig. 2. Two solution representation.

3.3. Initialization

In the initialization phase, the basic parameters of the algorithm are given, as well as the customer location, the distance between each customer, and the distance between each customer and the depot. Besides, set the initial total distance cost to infinity.

The basic parameters to be set are the maximum number of iterations (MI), population size (PS), the pheromone evaporation rate (ρ), the relative influence of pheromone (α), the relative influence of visibility (β) and the amount of pheromone deposited by each ant (Q). The basic information of distance is the distance between customer i and customer j (d_{ij}) and the distance from customer i to the depot (d_{i0}).

3.4. Dynamic space reduction method

As one of the important contributions of this paper, the dynamic space reduction method enables the ant colony to explore the whole solution space in the initial stage of the algorithm, and with the iteration going on, each ant only chooses some nearest cities as the next destination. The dynamic space reduction method greatly improves the quality of the solution, and the method will be elaborated upon in the following content.

First, we need to get a set of cities that have not been visited $\Omega \subseteq (V \setminus v_0)$. Then, sort the elements in Ω by distance. After that, select the top SR elements from Ω according to equation (6) to form F .

$$SR = \text{floor}(n \times e^{-10 \times \frac{iter}{MI}} + 3) \quad (6)$$

Here floor denotes the rounding down function, n is the total number of cities to be visited, $Iter$ is the current iteration and MI is the maximum iteration. We can see that when $Iter$ is small, SR is approximately equal to n and when $Iter$ is large SR is approximately equal to 3. Dynamic space reduction is completed by the above method.

3.4. Method to build the initial ant colony

As with most ant colony algorithms[18], this paper uses the reciprocal of the distance between two customers (7) as a heuristic function.

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (7)$$

Transfer probability between customers is given by (8).

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h \in \Omega} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta}, & v_j \in F \\ 0, & v_j \notin F \end{cases} \quad (8)$$

Here, τ_{ij} denotes the pheromone content on the path between customer i and customer j , F is the feasible cities generated by dynamic space reduction, α and β represent the biases for pheromone trail and visibility respectively, and they represent parameters to weigh pheromone concentration.

The pheromone update rule is given by (9).

$$\tau_{ij}^{new} = (1 - \rho) * \tau_{ij}^{old} + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (9)$$

Each ant is constructed according to the following steps: (i) each route takes the depot as the start point and the endpoint;(ii) each customer is selected from the candidate list by (8), and once the customer is selected, it is deleted from the list to guarantee the customers is not selected again;(iii) the selected customer is added to the route in order; (iv) if the demand of the corresponding route does not exceed the load limit of the vehicle; repeat step(ii) and step(iii); Otherwise, add the depot to the route as the endpoint and construct the next vehicle's route; (v) steps (i) to (iv) are repeated until all customers' needs are satisfied; (vi) check the feasibility of each route. Step(vi) is essential to CVRP, but it is often overlooked.

3.5. Large-scale neighborhood search operators

The main goal of neighborhood search is to prevent the algorithm from falling into local optima and improve the quality of the solution. The neighborhood search methods of DSRACO mainly include 2-opt, shift, reverse, and random sort operators. These neighborhood search operators will be explained in the TSP solution.

2-opt operator. Randomly exchange a pair of nodes. Fig. 3 shows the new solution obtained by 2-opt operator.

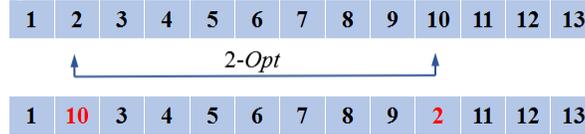


Fig. 3. New solution obtained by 2-Opt operator

Reverse operator. Randomly select two nodes, e.g. By reverse operator we can get a new sequence shown in Fig. 4.



Fig. 4. New solution obtained by reverse operator

Shift operator. Randomly select two nodes, e.g. The first customer between the two nodes is moved to the last position and the other customers are respectively moved to the position next to them. The new solution obtained by shift operator is shown in Fig. 5.



Fig. 5. New solution obtained by shift operator

Random sort operator. Randomly generate a row vector between 5 and $n/2$, use this vector as a node subscript, then shuffle its order and sort it randomly. The result is shown in Fig. 6.

