Journal of Computers Vol. 31 No. 2, 2020, pp. 35-44
doi:10.3966/199115992020043102005

Research on Hybrid Artificial Intelligence Optimization Algorithm for Grain Transportation

Yiting Zhen, Kang Zhou*, Haocheng Fang, Long Hu, Zhuo'er Dai, Wanying Liang, Sisi Zhou

School of Math and Computer, Wuhan Polytechnic University, Wuhan 430023, Hubei, China
zhoukang_wh@163.com

Received 13 January 2019; Revised 12 March 2019; Accepted 12 March 2019

Abstract. Grain Transportation Optimization Problem (GTOP) is a typical NP-complete problem. In this paper, a mathematical model of GTOP is constructed and a hybrid artificial intelligence optimization algorithm (HAIOA) for GTOP is proposed. In the algorithm, ant colony algorithm (ACA) is introduced into tabu search algorithm (TSA): the optimal solution of ACA used as calculation starting point of TSA can improve the quality of initial solution of TSA; the optimal pheromone of ACA used to guide neighborhood search of TSA can improve the quality of TSA can improve search quality of TSA. In ACA, adaptive expectation heuristic factor and initial solution distance formula are introduced to obtain some better results with certain differences. In addition, the search method of TSA is improved. The simulation results show that compared with other algorithms for GTOP, hybrid artificial intelligence optimization algorithm (HAIOA) has the advantages of less time consuming and better comprehensive performance, which improves the performance of the algorithm.

Keywords: ant colony algorithm (ACA), grain transportation optimization problem (GTOP), optimal pheromone, tabu search algorithm (TSA), vehicle routing problem (VRP)

1 Introduction

As an important part of the grain industry, grain logistics has a significant impact on economic development in China. Nowadays, there are still some problems in China's grain logistics system, such as high logistics cost, high empty car rate and low circulation efficiency. Therefore, it is urgent to arrange the grain transportation route scientifically and rationally, reduce transportation cost and increase the effective utilization rate of grain logistics vehicles. It is the key to improve the grain transportation efficiency and reduce the cost of grain circulation. The grain transportation problem can be transformed into vehicle routing problem (VRP), so this paper discusses the grain transportation related issues by studying VRP. VRP is a key issue in the field of logistics research, which was first proposed by Dantzig and Ramser in 1959 [1], and it is a kind of important combinatorial optimization problem. VRP has been widely used in logistics distribution, transportation and other fields. The research shows that VRP belongs to NP-complete problem [2], which is a typical complex combinatorial optimization problem. Only when there are fewer paths and path points can the exact solution be obtained. As soon as question was raised, it soon attracted attention of experts and scholars in operational research, graph theory, computer application and so on [3].

In view of VRP, many scholars at home and abroad have done some related researches. Dantzig and Ramser presented an algorithm when they first proposed VRP in 1959. Later they improved the algorithm, but both algorithms only focued on the complete composition of route without considering the minimization of the objective function (path saving). Shen used BP neural network to get a better vehicle routing [4], but the network parameter robustness was poor, and neural network might eventually converge to the local minimum solution, or even infeasible solution. Kirkpatrick proposed SA algorithm and applied it to solve combinatorial optimization problems [5]. However, due to time constraints, the

* Corresponding Author

approximate optimal solution was usually obtained. Gendreau [6] and others first applied TSA proposed by Glover [7] to solve VRP, but TSA has strong dependence on the initial solution, and could not be searched in parallel, so its global performance was poor. Cambardella [8] and Dorigo [9] gave ACA which has strong robustness and can be searched in parallel, but it is easy to fall into local optimum, slow convergence speed and even stagnation. In recent years, with the research of many scholars, many algorithms have been put forward, such as K.M. Salama's Classification with cluster-based Bayesian multi-nets using Ant Colony Optimisation [10], Koushik Venkata Narasimha's An ant colony optimization technique for solving min–max Multi-Depot Vehicle Routing Problem [11], S.R. Balseiro's An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows with Time Windows [12], Dervis Karaboga's A novel clustering approach: Artificial Bee Colony (ABC) algorithm [13], etc.

