[1] Xinyu Zhang
[1] Kaiwen Geng
[*, 2] Yang Li

Hybrid Artificial Bee Colony Algorithm with Variable Neighborhood Search for Capacitated Vehicle Routing Problem

Journal of Electrical Systems

Abstract: - Aiming at the capacitated vehicle routing problem, a hybrid integer programming model with goal of lowest path cost is constructed, and a hybrid artificial bee colony algorithm with variable neighborhood search based on the model and the characteristics of the CVRP problem is proposed to solve the problem. The hybrid algorithm integrates the artificial bee colony algorithm and the variable neighborhood search algorithm, embeds a multi-variable neighborhood operator in the local search link of the artificial bee colony to carry out iteration. And the operator contains targeted transformation operations on path nodes, strings, and sub paths to ensure the diversity of the bee population. In addition, a variable neighborhood perturbation strategy is used to strengthen the algorithm's ability to escape from local optima. The comparative analysis of the literature study set and its algorithm solution shows that the designed hybrid variable neighborhood artificial bee colony algorithm has strong global search ability and high solution accuracy, especially in solution stability. HABC-VNS can obtain 47 optimal solutions in 74 examples. The average minimum deviation of the optimal solution is 0.34%, and the average deviation of the average CVRP set is 0.57%. The overall performance is better than the algorithm in the comparative literature.

Keywords: vehicle routing problem; artificial bee colony; variable neighborhood search; capacity constraint.

I. INTRODUCTION

Since the beginning of the 21st century, the Internet industry has developed vigorously. With the rapid development of information network technology and the widespread popularization of mobile intelligent terminals, the mobile Internet industry has achieved corner overtaking. The rapid development of Internet and mobile Internet industry has given birth to the individuation of consumer demand, resulting in great changes in the consumption mode of residents. Especially during the global COVID-19 pandemic, new forms and models of sharing economy such as online and offline, life service e-commerce and rural e-commerce have expanded rapidly [1]. However, the ultimate service of the sharing economy is inseparable from the support of the e-commerce logistics system. At present, e-commerce logistics still has great room for improvement in key links such as the "last mile" and service quality. The core problem is the capacitated vehicle routing problem(CVRP). The research on CVRP can provide a universal basic optimization framework for e-commerce logistics operations in complex environment. The path planning scheme will be continuously improved to improve logistics efficiency and provide reference and basis for solving combinatorial optimization problems such as resource allocation and business scheduling.

The research on CVRP has received extensive attention from many scholars, and its focus is mainly on the research and improvement of the solution algorithm. It mainly includes two categories: exact algorithm and heuristic algorithm [2,3]. In terms of exact algorithms, Liu et al. [4] proposed a branch-and-cut algorithm for the two-echelon capacitated vehicle routing problem with grouping constraints. The problem was formulated as a mixed 0-1 linear program and five families of valid inequalities were proposed to strengthen the model. In addition, dynamic programming [5], Branch-price-and-cut [6], Branch-and-price [7] and other exact solution methods are also extensively used [8,9]. However, when the scale of customer points tends to become larger, the solution efficiency of the exact algorithm will decrease significantly. At this time, most scholars will use heuristic and meta-heuristic algorithms for optimization [10,11]. Dimitra et al. [12] introduced the Environmental Prize-Collecting Vehicle Routing Problem (E-PCVRP) and the Firefly Algorithm based on Coordinates (FAC) was proposed, which incorporated the proposed “Coordinates Related” (CR) encoding/decoding process in the original FA scheme. Li et al. [13] developed a multi-depot green vehicle routing problem (MDGVRP) by maximizing revenue and minimizing costs, time and emission, and then, apply an improved ant colony optimization (IACO) algorithm that aims to efficiently solve the problem. The IACO had higher solution quality when compared to the conventional ACO. Vincent et al. [14] presented the symbiotic organisms search (SOS) heuristic for solving the CVRP. They applied two solution representations, SR-1 and SR-2, to transformed SOS

[1] School of Economics, Liaoning University, Shenyang 110136, China
[2] School of Economics and Management, Liaoning Petrochemical University, Fushun 113001, China
[*]Corresponding Author: Yang Li

into an applicable solution approach for CVRP and then applied a local search strategy to improve the solution quality of SOS.

Hyperheuristics use hyperheuristic framework, high-level heuristic strategy and low-level heuristic operator to solve the optimization problem which provides a more general process for optimization [15]. Hyperheuristics solves the problem through the management of low-level heuristic algorithms, without specific parameter adjustment. Some scholars have introduced it into the solution of various combinatorial optimization problems [16,17]. In addition, some scholars have also used machine learning, reinforcement learning and other methods to try to solve the VRP problem [18,19].

For the capacitated vehicle routing problem, a hybrid artificial bee colony algorithm with variable neighborhood search (HABC-VNS) is designed to solve the problem. The hybrid algorithm designs six variable neighborhood operators based on node, string and path operations to participate in the optimization. The neighborhood radius reduction strategy is applied to control the search range, and the individual disturbance of the population is expanded through the tabu table setting. And the algorithm accepts inferior solutions, and the effectiveness of the parameter setting of the hybrid algorithm and the combination of variable neighborhood operators is verified through experiments. The variable neighborhood search combined with the population iteration mechanism of artificial bee colony enables HABC-VNS to have strong global and local search capabilities, which can effectively jump out of the local optimum and meet the needs of CVRP solution.

The rest of this study is arranged as follows. In Section 2, we provide a problem statement and an integer programming model is constructed. In Section 3, the implementation of the hybrid heuristic algorithm for solving the model is described in detail. Parameter experiment of hybrid algorithm and analysis of combination effect of variable neighborhood operators is described in detail in Section 4. The comparative analysis of the calculation results with related literatures is reported in Section 5. Section 6 summarizes the optimization results and mentions future works.

II. Problem Description and Model Formulation

A. Problem Description

Capacitated vehicle routing problem (CVRP) is studied in the paper. The vehicle capacity constraint is considered in CVRP, and the loading volume cannot exceed the maximum capacity of the vehicle. After the vehicle departs from the distribution center, the service is launched to the customer point according to the optimized path, and returns to the distribution center after the service is completed. For CVRP, the paper has the following assumptions: (1) all nodes in the distribution network have known information and are fully interconnected, and the distribution routes start and end at the distribution center; (2) customer demand does not exceed the maximum vehicle capacity and should be satisfied by the distribution vehicles at one time; (3) the distribution vehicles are all of the same type; (4) the distribution center has sufficient capacity to send multiple vehicles to meet the demand of all customer points.

The CVRP could be defined on a complete and directed graph $G = (V, E)$, where V is the node set and E is the arc set. In particular, $V = \{0\} \cup V_0(V, E)$, include distribution center {0} and V0, where $V_0 = \{1,2, \cdots, n\}$ is the customer set. $E = \{(i, j) | i, j \in V\}$ is the set of arcs between nodes i and j. c_{ij} denotes the travel cost between nodes i and j, which related only to the distance between two points. $K = \{1,2, \cdots, \varphi\}$ represents the set of available delivery vehicles. The maximum vehicle capacity of the vehicle is Q. Demand of any customer d_i is assumed to be less than the vehicle capacity Q. x_{ijk} is the binary decision variables which indicates that vehicle k drives from node i to j. y_{ik} is the binary decision variables which indicates whether customer i is served by vehicle k.

B. 2.2 Model Formulation

Do not number your paper: All manuscripts must be in English, also the table and figure texts, otherwise we cannot publish your paper. Please keep a second copy of your manuscript in your office. When receiving the paper, we assume that the corresponding authors grant us the copyright to use the paper for the book or journal in question. Should authors use tables or figures from other Publications, they must ask the corresponding publishers to grant them the right to publish this material in their paper. Use italic for emphasizing a word or phrase. Do not use boldface typing or capital letters except for section headings (cf. remarks on section headings, below).