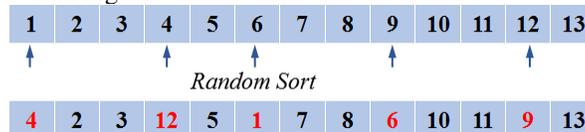


Fig. 6. New solution obtained by random sort operator

3.6. Update of global optimal solution and elite enhanced mechanism

Once the ant is constructed, its corresponding objective function value is calculated. The objective function is the total distance cost, which includes the distance from the depot to the first customer and the last customer to depot. Since the goal of CVRP is to minimize the total distance, the ant with a lower objective function will replace the current global optimal solution.

As mentioned above, only elites can update pheromones. Thus, the number of elites is particularly critical. If the number of elites is too small, the global search ability of the algorithm is poor, and the solution is not ideal, but if there are too many elites, the convergence rate of the algorithm will be significantly slower. Each elite updates the pheromone according to (9), and discussion on the number of elites will be introduced later.

4. Parameter setting

There are many adjustable parameters in ant colony optimization (ACO)[25], and these parameters have a great influence on the final solution. According to [26][27][28], the relative influence of pheromone α is set to 1, the relative influence of visibility β is set to 2, and the pheromone evaporation rate ρ is set to 0.1. Considering the computing resources required by different problem dimensions, the population size (PS) is set to $10 \times N$, where N is the number of customers including the depot. The maximum iteration (MI) is set to 1500 to stop the main loop of DSRACO.

Apart from α , β , ρ , PS and MI , the number of elites (NE) is also a key parameter for the proposed algorithm. To determine NE , a series of comparison experiments are carried out on different standard benchmarks **Error! Reference source not found.** For A, B, P set benchmarks, A-n33k6, A-n38k5, A-n46k7, A-n65k9, B-n31k5, B-n44k7, B-n68k9, P-n19k2, P-n22k8, P-n60k15 are selected. DSRACO runs 30 times independently for each tested instance in different NE (1,2,3,4,5,10,15) and the statistical results in terms of mean value are shown in Table 1. For each instance, the optimal result is bold and labeled.

Table 1 Statistical results of different number of elites

Instance	BKS	Number of Elite (NE)						
		1	2	3	4	5	10	15
A-n33-k6	742	744.2	744.7	743.1	742.9	743.9	745.8	746.3
A-n38-k5	730	734.6	734.3	732.2	733.7	735.1	737.2	736.9
A-n46-k7	914	917.2	915.9	917.6	917.8	919.3	920.5	920.4
A-n65-k9	1174	1185.4	1184.6	1182.3	1185.1	1186.4	1190.3	1189.8
B-n31-k5	672	674.1	673.7	672.9	673.4	672.7	677.5	679.6
B-n44-k7	909	914.6	913.9	912.3	913.7	915.2	916.4	918.5
B-n68-k9	1272	1297.6	1295.8	1293.0	1297.4	1299.8	1301.4	1300.1
P-n19-k2	212	213.8	213.6	212.5	213.2	214.6	214.7	215.3
P-n22-k8	590	591.1	592.3	591.5	593.6	594.9	594.7	597.0
P-n60-k15	968	979.4	977.9	976.7	979.2	982.3	981.9	983.4

As shown in Table 1, the number of elite NE has a great influence on the final solution of the proposed DSRACO. When NE is small, the algorithm is more likely to be guided by optimal individuals and tends to perform local search. When NE is large, the algorithm is more likely to perform global exploration and obtain the global optimal. In all ten tested instances, six optimal results are achieved by $NE=3$ and the other four optimal results are obtained by $NE=1$, $NE=2$, $NE=4$, and $NE=5$, respectively. Without doubt, when $NE=3$, DSRACO can have a good balance between local exploitation and global exploration and show the best performance. The parameter settings are summarized in Table 2.

Table 2 Parameter settings of DSRACO

Parameters Setting					
PS	MI	α	β	ρ	NE
$10 * N$	1500	1	2	0.1	3

5. Computational study

A series of comparison experiments were carried out to verify the effectiveness of the proposed DSRACO. DSRACO is compared with the original ant colony optimization in terms of mean value, best solution, and runtime for the same population size and maximum iterations. Besides, an improved version of ACO (LNS-ACO)[19] and other two state-of-the-art algorithms (OHGA[14], CVRP-FA[17]) are also tested in three series of standard benchmarks A, B, and P **Error! Reference source not found.** For DSRACO and ACO, each algorithm runs 30 times independently and is tested with specifications of Intel i5-8400 at 2.80GHz CPU and 8GB RAM. Runtime is measured in units of seconds. For LNS-ACO, OHGA, and CVRP-FA, the best solutions are intercepted from the paper to maintain the quality of the algorithm.