Aiming at the characteristics of tabu search algorithm (TSA) and ant colony algorithm (ACA), a hybrid artificial intelligence optimization algorithm (HAIOA) is proposed in this paper. The algorithm improves the data structure and algorithm structure of TSA and the shortcomings of ACA, respectively. At the same time, ACA is introduced to compensate the disadvantage of TSA relying on the initial solution. The results show that hybrid artificial intelligence optimization algorithm (HAIOA) has faster convergence speed and stronger robustness than Basic ACA. Compared with Basic TSA, hybrid artificial intelligence optimization algorithm (HAIOA) is more effective for solving combinatorial optimization problems. Compared with other improved hybrid algorithms, it can be used to solve large-scale problems. Compared with other algorithms, it also has better performance. At the same time, after introducing pheromone, the accuracy has been improved on the basis of not being introduced.

2 Mathematical Model of Grain Transportation Optimization Problem

This paper mainly introduces a simple grain transportation optimization problem with capacity constraints. In this model, there are only one grain depot, and there are $\boldsymbol{n}$ other grain supply points. The cost of the unit distance traveled by the vehicle is $\boldsymbol{c}$, the distance between $\boldsymbol{i}$ and $\boldsymbol{j}$ points is $\boldsymbol{d_{ij}}$, the demand of grain at point $\boldsymbol{i}$ is $\boldsymbol{q_{ij}}$, neighborhood search for the maximum load is $\boldsymbol{Q}$, and distribution point number is l, 2,... $\boldsymbol{N}$, define variable $\boldsymbol{x_{ijk}}$ is equal to 1 when the k^{th} car runs from point i to point j, otherwise it will equal 0; and variable $\boldsymbol{y_{ik}}$ is equal to 1 when the k^{th} car has served point i, otherwise it will equal 0.

Objective function
$$\text{Min } Z = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{m} c d_{ij} x_{ijk} \quad \textbf{(1)}$$

$$\text{s.t.} \begin{cases} \sum_{k=1}^{m} y_{ik} = 1 & i \in \{1,\ldots,n\} \quad \textbf{(2)} \\ \sum_{i=0}^{n}\sum_{k=1}^{m} x_{ijk} = 1 & j \in \{0,\ldots,n\} \quad \textbf{(3)} \\ \sum_{j=1}^{n}\sum_{k=1}^{m} x_{ijk} = 1 & i \in \{0,\ldots,n\} \quad \textbf{(4)} \\ \sum_{i} q_i y_{ik} \le Q & \quad \textbf{(5)} \\ \sum_{i=1}^{n}\sum_{k=1}^{m} x_{0ik} = \sum_{j=1}^{n}\sum_{k=1}^{m} x_{j0k} & \quad \textbf{(6)} \end{cases}$$

The expression (3) belongs to the objective function of model, which represents total cost generated during the delivery process. Subsequently (4), (5) and (6) are constraints to ensure that a path point is accessed only once by a vehicle, (7) are constraints to ensure that each vehicle is not overloaded in transit, (8) are constraints to indicate that the vehicle departs from the grain depot and finally returns to the grain depot.

3 Hybrid Artificial Intelligence Optimization Algorithm

3.1 Analysis of Hybrid Artificial Intelligence Optimization Algorithm

TSA is a global neighborhood search algorithm, which uses tabu table to avoid local optimization and thus obtain the final global optimal, and inferior solution can also be accepted. However, TSA has some deficiencies. First, it has strong dependence on the initial solution, and poor initial solution will reduce convergence speed of TSA. Second, iterative search process is to move one solution to another, thereby reducing probability of getting the global optimal solution.

ACA introduces heuristic information and uses principle of positive feedback. It has a great advantage in convergence speed of algorithm. Distributed calculating, inherently parallel, different individuals continuously carry on information exchange and transmission, can cooperate with each other and help to find better solutions. But at the same time, it is easy to get the local optimal solution and lose the global optimal solution. In addition, the calculation amount of the algorithm is relatively large, and result takes longer. It is difficult to determine control parameters of algorithm, and convergence is very slow. After a certain degree, search stagnation will occur.