The constraints of the CVRP are shown as follows:

$$\min \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} c_{ij}x_{ijk} \tag{1}$$

$$\text{s.t} \sum_{i\in V}\sum_{j\in V_o} x_{ijk} \cdot d_j \le Q, \forall k \in K \tag{2}$$

$$\sum_{i\in V} x_{ijk} = \sum_{i\in V} x_{jik} = 1, \forall j \in V_o, \forall k \in K \tag{3}$$

$$\sum_{j\in V_o} x_{ojk} = \sum_{j\in V_o} x_{jok} \le 1, \forall k \in K \tag{4}$$

$$\sum_{k\in K} y_{ik} = 1, \forall i \in V_o \tag{5}$$

$$\sum_{j\in V} x_{ijk} = y_{ik}, \forall i \in V_o, \forall k \in K \tag{6}$$

$$\sum_{i\in V} x_{ijk} = y_{jk}, \forall j \in V_o, \forall k \in K \tag{7}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijk} \le |S| - 1, \forall S \subseteq V_o, \forall k \in K \tag{8}$$

$$x_{ijk} \in \{0,1\}, y_{ik} \in \{0,1\}, \forall i, j \in V, \forall k \in K \tag{9}$$

Equation (1) is the objective function that aims to minimize the total cost. Equation (2) is the vehicle capacity constraint, the total demand at customer points on the vehicle path is less than Q. Equation (3) is the vehicle access balance constraint, which guarantees that each customer has and will have only one vehicle to serve them. Equation (4) ensures that each vehicle is assigned to at most one route and that it starts and ends at the distribution center. Equation (5) make sure that each customer belongs to only one route. Equations (6) and (7) relate the decision variables of x_{ijk} and y_{ik} to ensure that the customer must have a path to connect with it when it is served. Equation (8) ensures the elimination of subtours. Finally, Equation (9) are binary decision variables of x_{ijk} and y_{ik}.

III. The Proposed Method

In this section, a hybrid artificial bee colony algorithm with variable neighbor-hood search is proposed to deal with the CVRP. The variable neighborhood search algorithm iteratively solves the problem by a series of systematic changes in the neighborhood structure, and performs a continuous range search based on the variable-neighborhood operator centered on the current solution. The quality of the solution scheme is gradually improved by accepting improved solutions incrementally in the hope of finding a locally optimal solution [20]. In the analysis of combinatorial optimization problems, variable neighborhood search is suitable to be combined with local search algorithms to develop a concise and efficient optimization search through a circular reciprocal logical structural transformation of local search. The artificial bee colony algorithm is a swarm intelligence optimization algorithm designed by imitating the principle of bee colony foraging, and the ABC algorithm has few control parameters and self-organizing features such as distributivity, positive and negative feedback [21]. It is mainly used to achieve iterative population optimization through individual bee division of labor, and embedding variable neighborhood search into the artificial bee colony algorithm can expand the search range and improve the search accuracy, effectively enhancing the overall optimization capability of HABC-VNS algorithm.

A. Framework for hybrid algorithms

Artificial bee colony algorithm (ABC) is a heuristic algorithm proposed by Karaboga [22]. ABC belongs to a class of evolutionary algorithms inspired by the intelligent behavior of bees in searching for nectar sources around their hives. It is worth-while to evaluate the performance of the ABC algorithm for solving the CVRP with embedded variable neighborhood search. ABC consists of three kinds of bees with different functional attributes (employed bees, onlookers and scouts), whose search is mainly realized by honey source finding, honey source evaluation and bee behavior. The algorithm prioritizes the discovery of nectar sources by the employed bees, followed by the local search of nectar sources by the onlookers. When the nectar source reaches the iteration limit, the corresponding employed bee is transformed into a scout bee to search the nectar source again, and the swarm achieves the optimal search in successive iterations. HABC-VNS embeds VNS into the ABC local search link, and starts a systematic neighborhood structure change for some nodes and strings of the nectar source scheme, and VNS expands the search range of ABC bees to realize the continuous search of high-quality nectar sources. And the algorithm is used to jump out of the local optimum by accepting the inferior solution and the perturbation search of the scouts to strengthen the ABC searching ability and make the overall honey source quality continuously improved. The HABC-VNS process is shown in Figure 1.

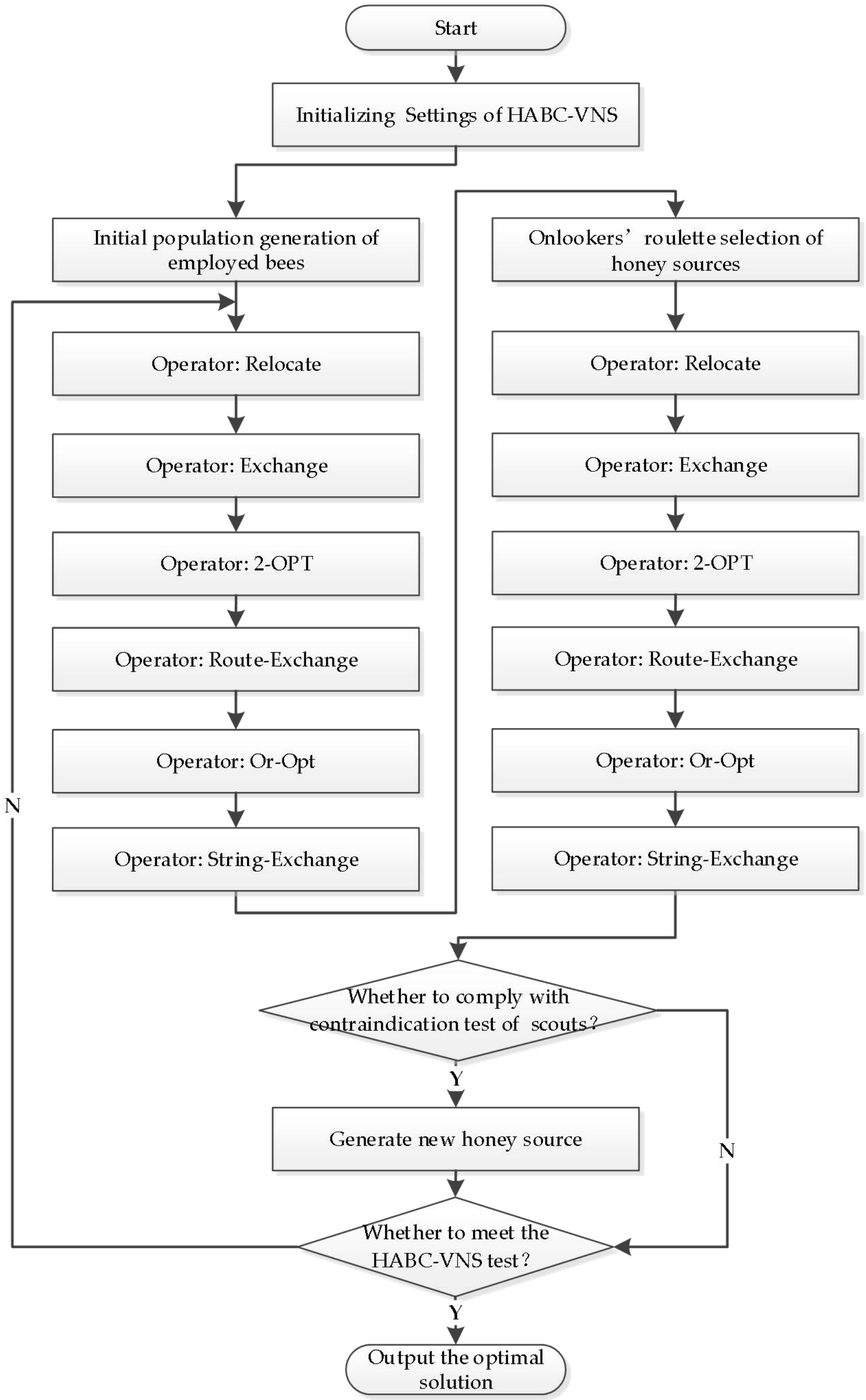

Fig. 1 Flowchart of HABC-VNS

B. 3.2 Variable Neighborhood Search Operators

HABC-VNS mainly sets six variable neighborhood search operators to strengthen the local search ability of the hybrid algorithm in the nectar-seeking phase of the employed bees and onlookers. The six local search operators are divided into three categories according to the object of the transformation operation. Based on node operation, the variable neighborhood search operator sets Insert and Exchange. Based on scheme subpath transformation, the variable neighborhood search operator sets 2-Opt and Route-Exchange. Based on string operation, the variable neighborhood search operator sets Or-Opt and String-exchange. Through systematic structural transformation of nodes, strings and paths, the variable neighborhood search operator can achieve continuous and progressive search optimization and improve HVNS -ABC search efficiency. The specific analysis of the operator is as follows:

(1) Insert. The Insert operation first selects two random points *i* and *j* (*i* is a customer node and *j* is an arbitrary node) in the optimization scheme, then removes *i* from the original path, and subsequently relocates point *i* to either side of node *j*, which is directly adjacent to it.