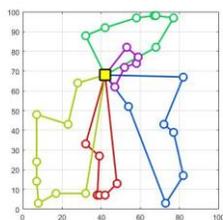
5.1. Comparison results of A-set benchmark

As an important standard benchmark, A set benchmark contains 27 instances from 32 customers to 80 customers, which can test the ability of each algorithm to solve discrete combinatorial problems. The comparison results of A-set benchmark are shown in

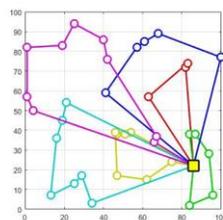
Table 3, and the best result of each instance is bold. Besides, if the best solution reaches the Best Known Solution (BKS), a grey label is added. To show the results more intuitively, Fig 7 demonstrates the planned routes obtained by DSRACO.

Table 3 Comparison results of A-set benchmark

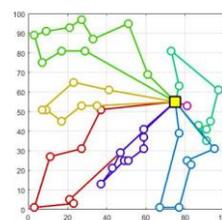
A Set Benchmark										
Instance	BKS	ACO			DSRACO			LNS-ACO	OHGA	CVRP-FA
		Mean	Best	Time	Mean	Best	Time	Best	Best	Best
n32k5	784	825.2	813	127.2	786.9	784	139.6	784	784	796
n33k5	661	691.4	679	130.5	662.7	661	142.9	661	661	661
n33k6	742	771.3	760	132.7	743.1	742	146.5	742	742	742
n34k5	778	812.0	801	137.9	780.3	778	150.7	778	778	778
n36k5	799	839.8	826	157.3	803.6	799	174.5	799	799	799
n37k5	669	728.7	716	164.7	672.5	669	182.2	669	669	669
n37k6	949	980.7	967	168.3	953.0	949	179.4	949	949	949
n38k5	730	763.6	752	177.8	732.2	730	194.6	730	730	730
n39k5	822	857.1	844	189.0	825.8	822	206.1	822	822	822
n39k6	831	872.9	858	187.6	835.4	832	202.6	831	833	831
n44k6	937	973.6	960	266.6	942.0	938	285.3	937	937	937
n45k6	944	987.5	972	252.0	956.1	953	274.7	958	953	953
n45k7	1146	1175.4	1162	246.2	1151.3	1148	260.9	1146	1146	1147
n46k7	914	941.0	931	263.1	917.6	914	284.2	914	914	914
n48k7	1073	1109.2	1098	286.1	1078.4	1075	306.1	1084	1073	1073
n53k7	1010	1047.8	1035	361.2	1016.5	1013	393.7	1010	1017	1011
n54k7	1167	1195.4	1182	375.3	1172.6	1167	397.8	1167	1167	1172
n55k9	1073	1108.6	1095	394.1	1078.8	1073	425.6	1073	1074	1074
n60k9	1354	1384.5	1371	481.5	1359.7	1356	507.9	1354	1355	1355
n61k9	1034	1062.6	1053	499.3	1039.1	1035	539.2	1067	1035	1039
n62k8	1288	1337.7	1320	515.2	1312.0	1306	551.3	1308	1308	1298
n63k9	1616	1668.4	1655	536.8	1635.2	1630	574.3	1649	1630	1630
n63k10	1314	1362.1	1347	648.9	1318.5	1316	694.6	1329	1329	1314
n64k9	1401	1460.3	1443	627.8	1421.8	1418	684.5	1415	1416	1420
n65k9	1174	1217.0	1200	655.9	1182.3	1178	721.4	1185	1184	1178
n69k9	1159	1199.2	1183	752.1	1169.0	1165	804.7	1170	1170	1162
n80k10	1763	1822.9	1803	1067.2	1784.4	1778	1152.5	1815	1790	1773



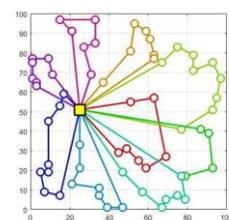
(a)A-n33k5



(b)A-n37k6



(c)A-n46k7



(c)A-n65k9

Fig. 7. A-set benchmark results obtained by DSRACO

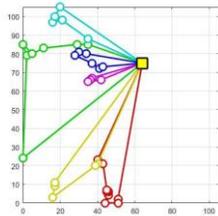
As shown in Table 3, DSRACO gets 15 optimal solutions in 27 instances in comparison with ACO, LNS-ACO, OHGA, and CVRP-FA. It is also worth noting that DSRACO achieves BKS in 12 instances, which fully proves that the proposed algorithm has a satisfactory solution quality. In all 27 tested A-set instances, DSRACO outperforms ACO in terms of mean value and best solution. However, DSRACO takes a little longer runtime than ACO as stated above. Compared to LNS-ACO, DSRACO performs better in A-n45k6, A-n61k9, A-n65k9 but is slightly inferior in six instances.