Hybrid artificial intelligence optimization algorithm (HAIOA) is based on TSA. First, the ants of ACA are discretely distributed in solution space, and enable adaptive expectation heuristics. ACA performs large-scale search to achieve a reasonable iteration of ACA pheromone to obtain global optimal solution and hybrid algorithm global pheromone network, then TSA takes global better solution as initial solution. Under the guidance of global pheromone network, different neighborhood search methods are implemented, neighborhood solution set of the initial solution is obtained, and values of the expected functions of all solutions in the neighborhood solution set are calculated and sorted from superior to bad. The global optimal solution is obtained by comparing with expected value of the optimal solution in TSA bulletin board. Hybridizing these two algorithms not only utilizes ACA's large-scale search capability, but also maximizes the local search ability of TSA, and promotes strengths and avoids shortcomings, which makes algorithm greatly improved in terms of convergence and avoiding local minimum.

3.2 Hybrid Artificial Intelligence Optimization Algorithm for VRP

The specific steps of the hybrid artificial intelligence optimization algorithm (HAIOA) to solve VRP:

Step 1: Set $\boldsymbol{t} = 0$, $\boldsymbol{Nc} = 1$, $\boldsymbol{\tau_{ij}(t)} = const$, $\Delta\, \tau_{ij} = 0$, $\boldsymbol{Xa} = \boldsymbol{\phi}$. ACA maximum cycle number is $\boldsymbol{Nc_{ACA}}$, ACA ant number is $\boldsymbol{m}$, path point number is $\boldsymbol{n}$.

Step 2: $\boldsymbol{m}$ ants are randomly placed in $\boldsymbol{n}$ path points.

Set $\boldsymbol{s} = 1$

For $\boldsymbol{k} = 1$ to $\boldsymbol{m}$ do

$\boldsymbol{allowed_k} = \{1, 2, \ldots, n\}$

Randomly get $\boldsymbol{l} \in \boldsymbol{allowed_k}$

Set $tabu_k(s) \leftarrow l$

Set $\text{allowed}_k = \text{allowed}_k - \{1\}$

Step 3: Update the route of $\boldsymbol{m}$ ants.

For $\boldsymbol{s} = 2$ to $\boldsymbol{n}$ do

For $\boldsymbol{k} = 1$ to $\boldsymbol{m}$ do

Calculate transition probability $\boldsymbol{p^k_{ij}(t)}$, according to formula (7) and

$$\boldsymbol{p^k_{ij}(t)} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{S \in allowed_k} [\tau_{is}(t)]^{\alpha} \cdot [\eta_{is}(t)]^{\beta}} & j \in allowed_k \\ \mathbf{0} & j \notin allowed_k \end{cases}$$

$$\boldsymbol{allowed_k} = \{1, 2, \ldots, n\} - \boldsymbol{tabu_k},\ \eta_{ij}(\boldsymbol{t}) = 1/\boldsymbol{d_{ij}}$$

Randomly get $\boldsymbol{l} \in \boldsymbol{allowed_k}$, according to $\boldsymbol{p^k_{ij}(t)}$

Set $\boldsymbol{tabu_k(s) \leftarrow l}$

Set $\boldsymbol{tabu_k(s) \leftarrow l}$, until $\boldsymbol{tabu_k}$ $(\boldsymbol{Nc})$ is filled, jump out of the loop.

Set $\text{allowed}_k = \text{allowed}_k - \{1\}$

Step 4: Calculate and update the pheromone concentration.
For $k = 1$ to m do
Calculate $\Delta\, \tau_{ij}$:

$$\Delta\, \tau_{ij} = \sum(\Delta\, \tau^{k}_{ij} | k = 1, 2, \ldots, m)$$

$$\Delta \tau_{ij} = \begin{cases} Q / L_k & (i, j) \subset tabu_k(s) \\ 0 & (i, j) \not\subset tabu_k(s) \end{cases}$$

where $(i, j) \subset tabu_k(s)$ means i, j is the same as the two adjacent points in the path encoding stored in $tabu_k(s)$, otherwise it will be expressed as $(i, j) \not\subset tabu_k(s)$.
Calculate $\tau_{ij}(t + n)$:

$$\Delta \tau_{ij}(t + n) = \rho \times \tau_{ij}(t) + \Delta \tau_{ij} \; (\rho \in (0,1))$$