(2) Exchange. The Exchange operator will select random customer nodes *i* and *j* in the optimization scheme, and just exchange their node positions.

(3) 2-Opt. Based on the number of selected sub-paths within the optimization scheme, the 2-Opt operator includes both single-sub-path operation and two-sub-path operation. For single-sub-path operation, customer nodes *i* and j in the same subpath are selected first, and when *i* and *j* are not adjacent, the paths between *i* and j are directly inverted to make them combine to form a new path. For two-sub-path operation, customer nodes *i* and j in different sub-paths are selected, and the sub-paths where the two points are located are separated separately; subsequently, sequential crossover and reverse-order crossover are executed for the sub-paths where *i* and *j* are located, and the sub-paths are recombined to generate a new scheme by 2-OPT operation. The sequential crossover and reverse-order crossover of two-sub-path operation are shown in Figure 2.

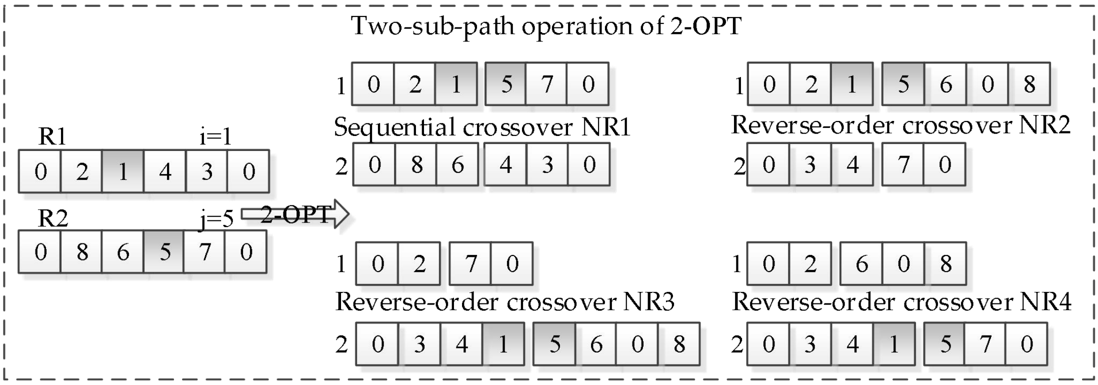

Fig. 2 Two-sub-path operation of 2-OPT

(4) Route-Exchange. Route-Exchange selects the customer nodes *i* and *j* of the optimized scheme that belong to the two sub-paths respectively, and generates a new scheme by exchanging the remaining paths after nodes *i* and *j* in the two sub-paths.

(5) Or-Opt. Or-Opt operation is similar to Insert operation. First, two random points *i* and *j* (*i* is a customer node and *j* is an arbitrary node) in the optimization scheme are selected, and the point *i* and the part of consecutive strings after it in the sub-path where *i* is located are separated. Then, the separated string can be repositioned on both sides of point *j*, so that Or-Opt can achieve partial string reset in the scheme.

(6) String-Exchange. The String-Exchange operation first selects random customer nodes *i* and j in different sub-paths of the optimization scheme, and similar to Or-Opt, separates the part of consecutive strings adjacent to *i* and *j*, respectively, and then just exchanges the two strings.

C. 3.3 Iteration steps of hybrid algorithms

HABC-VNS uses integer coding to construct the honey source solution scheme and selects the insertion method to generate the initial population based on the principle of improved saving algorithm. The insertion cost of customer point r between points i, j is set to $l_{ijr} = l_{ir} + l_{rj} - g * l_{ij} + f * |l_{ir} - l_{rj}|$, g and f are set to be random numbers ($g \in [0,3]$, $f \in [0,1]$) [23,24]. Due to the setting of random numbers g and f, the customer point insertion cost is no longer fixed, which solves the problem of relatively single arrangement of customer points in the point insertion scheme and enhances the diversity of the initial population.

Algorithm 1 displays the pseudo-code of HABC-VNS, and its main steps are outlined as follows.

Algorithm 1：HABC-VNS

```
Start
  Input：Initialize algorithm parameters (instances information, bee
  information nf, perturbation information D1 and D2)
  Output：Current optimal solution
  Generate initial population by random insertion
  while reach maximum number of iterations or find optimal solution
      for each employed bee
          VNS(xi) /* Local search improvement with variable
          neighborhood operators
      end for
      for each onlooker
          Roulette_Selection(xi) /* Roulette selection for solution
          VNS(xi) /* Local search improvement with variable
          neighborhood operators
      end for
      for each scout
          if Solution xi has no improvement in D1 iterations
              LNS(xi) /* Perform large neighborhood search
              perturbations
          elseif Solution xi has no improvement in D2 iterations
          after perturbation of D1
              SR(xi) /* Perturbation: generate new population by
              random insertion
          end if
      end for
  end while
End
```

The specific algorithm execution steps are as follows.

Step1 Algorithm initialization

Initialize the algorithm parameters, set the number of employed bees and onlookers (nf) to be the same, and set the number of individuals in the honey bee population to be 2* nf. The hybrid algorithm sets a fixed number of iterations as the termination condition, and the failure path penalty cost is directly related to the total cost. D1 and D2 are perturbation contraindication values for scouts disturbance actions.

Step2 Honey source search optimization: employed bees

The insertion method is designed to generate the initial population of employed bees based on the principle of the improved saving algorithm. Its random setting ensures the diversity, and then the initial scheme is locally searched by variable neighborhood operators.

Step3 Honey source search optimization: onlookers

When the employed bees complete the search for superiority, onlookers will select the higher quality solution to continue the variable neighborhood search based on the roulette setting. The roulette selection probability is linearly related to the cost of the nectar solution. The meritocratic iteration of onlookers can increase the search intensity of the local optimal solution.

Step4 Inspection and Perturbation: scouts

The hybrid algorithm requires a taboo test after each iteration of the scheme to determine whether the current nectar source has improved or not. When the number of consecutive iterations of a scheme reaches the perturbation contraindication value D1 and still has no improvement, the scheme will be perturbed by the scout bee for the perturbation operation. The perturbation will perform the operation by the large neighborhood search operator (LNS), which will jump out of the local optimum on the basis of retaining most of the current scheme information, and continue to carry out the iteration of finding the optimum. When the solution reaches the perturbation contraindication value D2 after the first perturbation, the employed bee will be called to regenerate a new solution for replacement based on the improvement insertion meth-od. Perturbations of scout enhance the ability of HABC-VNS to jump out of the local optimum.

Step5 Inspection: HABC-VNS

HABC-VNS sets the maximum number of iterations as the algorithm termination condition. When the number of iterations exceeds 1500, HABC-VNS will stop running and output the current optimal solution.

IV. Simulation Experiment and Result Analysis

A. Description of Instance and Experimental Environment

In order to verify the superiority of the proposed hybrid algorithm, 74 instances from three standard benchmarks are selected from Capacitated Vehicle Routing Problem Library (CVRPLIB) to test HABC-VNS, which consists of Set A (27), Set B (23), and Set P (24) [25]. The customer points in Set A are randomly distributed in the grid of [1,110] × [1,110], and their demands obey the uniform distribution U(1,30). The customer points in Set B are also located in the grid of [1,100] × [1,100], and have the characteristics of uniform discrete distribution of clusters, and their demands are dis-tributed with the same characteristics as Set A. The instances in Set P is obtained by integrating the instance information of Set A, Set B and Set E and modifying the demand quantity and vehicle capacity of the relevant customer points. Figure 3 (a/b) gives a schematic diagram of customer locations and their demand information under two typical distribution characteristics: random distribution and uniform discrete distribution of clusters (the circles in the figure represent customer point locations, whose size is related to demand, and the squares are distribution center locations).