5.2. Comparison results of B-set benchmark

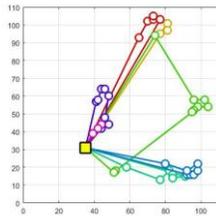
B set benchmark plays an indispensable role in CVRP, which consists of customers from 31 to 78 and vehicles from 5 to 10. In B set benchmark, ACO and DSRACO are compared in terms of mean value, best solution, and runtime. For LNS-ACO, OHGA, and CVRP-FA, the best solutions are intercepted from the corresponding papers. The statistical results are summarized in Table 4 and the optimal value for each instance is bold. Additionally, if the BKS is found, a grey label will be added. Fig.8 shows the planned routes obtained by DSRACO to show the results more intuitively.

Table 4 Comparison results of B-set benchmark

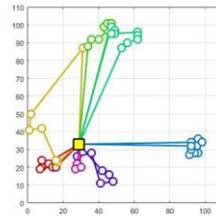
B Set Benchmark										
Instance	BKS	ACO			DSRACO			LNS-ACO	OHGA	CVRP-FA
		Mean	Best	Time	Mean	Best	Time	Best	Best	Best
n31k5	672	712.4	705	116.7	672.9	672	126.1	672	672	672
n34k5	788	819.2	806	140.1	780.1	788	149.9	788	788	788
n35k5	955	992.5	979	165.0	957.3	955	179.8	955	955	955
n38k6	805	840.7	827	178.6	808.0	805	194.7	805	805	806
n39k5	549	579.0	563	187.8	551.4	549	206.6	549	549	550
n41k6	829	864.3	844	209.7	831.8	829	226.5	829	829	829
n43k6	742	782.3	767	232.8	744.5	742	258.4	742	742	742
n44k7	909	948.6	932	235.4	912.3	909	254.2	909	909	909
n45k5	751	791.2	775	245.9	754.7	751	268.1	751	751	751
n45k6	678	713.8	702	248.6	680.6	678	271.0	678	680	686
n50k7	741	782.9	767	316.7	744.8	741	348.4	741	741	741
n50k8	1312	1365.0	1351	320.0	1319.6	1316	345.6	1319	1315	1318
n51k7	1016	1067.1	1049	331.2	1020.0	1017	364.4	1016	1017	1032
n52k7	747	794.7	778	358.8	746.4	747	391.1	747	747	747
n56k7	707	751.1	736	409.4	713.2	710	442.1	707	711	709
n57k7	1153	1194.2	1180	484.5	1157.1	1153	533.0	1153	1140	1153
n57k9	1598	1644.6	1627	483.9	1608.9	1605	522.6	1598	1603	1610
n63k10	1496	1552.8	1539	611.1	1511.2	1508	659.9	1514	1531	1503
n64k9	861	900.4	883	642.6	867.7	866	706.8	874	867	862
n66k9	1316	1366.5	1347	680.5	1325.6	1322	748.5	1330	1324	1319
n67k10	1032	1088.7	1070	704.5	1049.8	1046	775.0	1050	1042	1042
n68k9	1272	1340.2	1325	734.7	1293.0	1290	793.5	1290	1290	1278
n78k10	1221	1268.8	1251	1024.1	1240.5	1237	1126.8	1228	1245	1224



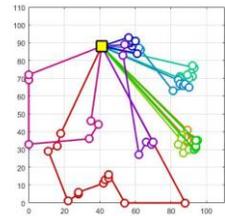
(a)B-n38k6



(b)B-n44k7



(c)B-n52k7



(d)B-n66k9

Fig. 8. B-set benchmark results obtained by DSRACO

As shown in Table 4, the optimal solutions obtained by DSRACO are 14 out of 23. Moreover, DSRACO finds 13 BKS, outperforming OHGA and CVRP-FA. DSRACO outperforms ACO in terms of mean value and best solution in all tested B-set instances. It's also worth mentioning that, because of the large-scale neighborhood search operators, DSRACO takes a little longer to run than ACO. As the data in Table 4, DSRACO performs better in B-n63k10, while the LNS-ACO performs better in B-n51k7, B-n56k7, and B-n57k9.