Set $t = t + n$
If $Nc < Nc_{ACA}$ and not all ants choose the same path, then
For $k = 1$ to m do
If $tabu_k(s)$ satisfies the similarity requirement and distance, then $Xa \leftarrow tabu_k(s)$
Set $Nc \leftarrow Nc + 1$
For $k = 1$ to m do
Set $s = 1$
Set $tabu_k(s) = \phi$
Go to Step2

Step 5: Instantiation.
Set $D(X_i) = \{\phi,\ \phi,\ \phi\}$, $G(X_i) = \{\phi,\ \phi\}$, tabu length is L, TSA maximum cycle number is Nc_{TSA}.
Set initial solution X ($X \in Xa$)
Calculate fitness value function $f(X)$
Set $X_{best} \leftarrow X$

Step 6: For X ($X \in Xa$) do
(1) Set $Nc = 1$, $X_i \leftarrow X$
(2) Randomly select the *adjacent-swap* operator, the *insert* operator, the *inverse* operator and the *2-opt random exchange* operator as *neighborhood operator*
(3) For $\mathbf{k} = 1$ to $\mathbf{l}$ do neighborhood operator
Calculate neighborhood operator($\mathbf{X_i}$) = ($\mathbf{X_i^j}$, (u, v), $\mathbf{f(X_i^j)}$)

where if the neighborhood search is *adjacent-swap*, *insert*, or *2-opt random exchange*, calculate the competitiveness C of each path point of the path (according to (8)), and calculate the path point to be operated by the greedy algorithm.

Set $\mathbf{D(X_i)} \leftarrow$ neighborhood operator($\mathbf{X_i}$)
(4) Calculate min($D(X_i)$) = (X_i^j, (u^j, v^j), $f(X_i^j)$)
(5) If $(u^j, v^j) \in G(X_i)(1)$ and $f(X_i^j) > f(X_{best})$, then
Set $D(X_i) = D(X_i) - (X_i^j, (u^j, v^j), f(X_i^j))$
Go to (4) of Step6
(6) Set $X_i \leftarrow X_i^j$
If $f(X_i^j) < f(X_{best})$, then set $X_{best} \leftarrow X_i^j$
For any k ($((u, v)_k, l_k) \in G(X_i)$) do
If $(u^j, v^j) = (u, v)_k$, then $G(X_i) = G(X_i) - ((u, v)_j, l_j)$
Set $l_j \leftarrow l_j - 1$
If $l_j = 0$, then $G(X_i) = G(X_i) - ((u, v)_j, l_j)$
Set $G(X_i) \leftarrow ((u^j, v^j), L)$
(7) If $Nc < Nc_{TSA}$, then go to (2) of Step6

Step 7: The global optimal solution is output and ends.

3.3 The Specific Improvement of Hybrid Artificial Intelligence Optimization Algorithm

3.3.1 Definition of Path Similarity

For VRP, traditional definition method cannot fully reflect the difference between bit path coding. So the purpose of defining distance is to distinguish differences in path coding. This paper introduces similar fragment distance and redefines concept of distance. For example, path code $a[5] = (2, 5, 3, 4, 1)$ and $b[5] = (1, 2, 5, 3, 4)$. For traditional method, distance between path encoding a and b is 5. In fact, difference between path encoding a and b is very small. When the sliding distance is taken, distance between is 2. Therefore, in the TSA of HAIOA, distance needs to be redefined.

Definition 1: For the set $S = \{1, 2, \ldots, n\}$, path coding $a[n] = (a_1, a_2, \ldots, a_n)$ and path encoding fragment $p[k] = (p_1, p_2, \ldots, p_k)$,

(1) If $\exists r(r \in N), p_1 = a_r + 1, p_2 = a_r + 2, \ldots, p_k = a_r + k$;

So we call p contained in a, marked as $p \subseteq a$.

(2) We call set $S(p) = S - \{p_1, p_2 \ldots, p_k\}$ is the complement set of $p[k]$.

(3) For $q(q \in S(p))$, $p' = q + p[k]$ is code fragment obtained from extended code word q on the left of $p[k]$; $p' = q + p[k]$ is code fragment obtained from extended code word q on the right of $p[k]$.