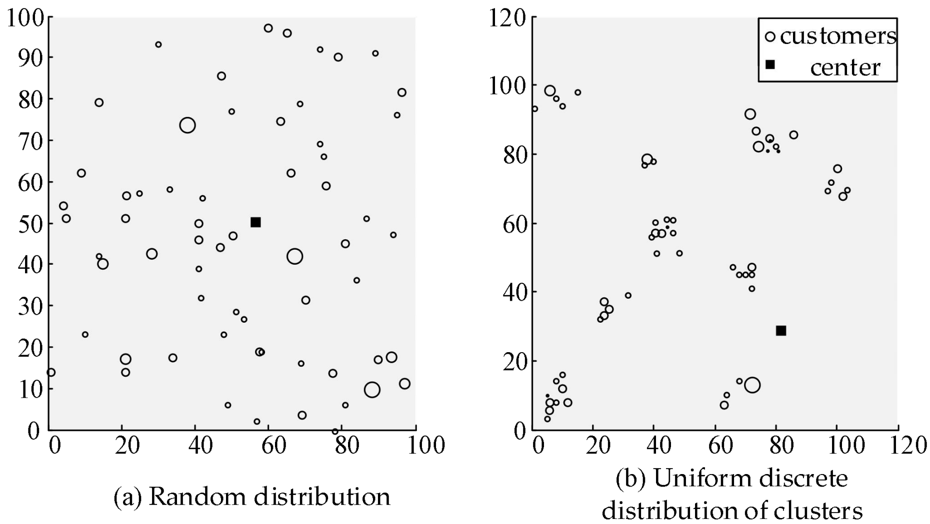

Fig. 3 Figure for representative instances with heterogeneous distribution characteristics

The cases in Set A, Set B and Set P contain sets of customers with different distribution characteristics, and the size of the nodes in the set of instances varies from 18 to 101, through which the performance of HABC-VNS can be tested with good results. The algorithm in the paper is implemented using MATLAB r2014a with Window10 operating system, 16.00 GB of computer memory, and Core i7-10875H CPU with 2.3 GHz. A total of 10 runs were performed for each instance.

B. Parameter Experiment and Analysis

Taguchi optimization is a method to minimize the effect of noisy parameters and determine the appropriate value settings of important parameters based on the idea of robustness [26]. Minitab software is used to generate and analyze the Taguchi results. Based on Taguchi optimization method, the paper applies Minitab to develop numerical analysis of key parameters of HABC-VNS and determine its better default parameter settings.

The ABC algorithm has few control parameters, and it is found that the number of employed bees and onlookers (n_f), perturbation contraindication values D_1 and D_2 have a greater impact on the performance of HABC-VNS, and the above three key global parameters are selected in the paper to develop an orthogonal test of scale L9 (3^3). Taguchi method generates an appropriate orthogonal array with 9 experiments (L9) which provides a balance among the orthogonal index, parameters. The selected HABC-VNS parameters and levels are shown in

Table 1, and the setting of L9 (3^3) can balance the orthogonal indicators, parameters, and their levels, and the effective parameters and their levels can be identified. The orthogonal tables are shown in Table 2.

Table 1 Three Scheme comparing

Factors	Level 1	Level 2	Level 3
n_f	20	25	30
D_1	30	40	50
D_2	20	30	40

In Taguchi optimization test, medium-scale instances (A-n55-k9/P-n50-k7/B-n57-k9) were selected from Set A, Set P and Set B. The deviation of the optimal value of HABC-VNS and its average value from the known optimal solution of instances was examined for different parameter level settings in relatively suitable time. The calculation results are also shown in Table 2, where BK is the best-known solution of the instance, Best and Average are the best found solutions over all runs and average of solutions over all runs. Besides, %Dev indicates the deviation of Best from BK, StdDev indicates the deviation of Average from BK. A total of 10 runs were performed for each experiment.

Table 2 The orthogonal array L9 and results of experiments

Experiments	n_f	D_1	D_2	A-n55-k9		P-n50-k7		B-n57-k9	
				%Dev	StdDev	%Dev	StdDev	%Dev	StdDev
1	1	1	1	0.19	1.16	0.18	1.28	1.69	2.09
2	1	2	2	0.19	1.21	0.90	1.37	1.81	2.15
3	1	3	3	0.19	1.16	0.18	1.28	1.81	2.15
4	2	1	2	0.37	0.75	0.18	0.88	1.63	2.05
5	2	2	3	0.56	1.01	0.36	1.03	1.63	2.11
6	2	3	1	0.56	1.14	0.36	0.99	1.63	2.15
7	3	1	3	0.93	1.41	0	0.97	1.88	2.12
8	3	2	1	0.93	1.38	0.36	1.03	1.75	2.08
9	3	3	2	0.93	1.46	0.36	1.05	1.88	2.11

The experimental results of the above parameters were analyzed by Minitab, and the mean values of the solutions obtained from the experiments under different level settings for the three instances in Table 2 were transferred to the S/N ratio under Taguchi optimization for evaluation. The response values of S/N for number of employed bees and onlookers (n_f), perturbation contraindication values D_1 and D_2 with different parameter settings are shown in Table 3, and the main effects of each parameter and level value on the mean value and S/N are shown in Figure 4. Delta in Table 3 is the difference between the maximum and minimum values of S/D response values for different level settings of the parameters. Given that the objective function of the model is total cost minimization, the Smaller criterion is used in the calculation of S/D response values in the results of the Minitab algorithm.

Table 3 S/N ratios obtained from the Taguchi experimental design

Level	n_f	D_1	D_2
1	-61.42	-61.41	-61.42
2	-61.41	-61.42	-61.41
3	-61.42	-61.42	-61.42
Delta	0.01	0.01	0
Rank	1	2	3

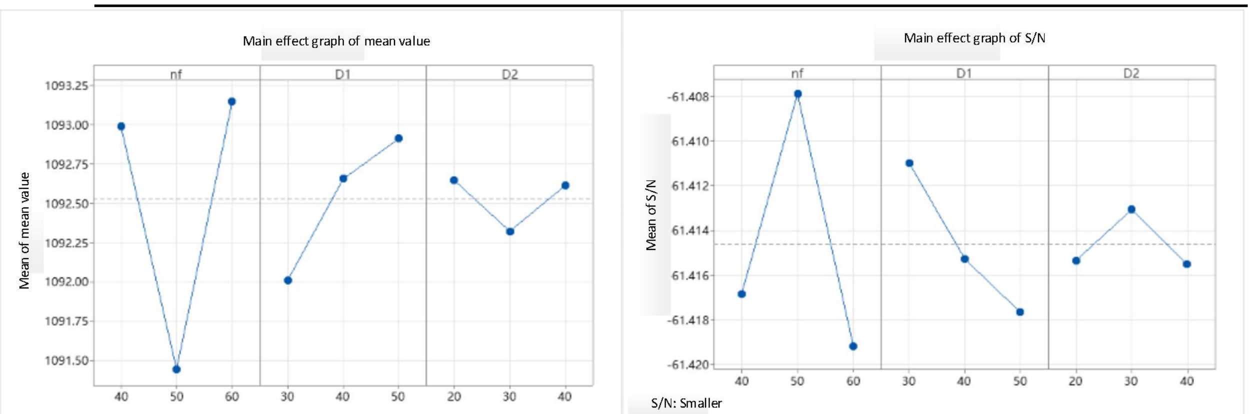

Fig. 4 Main effect diagram of mean and S/N ratio obtained from the Taguchi experimental design

According to the parameter ranking results shown in Table 3, it can be observed that number of employed bees and onlookers (n_f) has the greatest impact on the calculation results for each case, followed by the perturbation contraindication values D_1 and D_2. Furthermore, in terms of minimizing the objective function, the optimal value for the parameter level occurs when the S/N ratio response value is maximized [27]. Therefore, as shown in the main effects diagram in Figure 4, the level values for n_f, D_1 and D_2 should be set to 2 (25), 1 (30), and 2 (30), respectively, which are the most appropriate values. Under this level value setting, the mean index effect of the instance mean values in the mean main effects diagram in Figure 4 is better. The aforementioned parameter values of 2/1/2 correspond to Experiment 4 in Table 2. The results of solving the instances in Table 2 indicate that this parameter setting can achieve better solutions for A-n55-k9/P-n50-k7/B-n57-k9, with certain advantages in optimization and mean value solving. Therefore, the algorithm will adopt the aforementioned parameter level values for subsequent calculations.

C. Comparative Analysis of Variable Neighborhood Search Operators

In order to verify the effectiveness of the variable neighborhood search operators in HABC-VNS, the paper conducts experimental tests on different subsets of decision strategies by combining the operators. The decision strategies are divided into three subsets based on the operator's operation object: node strategy DS1, string strategy DS2, and sub-path transformation strategy DS3. DS1 includes point reset operator (Insert) and point exchange operator (Exchange), DS2 includes string reset operator (Or-Opt) and string exchange operator (String-Exchange), and DS3 includes 2-Opt operator (2-Opt) and path exchange operator (Route-Exchange).