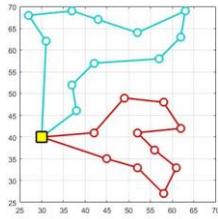
5.3. Comparison results of P-set benchmark

P set benchmark is an essential standard benchmark that has 23 instances ranging from 16 to 101 customers and may be used to assess the ability of each algorithm to tackle discrete combinatorial problems. Table 5 shows the comparative results of the P-set benchmark, with the best result of each instance bolded. A grey label is also placed if the optimal solution achieves the Best Known Solution (BKS). The planned routes obtained by DSRACO are shown in Fig.9 to make the results more apparent.

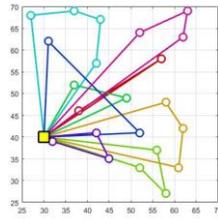
Table 5 Comparison results of P-set benchmark

P Set Benchmark										
Instance	BKS	ACO			DSR-ACO			LNS-ACO	OHGA	CVRP-FA
		Mean	Best	Time	Mean	Best	Time	Best	Best	Best
n16k8	450	470.3	462	29.6	450.9	450	32.0	450	450	450
n19k2	212	216.8	212	39.9	212.7	212	43.1	212	212	212
n20k2	216	223.5	218	44.5	217.0	216	47.6	216	216	216
n21k2	211	218.7	213	49.4	212.3	211	53.3	211	211	211
n22k2	216	228.1	220	54.3	217.8	216	58.7	216	216	216

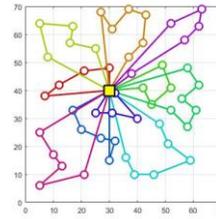
n22k8	590	625.3	615	56.3	591.5	590	61.4	590	590	590
n23k8	529	563.0	551	65.8	531.2	529	71.0	529	529	529
n40k5	458	488.9	477	199.4	460.6	458	213.3	458	458	458
n45k5	510	550.6	536	256.5	513.1	510	274.4	510	510	510
n50k7	554	593.4	580	330.0	557.3	555	356.9	554	556	554
n50k8	631	667.3	652	316.2	638.6	635	344.1	643	630	631
n50k10	696	732.5	719	318.3	701.9	698	346.8	696	700	697
n51k10	741	780.0	766	332.2	743.4	741	365.5	747	741	742
n55k7	568	613.8	592	399.3	570.8	568	431.2	568	568	568
n55k10	694	726.9	713	447.4	698.2	695	492.6	694	698	698
n55k15	989	1032.7	1014	461.7	992.0	989	503.3	989	989	\
n60k10	744	797.4	777	545.4	749.1	747	600.0	755	749	749
n60k15	968	1008.7	995	558.6	976.7	972	614.5	977	985	968
n65k10	792	835.6	820	663.7	799.2	795	710.2	800	797	792
n70k10	827	868.0	852	805.1	830.9	827	861.4	837	841	827
n76k4	593	625.5	611	951.5	598.6	595	1027.6	598	600	593
n76k5	627	677.2	658	939.0	632.5	630	1023.5	645	630	628
n101k4	681	722.1	706	1801.3	685.9	683	1977.4	\	696	681



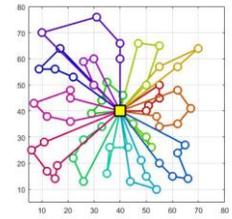
(a)P-n21k2



(b)P-n23k8



(c)P-n51k10



(d)P-n60k15

Fig.9. P-set benchmark results obtained by DSRACO

The optimal solutions obtained by DSRACO are 14 out of 23, as indicated in Table 4. DSRACO also detects 13 BKS, exceeding OHGA and CVRP-FA. In all B-set instances tested, DSRACO exceeds ACO in terms of mean value and best solution. Compared with ACO, DSRACO takes less time to update pheromones but more time to operate the large scale neighborhood search. According to Table 4, DSRACO outperforms LNS-ACO in P-n51k10 and P-n70k10, while LNS-ACO outperforms P-n50k7, P-n51k7, and P-n55k10.