Definition 2: For path encoding a, b and a path encoding fragment p_0, if $p_0 \subseteq a \wedge p_0 \subseteq b \wedge \forall q(q \in S(p) \wedge (p' = q + p_0 \vee p' = p_0 + q) \rightarrow p' \not\subset a \vee p' \not\subset b)$, then we call p_0 very similar fragments of a and b, marked as $p_0 \subseteq a \cap b$.

Definition 3: For path coding $a[n] = (a_1, a_2, a_3, \ldots, a_n) = (t_1, t_2, t_3, \ldots, t_m)$ and $b[n] = (b_1, b_2, b_3, \ldots, b_n) = (s_1, s_2, s_3, \ldots, s_m)(n \geq m$ and $n, m \in N^+)$, If $\forall t_i \exists s_j$(only) have $t_i = s_j \subseteq a \cap b$, It is specified that distance between a and b similar fragments $\rho(a, b)$ is m.

3.3.2 Adaptive β of ACA

The expected heuristic β represent the relative importance of visibility which reflects the relative importance of enlightening information in guiding ant colony search. If the value is too small, the ant population will be stuck in a pure random search. Usually in this case, it is difficult to find optimal solution. If the value is too large, the state transition probability is closer to greedy rule. The convergence speed is accelerated, randomness is not high, and the local relative optimum is easily obtained. Therefore, adaptive β is added in this paper:

$$\beta = \frac{(Max[\beta] - Min[\beta])(N_c)}{N_{c\max}} + Min[\beta] \tag{7}$$

$\beta \in [10, 25]$, $\boldsymbol{Nc}$ is current iteration number, $\boldsymbol{Nc_{max}}$ is the maximum iteration number of ACA.

3.3.3 The Initial Solution of TSA Comes From the Results of ACA

The choice of initial solution has a great influence on the result of TSA. Basic TSA is to randomly obtain an initial solution and conduct neighborhood search, which cannot guarantee accuracy of final solution. The initial solution of TSA in this paper is from the global superior solution obtained by ACA. This lays a good foundation for neighborhood search of TSA and the precision is improved.

3.3.4 TSA Introduces Pheromone Network

The pheromone network is introduced into TSA to guide TSA search. The higher pheromone concentration is, the easier path to be selected by ants. In order to improve probability of selected fragments with low pheromone concentration, a competitive C and greedy algorithm are set up for customer points in each path, so that TSA could converge rapidly.

The method of calculating pheromone concentration by ACA:

For k = 1 to m do, firstly, move the k^{th} ant from the path point l to n. And then calculate total path length L_k of the k^{th} ant. Finally, update found shortest path.

The total amount of information is a constant Q, which is the amount of pheromone in path walked by ants. L_k represents the length of path taken by the k^{th} ant in this cycle.

TSA method of calculating competitiveness:

A path is (1, 2, …, i, j, k, …, n-1, n), The pheromone between the path points is $\boldsymbol{\tau}$, then the competitiveness of the j^{th} path point is C_j:

$$C_j = \frac{1}{\tau_{ij} + \tau_{jk}} \tag{8}$$

According to the competitiveness of each path point, there is a directional selection that requires neighborhood operation. If the randomly selected operator is an *adjacent-swap* operator or an *inverse* operator, use the competitiveness of this path. If a *insert* operator is selected, the competitiveness corresponding to each path point is not used.

3.3.5 The Extension of Neighborhood Search Method of TSA

In this paper, the neighborhood search method of TSA, in addition to the *2-opt random exchange* operator, also joins three new neighborhood search methods to enrich the search scope, and has a great increase in breadth and depth of search, which will reflect the improvement of algorithm precision.

The initial route as shown in Fig. 1, three new neighborhood search methods are as follows:

(1) ADJACENT-SWAP operator: Select any two path points in the initial route of the point i and point j, position of point i and point j after exchange formed a new route;

(2) INVERSE operator: Select any two path points in the initial route of the point i and point j, the route fragment between point i and point j is flipped over;

(3) INSERT operator: After inserting the path point i into any path point j-1.