To test the effectiveness of the decision subset combination in a relatively short period of time, the paper conducted comparative analysis on %Dev and StdDev for three medium-sized test instances (A-n55-k9/P-n50-k7/B-n57-k9) selected in Section 4.2. The decision subset combinations are divided into three categories: (1) single decision subset application; (2) combination of two decision subsets; and (3) combination of three decision subsets. The test was repeated 10 times for each test instance, and the population individuals under the decision subset combination were called 6 times for variable neighborhood search operators in each iteration. The calculation results are shown in Table 4.

Table 4 S/N ratios obtained from the Taguchi experimental design

Experiments	**Decision subsets**	**A-n55-k9**		**P-n50-k7**		**B-n57-k9**	
		%Dev	**StdDev**	**%Dev**	**StdDev**	**%Dev**	**StdDev**
1	DS_1	0.47	1.00	**0**	0.69	1.19	1.83
2	DS_2	1.30	1.77	1.08	1.91	1.56	1.97
3	DS_3	5.13	6.91	5.05	7.33	2.13	3.99
Average	Single decision subset	2.30	3.23	2.04	3.31	1.63	2.60
4	DS_1/ DS_2	0.56	1.28	0.36	0.87	1.56	1.92
5	DS_1/ DS_3	**0.37**	0.93	**0**	0.69	1.69	1.80
6	DS_2/ DS_3	1.03	1.59	1.26	1.82	1.56	1.93
Average	Two decision subsets	0.65	1.27	0.54	1.13	1.60	1.88
7	DS_1/ DS_2/ DS_3	**0.37**	**0.87**	**0**	**0.63**	**1.00**	**1.52**

Based on the results of %Dev and StdDev presented in Table 4, it can be concluded that when the algorithm is applied to the test instances by using different combinations of three decision subsets, the algorithm's optimization ability and solution stability are better when all three subsets are used compared to using only two subsets or a single subset. Furthermore, the performance of using two subsets is better than using a single subset. When testing with a single decision subset, the node strategy DS_1 per-forms better in terms of solution performance than the string strategy DS_2 and the sub-path transformation strategy DS_3, especially when DS_3 is used alone, its effect is not ideal.

Regarding instance optimization, the combination of all three subsets shows the strongest optimization ability for HABC-VNS. When the instance is relatively simple to solve, satisfactory solutions can also be obtained by using the combination of two subsets. When testing with a random combination of two decision subsets, the algorithm's performance is best under the combination of DS_1/DS_3, which leads to more stable solutions. Overall, the difference in solution performance between the combinations of DS_1/DS_2, DS_1/DS_3, and DS_2/DS_3 is relatively small, and using any combination of two subsets yields better results than using a single subset for algorithm optimization.

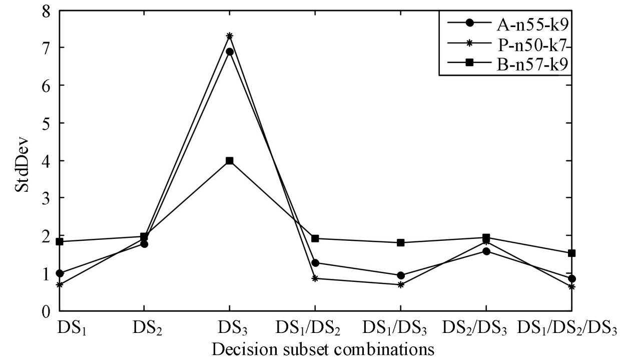

Fig. 5 Trend diagram of instances' mean deviation under decision subsets

Figure 5 shows the trend of average deviation in HABC-VNS when applying different subsets of decision strategies, based on the results in Table 4. For HABC-VNS, the following observations are summarized: (1) setting node neighborhood search for solution can improve local search efficiency, especially for smaller problem sizes, and quickly find the optimal solution; (2) the performance of using only the subpath neighborhood search is poor because it lacks node operations, making optimization more difficult. However, when combining subpath with node and string neighborhoods for local search, the algorithm's performance can be effectively improved; (3) as the problem size increases, the setting of string and subpath neighborhood operators can cause disturbance, avoiding node neighborhood operators from being trapped in local optimal solutions. Therefore, the combination of node, string, and subpath neighborhood transformation operations can effectively improve the algorithm's overall optimization ability and enhance its stability. Based on these observations, subsequent HABC-VNS testing will use a combination of three decision strategy sub-sets.

D. Comparisons with Other Algorithms

Table 5 presents the results of the 27 instances in Set A, including those from the literature [28-30] and the proposed algorithm in this paper. BK denotes the known optimal solution, while Best, Worst, and Average represent the best, worst, and average values obtained by the algorithms, respectively. %Dev indicates the deviation from BK (in %, $\%Dev = (Best - BK)/BK$), and StdDev represents the deviation of the average solution from BK (in %, $StdDev = (Average - BK)/BK$).

Table 5 The results comparison of set A CVRP instances

Instances	BK	AGGWOA[28]			H3[29]			LNS-ACO[30]			HABC-VNS			
		Best	StdDev	%Dev	Best	StdDev	%Dev	Best	StdDev	%Dev	Best	Worst	StdDev	%Dev
A-n32-k5	**784***	**784***	0.43	**0**	**784***	10.77	**0**	**784***	**0**	**0**	**784***	**784***	**0**	**0**
A-n33-k5	**661***	**661***	0.08	**0**	**661***	9.73	**0**	**661***	**0**	**0**	**661***	**661***	**0**	**0**
A-n33-k6	**742***	**742***	0.09	**0**	**742***	5.45	**0**	**742***	**0**	**0**	**742***	**742***	**0**	**0**
A-n34-k5	**778***	**778***	0.62	**0**	**778***	12.22	**0**	**778***	**0**	**0**	**778***	**778***	**0**	**0**
A-n36-k5	**799***	**799***	0.94	**0**	**799***	11.75	**0**	**799***	**0**	**0**	**799***	**799***	**0**	**0**
A-n37-k5	**669***	**669***	0.54	**0**	693	17.83	3.59	**669***	**0**	**0**	**669***	**669***	**0**	**0**
A-n37-k6	**949***	**949***	1.2	**0**	967	4.28	1.90	**949***	**0**	**0**	**949***	**949***	**0**	**0**
A-n38-k5	**730***	**730***	0.5	**0**	**730***	22.29	**0**	**730***	**0**	**0**	**730***	**730***	**0**	**0**
A-n39-k5	**822***	**822***	0.49	**0**	853	2.71	3.77	**822***	**0**	**0**	**822***	825	0.11	**0**
A-n39-k6	**831***	833	0.69	0.24	**831***	16.03	**0**	**831***	0.67	**0**	**831***	834	0.20	**0**
A-n44-k6	**937***	**937***	1.26	**0**	**937***	11.18	**0**	**937***	1.53	**0**	**937***	938	0.02	**0**
A-n45-k6	**944***	953	1.88	0.95	977	8.83	3.50	958	1.61	1.48	**944***	955	0.54	**0**
A-n45-k7	**1 146***	**1 146***	1.35	**0**	1155	8.4	0.79	**1 146***	1.64	**0**	**1 146***	1 157	0.49	**0**
A-n46-k7	**914***	**914***	1.12	**0**	**914***	4.57	**0**	**914***	0.84	**0**	**914***	917	0.15	**0**
A-n48-k7	**1 073***	**1 073***	2.4	**0**	1115	14.65	3.91	1 084.1	0	1.03	**1 073***	**1 073***	**0**	**0**
A-n53-k7	**1 010***	1 017	2.08	0.69	**1 010***	23.64	**0**	**1 010***	2.01	**0**	**1 010***	1 017	0.35	**0**
A-n54-k7	**1 167***	1 176	2.03	0.77	**1 167***	11.45	**0**	**1 167***	1.94	**0**	**1 167***	1 175	0.26	**0**
A-n55-k9	**1 073***	1 074	1.77	0.09	1 085	16.99	1.12	**1 073***	0.28	**0**	**1 073***	1 074	0.03	**0**
A-n60-k9	**1 354***	1 359	2.29	0.37	**1 354***	12.18	**0**	**1 354***	1.28	**0**	1 358	1 359	0.31	0.30
A-n61-k9	**1 034***	1 035	1.56	0.09	1 048	13.46	1.35	1 067	2.74	3.19	1 035	1 035	0.10	0.10
A-n62-k8	**1 288***	1 298	2.87	0.78	1 349	15.84	4.74	1 308	1.36	1.55	1 303	1 311	1.50	1.16
A-n63-k9	**1 616***	1 629	2.76	0.8	1 645	8.56	1.79	1 649	2.5	2.04	1 632	1 636	1.07	0.99
A-n63-k10	**1 314***	1 323	2.36	0.68	1 351	28.87	2.82	1 329	1.42	1.14	1 317	1 320	0.38	0.23
A-n64-k9	**1 401***	1 414	2.97	0.93	1 472	10.03	5.07	1 415	1.89	1	1 417	1 422	1.34	1.14
A-n65-k9	**1 174***	1 178	2.05	0.34	1 202	38.96	2.39	1 185	1.83	0.94	1 178	1 180	0.38	0.34
A-n69-k9	**1 159***	1 163	2.2	0.35	1 206	25.35	4.06	1 170	2.53	0.95	1 165	1 172	0.88	0.52
A-n80-k10	**1 763***	1 783	3.77	1.13	**1 763***	10.62	**0**	1 815	2.23	2.95	1 789	1 804	1.99	1.47
Average			1.57	0. 30		13.95	1.51		1.05	0.6			0.37	0.23