5.4. Performance analysis

This section conducts a series of comparison experiments with benchmarks A, B, and P. DSRACO is compared with ACO in terms of mean value, best solution, and runtime. Furthermore, the best solutions of DSRACO and three other state-of-the-art algorithms are compared. To show the statistical results more intuitively, the BKS and optimal solution found by each algorithm are summarized in Table 6.

Table 6 Statistical results of benchmark A, B, and P

Benchmark	DSRACO		LNS-ACO		OHGA		CVRP-FA		
	BKS	Optimal	BKS	Optimal	BKS	Optimal	BKS	Optimal	
A	27	12	15	17	18	14	17	12	17
B	23	13	14	16	16	11	13	10	15
P	23	13	14	14	14	13	13	17	18
Total	73	38	43	47	48	38	43	39	50

As Table 6 states, DSRACO finds 38 BKS and 43 optimal solutions out of 73, which is a competitive result. DSRACO and OHGA performed consistently in both BKS and Optimal. Compared with LNS-ACO and CVRP-FA, DSRACO appears to be less satisfactory mainly in terms of Optimal. However, it is also worth noting that DSRACO outperforms ACO in all tested instances with only a slightly longer runtime.

In summary, DSRACO performs much better than ACO and can solve capacitated vehicle routing problem (CVRP) with a satisfactory solution.

6. Conclusion

This paper presents a dynamic space reduction ant colony optimization to solve capacitated vehicle routing problem. In the proposed DSRACO, ACO is integrated with a unique dynamic space reduction method, an elite enhanced mechanism, and

large-scale neighborhood search methods to improve the solution. The dynamic space reduction method can avoid local optima. The elite enhanced mechanism further improves the quality of the solution. Besides, four methods of neighborhood search, including 2-opt, reverse, shift, and random sort are integrated to improve the exploitation of the solution space in which random sort can effectively explore the global space. The experimental results show that DSRACO is effective for solving CVRP instances with satisfactory performance. DSRACO shows great advantages in the comparison of ACO and is competitive compared with the other three state-of-the-art algorithms.

In future work, the DSRACO algorithm can be applied to solve other types of VRP or other discrete combination optimization problems. Besides, DSRACO can be further developed using a multi-objective optimization framework.

Compliance with Ethical Standards

Funding: This study was funded by the National Natural Science Foundation of China (Grant No. 51875466) and (Grant No. 52175251).

Conflict of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval: This article does not contain any studies with human participants performed by any of the authors.

Data availability statement

The data, models, and code required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

References

- [1] Richard M. Karp, Reducibility among combinatorial problems, *Complexity of computer computations*, (1972) 85-103. https://doi.org/10.1007/978-1-4684-2001-2_9.
- [2] Dantzig G.B., and Ramser J.H., The truck dispatching problem. *Management science* 6.1 (1959): 80-91. <https://doi.org/10.1287/mnsc.6.1.80>.
- [3] Baldacci R., Mingozzi A., and Roberti R., Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218.1 (2012): 1-6. <https://doi.org/10.1016/j.ejor.2011.07.037>.
- [4] Baldacci R., Christofides N. and Mingozzi A. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Math. Program.* 115, 351–385 (2008). <https://doi.org/10.1007/s10107-007-0178-5>.
- [5] Costa D., Hertz A. Ants can colour graphs. *Journal of the Operational Research Society* 48, 295–305 (1997). <https://doi.org/10.1057/palgrave.jors.2600357>
- [6] Huang R., Yang C. Ant colony system for job shop scheduling with time windows. *Int J Adv Manuf Technol* **39**, 151–157 (2008). <https://doi.org/10.1007/s00170-007-1203-9>
- [7] Euch J. and Mraih R. The urban bus routing problem in the Tunisian case by the hybrid artificial ant colony algorithm. *Swarm and Evolutionary Computation* 2 (2012): 15-24. <https://doi.org/10.1016/j.swevo.2011.10.002>
- [8] Gendreau, M., Potvin, JY. Metaheuristics in Combinatorial Optimization. *Ann Oper Res* **140**, 189–213 (2005). <https://doi.org/10.1007/s10479-005-3971-7>
- [9] Gendreau M., Potvin JY., Bräumlaysy O., Hasle G., Løkketangen A. (2008) Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography. In: Golden B., Raghavan S., Wasil E. (eds) *The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces*, vol 43. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-77778-8_7
- [10] Dorigo M. "Optimization, learning and natural algorithms." *Ph. D. Thesis, Politecnico di Milano* (1992).
- [11] Dorigo M., Maniezzo V., and Colomi A, Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26.1 (1996): 29-41. doi: 10.1109/3477.484436.
- [12] Wang X., Choi T., Liu H. and Yue X., Novel Ant Colony Optimization Methods for Simplifying Solution Construction in Vehicle Routing Problems. *IEEE Transactions on Intelligent Transportation Systems*, 17(11)(2016) : 3132-3141. doi: 10.1109/TITS.2016.2542264.
- [13] Zhang H., Zhang Q., Ma L., Zhang Z. and Liu Y., A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. *Information Sciences*, 490(2019):166-190. <https://doi.org/10.1016/j.ins.2019.03.070>
- [14] Lin N., Shi Y., Zhang T. and Wang X., An Effective Order-Aware Hybrid Genetic Algorithm for Capacitated Vehicle Routing Problems in Internet of Things. *IEEE Access*, 7(2019):86102-86114. doi: 10.1109/ACCESS.2019.2925831.
- [15] Szeo W.Y., Wu Y. and Ho S.C., An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research*, 215(1) (2011) :126-135. <https://doi.org/10.1016/j.ejor.2011.06.006>
- [16] Kao Y., Chen M. and, and Huang Y., A Hybrid Algorithm Based on ACO and PSO for Capacitated Vehicle Routing Problems. *Mathematical Problems in Engineering*, 8(2012):1-17. <https://doi.org/10.1155/2012/726564>
- [17] Altabeeb A.M., Mohsen A.M., and Ghallab A., An improved hybrid firefly algorithm for capacitated vehicle routing problem. *Applied Soft Computing*, 84(2019):105728. <https://doi.org/10.1016/j.asoc.2019.105728>