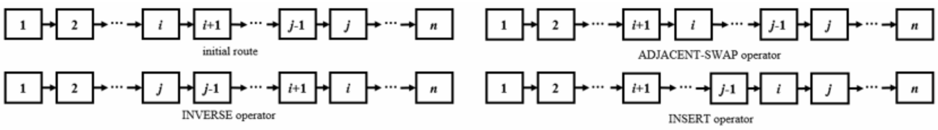

Fig. 1. Neighborhood Search methods

3.4 The Advantage of Hybrid Artificial Intelligence Optimization Algorithm for Solving VRP

HAIOA not only exploits the large-scale search ability of ACA, but also makes the local search ability of TSA play to extreme, so that HAIOA can be improved in convergence and avoiding local minimum. ACA is used first and TSA is used later. This method not only reduces the number of repeated calls to TSA, but also reduces the amount of calculation, and makes full use of the advantages of both. The TSA relies too much on initial solution, and now ACA bulletin board is used to get better solution which accords with difference as the initial solution of TSA, and the pheromone table obtained by ACA iteration is used to guide tabu search, so the iteration speed of TSA will be accelerated and the local optimum will be achieved quickly.

4 Algorithm Simulation Experiment and Comparison

4.1 Experiment Parameter Setting

Because GTOP can be abstracted as VRP, this paper tests proposed algorithm using the open standard test study (benchmark instances). The data originates from the internationally recognized VRP database (http://www.branchandcut.org/). The experimental operating environment is Inter (R) Core (TM) i5-4200h CPU 2.8GHz, 4.00GB Memory, Windows 10 64-bit operating system, with Visual C++6.0 programming.

In order to reduce uncertainties as much as possible, each group of experiments was tested 30 times, and then averaged. In order to ensure fairness of experiment, the number of iterations of HAIOA, that of Basic TSA and that of Basic ACA are all equal. For example, if there are $\boldsymbol{n}$ path points, the number of iterations of ACA in HAIOA is $\boldsymbol{Nc_{ACA}}$, the number of iterations of TSA in HAIOA is $\boldsymbol{Nc_{TSA}}$, ACA to get s solutions as initial solution of TSA, then the number of iterations of HAIOA is $\boldsymbol{n} * \boldsymbol{Nc_{ACA}} + s * \boldsymbol{Nc_{TSA}}$; that of Basic ACA is $(\boldsymbol{n} * \boldsymbol{Nc_{ACA}} + s * \boldsymbol{Nc_{TSA}}) / \boldsymbol{n}$; that of Basic TSA is $\boldsymbol{n} * \boldsymbol{Nc_{ACA}} + s * \boldsymbol{Nc_{TSA}}$. The columns

have following meaning:
・Path points: Number of path points;
・m: Number of ants m;
・$Iter_{ACA}$: Number of ACA iterations;
・ISS: Number of TSA initial solution strips;
・Candidate: TSA candidate Set;
・$Iter_{TSA}$: Number of TSA iterations;

Table 1. Number of iterations for HAIOA, ACA and TSA

HAIOA						ACA		TSA	
Path points	m	$Iter_{ACA}$	ISS	Candidate	$Iter_{TSA}$	m	$Iter_{ACA}$	$Iter_{TSA}$	Candidate
≤ 80	50	100	5	n-1	3000	50	(n*100+3000*5) /n	n*100+3000*5	n-1
≤ 120	50	100	5	n-1	5000	50	(n*100+5000*5) /n	n*100+5000*5	n-1
≤ 150	50	100	5	n-1	8000	100	(n*100+8000*5) /n	n*100+8000*5	n-1
≤ 200	50	100	4	n*2	10000	150	(n*100+10000*4) /n	n*100+10000*4	n*2
≤ 262	50	100	4	n*2	15000	200	(n*100+15000*4) /n	n*100+15000*4	n*2

Other parameters in ACA based on experimental simulation, the following results are better for global convergence and convergence speed: heuristic factor = 0.8, pheromone volatilization factor = 0.6.