The * denotes the known optimal solution.

From the results in Table 5, both the AGGWOA and H3 algorithms in the literature can obtain the optimal solutions for 13 test cases, with a success rate of 48.14%. The LNS-ACO algorithm can obtain the optimal solutions for 17 test cases, with a success rate of 62.96%. The proposed HABC-VNS algorithm can obtain the optimal solutions for 18 test cases, with a success rate of 66.67%. This represents an improvement of 18.53% over AGGWOA and H3, with a significantly more advantageous success rate. Compared to LNS-ACO, the improvement is 4.67%, demonstrating a certain advantage.

In terms of the statistical analysis of the average minimum deviation of the optimal solutions obtained by the algorithms, HABC-VNS (0.23%) is superior to H3 (1.51%), LNS-ACO (0.6%), and AGGWOA (0.3%). Except for the slightly larger deviation of the optimal solution obtained by the H3 algorithm, the performance differences between HABC-VNS, AGGWOA, and LNS-ACO algorithms in terms of obtaining the optimal solutions for the test cases are relatively small.

In terms of the average values obtained by the algorithms for the test cases, neither AGGWOA nor H3 algorithms can stably obtain the optimal solutions for any test case, with average deviations of 1.57% and 13.95%, respectively. The H3 algorithm performs poorly in terms of solution stability, with the average deviation of the test case solution significantly deviating from the optimal value. The LNS-ACO algorithm can stably obtain the optimal solutions for 9 test cases, with an average deviation of 1.05%. HABC-VNS can stably obtain the optimal solutions for 9 test cases, with an average deviation of 0.37%. HABC-VNS demonstrates significant advantages in terms of solution stability, with the exception of the A-n80-k10 test case, where the deviation of the average solution is within 1.5% in all test cases of different scales.

From the results and analysis in Table 5, it can be concluded that HABC-VNS per-forms relatively well in terms of obtaining the optimal solutions and average deviation of the solutions. The difference between its StdDev (0.37%) and %Dev (0.23%) is only 0.14%, which can simultaneously ensure its solution stability and accuracy based on obtaining the optimal solution for some test cases.

Table 6 The results comparison of set P CVRP instances

Instances	BK	AGGWOA[28]			H3[29]			LNS-ACO[30]			HABC-VNS			
		Best	StdDev	%Dev	Best	StdDev	%Dev	Best	StdDev	%Dev	Best	Worst	StdDev	%Dev
P-n16-k8	**450***	**450***	**0**	**0**	**450***	0.01	**0**	**450***	**0**	**0**	**450***	**450***	**0**	**0**
P-n19-k2	**212***	**212***	**0**	**0**	**212***	1.48	**0**	**212***	**0**	**0**	**212***	**212***	**0**	**0**
P-n20-k2	**216***	**216***	**0**	**0**	**216***	0.92	**0**	**216***	**0**	**0**	**216***	**216***	**0**	**0**
P-n21-k2	**211***	**211***	**0**	**0**	**211***	1.15	**0**	**211***	**0**	**0**	**211***	**211***	**0**	**0**
P-n22-k2	**216***	**216***	**0**	**0**	**216***	1.46	**0**	**216***	**0**	**0**	**216***	**216***	**0**	**0**
P-n22-k8	**603***	—	—	—	—	—	—	**603***	**0**	**0**	**603***	**603***	**0**	**0**
P-n23-k8	**529***	**529***	**0**	**0**	**529***	0.41	**0**	**529***	**0**	**0**	**529***	**529***	**0**	**0**
P-n40-k5	**458***	**458***	0.79	**0**	**468***	8.33	2.18	**458***	**0**	**0**	**458***	**458***	**0**	**0**
P-n45-k5	**510***	**510***	0.74	**0**	**510***	3.95	**0**	**510***	**0**	**0**	**510***	**510***	**0**	**0**
P-n50-k7	**554***	**554***	1.76	**0**	567	6.27	2.35	**554***	1.55	**0**	**554***	557	0.20	**0**
P-n50-k8	**631***	**631***	1.35	**0**	658	5.96	4.28	643	1.9	1.81	**631***	635	0.41	**0**
P-n50-k10	**696***	704	3.25	1.15	**696***	3.18	**0**	**696***	1.24	**0**	**696***	707	0.89	**0**
P-n51-k10	**741***	743	2.33	0.27	**741***	3.82	**0**	747	1.51	0.81	742	747	0.43	0.13
P-n55-k7	**568***	570	2.09	0.35	**568***	4.61	**0**	**568***	**0**	**0**	**568***	575	0.63	**0**
P-n55-k8	**588***	—	—	—	**588***	4.63	**0**	**588***	0.26	**0**	**588***	**588***	**0**	**0**
P-n55-k10	**694***	696	1.74	0.29	**694***	4.37	**0**	**694***	1.36	**0**	698	706	1.11	0.58
P-n55-k15	**989***	—	—	—	—	—	—	**989***	**0**	**0**	**989***	**989***	**0**	**0**
P-n60-k10	**744***	753	3.16	1.21	769	5.61	3.36	755	1.55	1.48	749	763	1.83	0.67
P-n60-k15	**968***	976	3.53	0.83	**968***	3.38	**0**	977	1.73	0.93	**968***	988	0.96	**0**
P-n65-k10	**792***	—	—	—	823	6.66	3.91	800	1.73	1.01	806	821	1.77	3.16
P-n70-k10	**827***	—	—	—	873	9.27	5.56	837	2.2	1.21	847	863	3.43	2.42
P-n76-k4	**593***	**593***	1.8	**0**	**593***	10.50	**0**	598	1.39	0.84	605	612	2.53	2.02
P-n76-k5	**627***	628	2.89	0.16	**627***	11.13	**0**	635	2.87	1.25	641	645	2.55	2.23
P-n101-k4	**681***	—	—	—	**681***	15.64	**0**	—	—	—	689	710	3.38	1.17
Average			1.41	0.24		5.13	0.98		0.84	0.41			0.89	0.46

The * denotes the known optimal solution.

Table 6 presents the calculation results of 24 instances in Set P, including algorithms from literature [28-30] and our algorithm. For some corresponding instances, the results of certain comparison algorithms were not provided. The symbols in the table have the same meaning as those in Table 5. Comparing the results in Table 6, AGGWOA can obtain 11 optimal solutions out of the 18 instances, H3 can obtain 17 optimal solutions out of the 22 instances, LNS-ACO can obtain 15 optimal solutions out of the 23 instances, and HABC-VNS can obtain 16 optimal solutions out of the 24 instances. In terms of the number of optimal solutions obtained, except for AGGWOA, the success rates of H3, LNS-ACO, and HABC-VNS are similar.

Regarding the statistical analysis of the average minimum deviation of the optimal solutions obtained by the algorithms listed in the literature, AGGWOA (0.24%) performs the best, followed by LNS-ACO (0.41%) and HABC-VNS (0.46%), while H3 (0.98%) performs moderately well. All algorithms perform well in obtaining optimal solutions for the given P instances.