- [18] Goel R., Maini R., A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems. *Journal of Computational Science*, vol.25, pp.28-37, 2018. <https://doi.org/10.1016/j.jocs.2017.12.012>.
- [19] Akpınar S., Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem. *Expert Systems with Applications*, 61(2016) :28-38. <https://doi.org/10.1016/j.eswa.2016.05.023>
- [20] Mahdi A. and Shokouhi N., Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. *Omega*, 76(2018): 85-99. <https://doi.org/10.1016/j.omega.2017.05.002>
- [21] Juan A., Faulin J., Ruiz R., et al., The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem, *Applied Soft Computing*, 10(1)(2010):215-224. <https://doi.org/10.1016/j.asoc.2009.07.003>
- [22] Amous M., Toumi S., Jarboui B., Eddaly M., A variable neighborhood search algorithm for the capacitated vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 58(2017):231-238. <https://doi.org/10.1016/j.endm.2017.03.030>
- [23] Lin S., Lee Z., Ying K., Lee C. Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Systems with Applications*, 36(2)(2009):1505-1512. <https://doi.org/10.1016/j.eswa.2007.11.060>
- [24] Narasimha K.V., Kivelevitch E., Sharma B., Kumar M., An ant colony optimization technique for solving min–max multi-depot vehicle routing problem. *Swarm and Evolutionary Computation*, 13(2013):63-73. <https://doi.org/10.1016/j.swevo.2013.05.005>
- [25] Oliveira S., Hussin M.S., Roli A., Dorigo M. and Stützle T., Analysis of the population-based ant colony optimization algorithm for the TSP and the QAP. *2017 IEEE Congress on Evolutionary Computation (CEC)(2017):1734-1741*, doi: 10.1109/CEC.2017.7969511.
- [26] Situ X., Chen W., Gong Y., et al., A parallel Ant Colony System based on region decomposition for Taxi-Passenger Matching, *2017 IEEE Congress on Evolutionary Computation (CEC)(2017) :960-967*, doi: 10.1109/CEC.2017.7969412.
- [27] Jangra R., Kait R, Analysis and Comparison Among Ant System; Ant Colony System and Max-Min Ant System with Different Parameters Setting. *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)(2017):1-4*, doi: 10.1109/CICT.2017.7977376.
- [28] Melo L., Pereira F. and Costa E., Extended Experiments with Ant Colony Optimization with Heterogeneous Ants for Large Dynamic Traveling Salesperson Problems, *2014 14th International Conference on Computational Science and Its Applications*, (2014): 171-175, doi: 10.1109/ICCSA.2014.39.
- [29] Augerat. (2006). The VRP Web. [Online]. Available: <http://neo.lcc.uma.es/radi-aeb/WebVRP/>