Regarding the average values of the instances obtained by the algorithms, AGGWOA and H3 have moderate solving performance, with average deviations of 1.41% and 5.13%, respectively. Although H3 algorithm can obtain more optimal solutions, it fails to stably obtain the optimal solution regardless of the instance size. LNS-ACO and HABC-VNS can stably obtain 11 optimal solutions out of the instances, with average instance average value deviations of 0.84% and 0.89%, respectively. The algorithm performance is comparable in terms of solution stability.

Table 7 The results comparison of set B CVRP instances

Instances	BK	H3[29]			CVRP-FA[31]			MAS-SA-PR[32]		HABC-VNS			
		Best	StdDev	%Dev	Best	StdDev	%Dev	Best	%Dev	Best	Worst	StdDev	%Dev
B-n31-k5	**672***	**672***	6.77	**0**	**672***	**0**	**0**	**672***	**0**	**672***	**672***	**0**	**0**
B-n34-k5	**788***	**788***	16.18	**0**	**788***	0.22	**0**	**788***	**0**	**788***	789	0.06	**0**
B-n35-k5	**955***	**955***	19.75	**0**	**955***	0.01	**0**	**955***	**0**	**955***	956	0.07	**0**
B-n38-k6	**805***	810	5.04	0.62	806	0.15	0.12	**805***	**0**	**805***	807	0.15	**0**
B-n39-k5	**549***	**549***	20.92	**0**	550	0.89	0.18	**549***	**0**	**549***	550	0.05	**0**
B-n41-k6	**829***	**829***	22.16	**0**	**829***	0.10	**0**	**829***	**0**	**829***	834	0.19	**0**
B-n43-k6	**742***	**742***	20.09	**0**	**742***	**0**	**0**	**742***	**0**	**742***	743	0.02	**0**
B-n44-k7	**909***	926	3.69	1.87	**909***	0.47	**0**	**909***	**0**	**909***	**909***	**0**	**0**
B-n45-k5	**751***	**751***	7.58	**0**	**751***	0.43	**0**	**751***	**0**	**751***	752	0.05	**0**
B-n45-k6	**678***	690	22.65	1.77	686	2.18	1.18	**678***	**0**	**678***	680	0.18	**0**
B-n50-k7	**741***	**741***	10.58	**0**	**741***	0.51	**0**	**741***	**0**	**741***	**741***	**0**	**0**
B-n50-k8	**1 312***	1 340	10.03	2.13	1 318	1.34	0.46	1 322	0.76	1 318	1 329	1.03	0.46
B-n51-k7	**1 032***	**1 032***	14.05	**0**	**1 032***	—	**0**	**1 032***	**0**	**1 032***	**1 032***	**0**	**0**
B-n52-k7	**747***	**747***	5.17	**0**	**747***	0.08	**0**	**747***	**0**	751	755	0.78	0.54
B-n56-k7	**707***	**707***	17.76	**0**	709	0.98	0.28	709	0.28	708	717	0.83	0.14
B-n57-k7	**1 153***	**1 153***	36.52	**0**	**1 153***	0.82	**0**	1 157	0.35	**1 153***	**1 153***	**0**	**0**
B-n57-k9	**1 598***	1 636	18.94	2.38	1 610	1.08	0.75	1 611	0.81	1 605	1 617	0.92	0.44
B-n63-k10	**1 496***	**1 496***	19.31	**0**	1 503	2.98	0.47	1 521	1.67	1 508	1 511	0.89	0.80
B-n64-k9	**861***	890	24.66	3.37	862	3.11	0.12	864	0.35	864	874	1.11	0.35
B-n66-k9	**1 316***	1 344	26.77	2.13	1 319	0.67	0.23	1 321	0.38	1 336	1 340	1.67	1.52
B-n67-k10	**1 032***	**1 032***	11.74	**0**	1 042	3.45	0.97	1 039	0.68	1 040	1 051	1.38	0.78
B-n68-k9	**1 272***	1 292	8.29	1.57	1 278	1.27	0.47	1 290	1.42	1 278	1 290	1.02	0.47
B-n78-k10	**1 221***	**1 221***	19.35	**0**	1 224	2.21	0.25	1 239	1.47	1 237	1 249	1.75	1.31
Average			16	0.69		1.04	0.24		0.36			0.53	0.30

The * denotes the known optimal solution.

Table 7 shows the computation results of 23 instances from Set B using algorithms from literature [29, 31-32] and the proposed algorithm in this paper. Among these algorithms, H3, CVRP-FA, MAS-SA-PR, and HABC-VNS could find the optimal solutions for 15, 11, 13, and 13 instances, respectively. Except for the optimal solutions of Set B instances, HABC-VNS outperformed H3 in 8 instances and CVRP-FA in 6 in-stances, as well as MAS-SA-PR in 7 instances. The average minimum deviation of the optimal solutions obtained by CVRP-FA (0.24%) and HABC-VNS (0.3%) was better than that of H3 (0.69%) and MAS-SA-PR (0.36%). Regarding the average instance values obtained by the algorithms, MAS-SA-PR did not provide data on the average deviation values, and H3 could not stably obtain the optimal solutions for Set B instances, with an average deviation of the instance mean value reaching 16%. Additionally, 7 instances had a deviation of the solution mean value from the optimal value of over 20%, indicating poor stability in the solution. CVRP-FA could stably obtain the optimal solutions for 2 instances with an average instance mean value deviation of 1.04%, while HABC-VNS could stably obtain the optimal solutions for 5 instances with an average instance mean value deviation of 0.53%.

Subfigures 6(a), 6(b), and6(c) show the distribution of customer points for the A-n55-k9, B-n57-k7, and P-n50-k7 problem instances, respectively. Subfigures 6(d), 6(e), and 6(f) depict the corresponding optimal solutions and costs. It is evident from the figures that the proposed algorithm can effectively satisfy the requirements of geographical distribution of customer points, service sequences, and demand sizes, resulting in reasonable optimization solutions. Based on a comparative analysis of the series of problem instances in Sets A/B/P, the proposed HABC-VNS algorithm exhibits good performance in solving small and medium-sized problems, outperforming other benchmark algorithms in terms of both the optimal solution and the average solution. Moreover, the algorithm shows better stability in problem-solving, making it an effective approach to solving the capacitated vehicle routing problem.

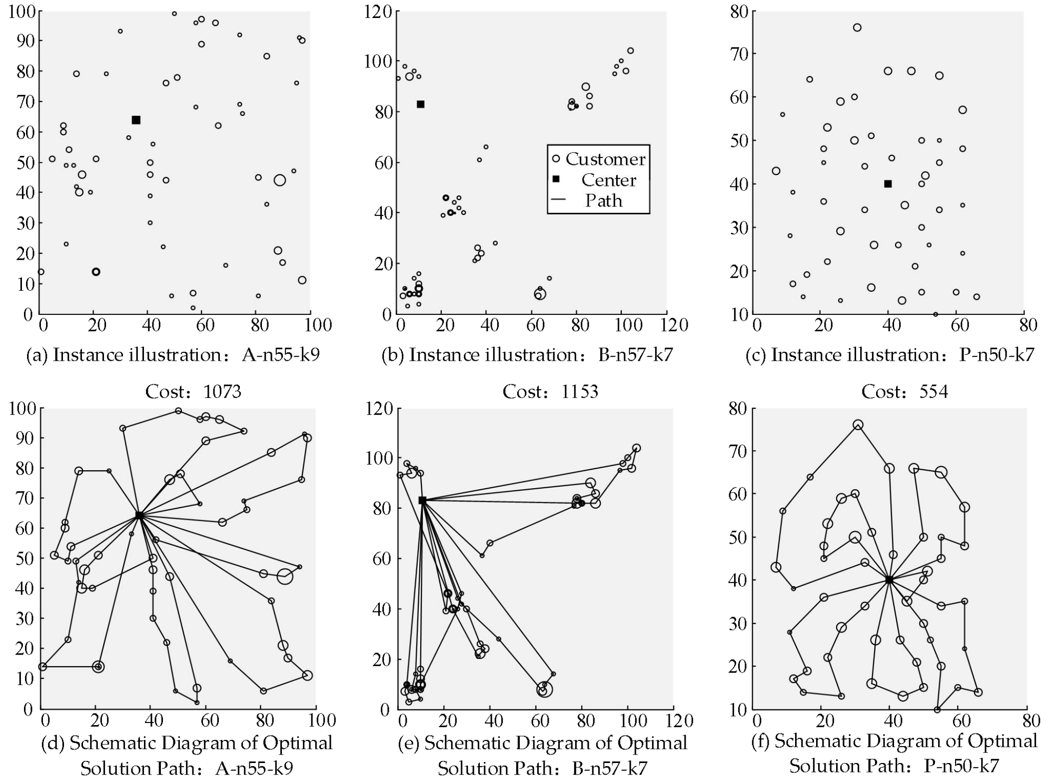

Fig. 6 Schematic diagram of example and the optimal route: A-n55-k9/ B-n57-k7/ P-n50-k7

V. CONCLUSIONS

For the vehicle routing problem with capacity constraints, we established and analyzed a classical vehicle routing problem model and designed a hybrid artificial bee colony algorithm with variable neighborhood search to solve it. We conducted experiments and analyses on the parameter settings of the hybrid algorithm based on Taguchi optimization, and designed six variable neighborhood operators including node, string, and path transformations according to the problem characteristics. The variable neighborhood search algorithm has a simple structure and is easy to integrate with other heuristic algorithms for iterative optimization. Based on the validation of test instances and comparison analysis with literature, our hybrid artificial bee colony algorithm performed well in obtaining optimal solutions and ensuring solution stability, with high precision and strong global search capability. However, as the problem size increases, the effectiveness of the algorithm decreases, and further research is needed to effectively solve large-scale vehicle routing problems.

ACKNOWLEDGEMENTS

This research was funded by the National Social Science Fund of China (20CJY027), Ministry of education of Humanities and Social Science project (20YJC630070), Scientific Research Fund of Liaoning Educational Department (L2019050), Fund of Liaoning Federation of Social Sciences (2022lslqnrcwtkt-39) and Social Science Planning Fund of Liaoning, China (L22BGL034).

REFERENCES

[1] Alcaraz, J. J.; Caballero-Arnaldos, L.; Vales-Alonso, J.. Rich vehicle routing problem with last-mile outsourcing decisions[J]. Transportation Research Part E: Logistics and Transportation Review, 2019, 129:263-286.

[2] Baldacci, R.; Toth, P.; Vigo, D.. Exact algorithms for routing problems under vehicle capacity constraints[J]. Annals of Operations Research, 2010, 175:213-245.

[3] Machado A. M.; Mauri G. R.; Boeres M.; Rodrigo A. R.. A new hybrid matheuristic of GRASP and VNS based on constructive heuristics, set-covering and set-partitioning formulations applied to the capacitated vehicle routing problem[J]. Expert Systems with Applications, 2021, 184, 115556.

[4] Liu T.; Luo Z X.; Qin H.; Lim A.. A branch-and-cut algorithm for the two-echelon capacitated vehicle routing problem with grouping constraints[J]. European Journal of Operational Research, 2018,266:487-497.

[5] Marinelli M.; Colovic A.; Dell'Orco M.. A novel dynamic programming approach for Two-Echelon Capacitated Vehicle Routing Problem in City Logistics with Environmental considerations[J]. Transportation Research Procedia, 2018, 30:147-156.

[6] Yang W.; Ke L.; Wang D. Z. W.; Jasmine S. L. L.. A branch-price-and-cut algorithm for the vehicle routing problem with release and due dates[J]. Transportation Research Part E: Logistics and Transportation Review, 2021, 145:102167.

[7] Yu Y.; Wang S.; Wang J.; Huang M.. A branch-and-price algorithm for the heterogeneous fleet green vehicle routing problem with time windows[J]. Transportation Research Part B: Methodological, 2019, 122:511-527.

[8] Tolga B,; Luis G.; Antonio M. S.; Juan-José S. G.. Balanced vehicle routing: Polyhedral analysis and branch-and-cut algorithm[J]. European Journal of Operational Research, 2019, 273:452-463.

[9] Claudio C.; Rafael M.. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints[J]. Discrete Optimization, 2014, 12:129-146.

[10] Babak R.; Frederico G. G.; Rasul E.; Pauline C. H.. Combining genetic local search into a multi-population Imperialist Competitive Algorithm for the Capacitated Vehicle Routing Problem [J]. Applied Soft Computing, 2023, 142: 110309.

[11] Arit T.; Petcharat R.. Hybrid modified ant system with sweep algorithm and path relinking for the capacitated vehicle routing problem[J]. Heliyon, 2021, 7: e08029.

[12] Dimitra T.; Manousos R.; Magdalene M.; Yannis M.. A firefly algorithm for the environmental prize-collecting vehicle routing problem[J]. Swarm and Evolutionary Computation, 2020, 57:100712.

[13] Li Y. B.; Hamed S.; Mostafa Z.. An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives[J]. Journal of Cleaner Production, 2019, 227: 1161-1172.

[14] Vincent F. Y.; Redi A. A. N. P.; Yang C. L.; Eki R.; Budi S.. Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem[J]. Applied Soft Computing, 2017, 52:657-672.

[15] Imen O.; Moez H.. Honey bee cooperative hyperheuristic[J]. Procedia Computer Science, 2021, 192: 2871-2880.

[16] Chi J. S.; He S. W.. Pickup capacitated vehicle routing problem with three-dimensional loading constraints: Model and algorithms[J]. Transportation Research Part E: Logistics and Transportation Review, 2023, 176:103208.

[17] Vincent F. Y.; Renan M.; Tsao Y. C.. A reduced variable neighborhood search-based hyperheuristic for the shelf space allocation problem[J]. Computers & Industrial Engineering, 2020, 143: 106420.

[18] André S.; Daniel M.; Matthias W.. Route learning: a machine learning-based approach to infer constrained customers in delivery routes[J]. Transportation Research Procedia, 2020, 46: 229-236.

[19] Zhang K.; He F.; Zhang Z. C.; Lin X.; Li M.. Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach[J]. Transportation Research Part C: Emerging Technologies, 2020, 121: 102861.

[20] Sadati M.; Atay B.; Aksen D.. An Efficient Variable Neighborhood Search with Tabu Shaking for a Class of Multi-Depot Vehicle Routing Problems[J]. Computers & Operations Research, 2021,133:231-244.

[21] Zhou J. J.; Yao X. F.; Felix T. S. Chan.; Lin Y. Z.; Jin H.; Gao L.; Wang X. P.. An individual dependent multi-colony artificial bee colony algorithm[J]. Information Sciences, 2019, 485:114-140.

[22] Karaboga, D. ; Basturk, B.. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm[J]. Journal of Global Optimization, 2007, 39:459-471.

[23] Zachariadis E.; Tarantilis C.; Kiranoudis C. A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service[J]. Expert Systems with Applications, 2009, 36:1070-1081.

[24] Paessens H. The savings algorithm for the vehicle routing problem[J]. European Journal of Operational Research, 1988, 34:336-344.

[25] Capacitated vehicle routing problem library (CVRPLIB) [Online] Available: http://vrp.atd-lab.inf.puc-rio.br/index.php/en/.

[26] Tsai P. W . ; Pan J. S. ; Chen S. M. Liao B. Y.. Enhanced parallel cat swarm optimization based on the Taguchi method[J]. Expert Systems with Applications, 2012, 39:6309–6319.

[27] Al-Tabeeb A.; Mohsen A. M.; Ghallab A. S.. An improved hybrid firefly algorithm for capacitated vehicle routing problem[J]. Applied Soft Computing, 2019, 84(11):1-9.

[28] Huang G. W.; Cai Y. G.; Qi Y. H.; Chen H. R.; Wang S. H.. Adaptive Genetic Grey Wolf Optimizer Algorithm for Capacitated Vehicle Routing Problem[J]. Acta Electronica Sinica, 2019, 47(12): 2602-2610.

[29] Yildirim U.; Kuvvetli Y.. Solution of capacitated vehicle routing problem with invasive weed and hybrid algorithms [J]. International Journal of Industrial Engineering Computations, 2021, 12(4):441-456.

[30] Akpinar S. Hybrid large neighborhood search algorithm for capacitated vehicle routing problem[J]. Expert Systems with Applications, 2016, 61(11):28-38.

[31] Asma M. A.; Abdulqader M. M.; Abdullatif G.. An improved hybrid firefly algorithm for capacitated vehicle routing problem[J]. Applied Soft Computing, 2019, 84(11): 10572-10581.

[32] Arit T.; Petcharat R.. Hybrid modified ant system with sweep algorithm and path relinking for the capacitated vehicle routing problem[J]. Heliyon, 2021, 9(7):1-15.