Compared with Basic ACA, the data of different scale distribution types in standard database were selected, and the best values, average value, relative error and variance of Basic ACA and HAIOA were tested. The specific data is shown in Table 2. The columns have following meaning:
・Example: Numerical example number;
・Known: Known optimal solution;
・Best: Best solution;
・Value: Average value;
・#Error: Relative error;
・Error: Fractional error;
・Time: Average Time;
・Path points: Number of path points;
・Deviation: Standard deviation;

Table 2. Comparison of the results between basic ACA and HAIOA

Example	Known	Basic ACA				HAIOA			
		Best	Value	#Error	Time	Best	Value	#Error	Time
A-n55-k9	1073	1121	1141	6.34%	14	1074	1076	0.28%	3
A-n60-k9	1354	1445	1463	8.05%	16	1363	1376	1.62%	5
B-n56-k7	707	762	772	9.19%	14	718	725	2.55%	3
B-n63-k10	1496	1587	1599	6.89%	17	1541	1549	3.54%	4
B-n78-k10	1221	1291	1315	7.70%	25	1228	1248	2.21%	6
E-n76-k7	682	763	788	15.54%	24	695	712	4.40%	6
E-n101-k14	1071	1225	1250	16.71%	57	1107	1127	5.23%	11
M-n121-k7	1034	1132	1144	10.64%	88	1038	1046	1.16%	25
M-n200-k17	1373	1511	1531	11.51%	485	1355	1395	1.60%	118
P-n50-k8	631	667	678	7.45%	11	633	644	2.06%	2
P-n70-k10	827	914	926	11.97%	27	842	853	3.14%	5
P-n76-k4	593	651	666	12.31%	29	605	618	4.22%	6
G-n262-k25	6119	6413	6477	5.85%	3413	5788	5942	--	354

As can be seen from Table 2, the solution of VRP obtained by using HAIOA is better than that of Basic ACA, and the best solution obtained by the hybrid algorithm is almost close to the known optimal solution, which is much better than Basic ACA. The HAIOA has small relative error between the optimal solution value and the known optimal solution value, and the error increases when the problem scale is enlarged. The relative error shows that HAIOA is better than Basic ACA, and the convergence time of HAIOA is much shorter than that of Basic ACA, the above parameters also fully demonstrate that the hybrid algorithm converges faster and stronger than Basic ACA.

Table 3. After comparing basic ACA with basic TSA, we can show the fine line of HAIOA

Example	Known	Basic TSA				HAIOA			
		Best	Value	#Error	Deviation	Best	Value	#Error	Deviation
A-n55-k9	1073	1075	1078	0.46%	3.26	1074	1076	0.28%	2.90
A-n60-k9	1354	1366	1380	1.92%	9.29	1363	1376	1.62%	7.98
A-n80-k10	1763	1785	1811	2.72%	13.24	1779	1806	2.44%	11.95
B-n50-k8	1312	1325	1332	1.52%	4.05	1319	1327	1.14%	3.11
B-n78-k10	1221	1237	1263	3.44%	9.51	1228	1248	2.21%	7.15
E-n51-k5	521	526	541	3.84%	8.59	525	530	1.73%	6.66
E-n101-k8	817	862	871	6.61%	10.18	845	864	5.75%	9.72
M-n101-k10	820	836	864	5.37%	13.41	829	845	3.05%	11.05
M-n121-k7	1034	1043	1057	2.22%	5.55	1038	1046	1.16%	2.92
P-n50-k8	631	638	653	3.49%	5.72	633	644	2.06%	4.07
P-n55-k7	568	584	598	5.28%	6.50	577	586	3.17%	4.69
P-n101-k4	681	711	727	6.75%	9.21	699	715	4.99%	7.78
G-n262-k25	6119	6114	6264	2.37%	48.85	5788	5942	--	50.71

The experiment extracts 13 examples and 6 different types from various types in VRP database. The minimum scale starts from 45 path points and increases at about every 5 path points. Because the convergence speed of Basic TSA is fast, the computation time of HAIOA is definitely worse than that of Basic TSA, so the comparison of time is not added in Table 2, but the experimental results are analyzed. The computational time of HAIOA is within a reasonable range. It can be seen from Table 2 that Basic TSA is not only difficult to find the optimal solution, but also has poor stability. HAIOA can almost find the optimal solution when the size of problem is very small, and the error is within acceptable range when the size of problem increases. It shows that HAIOA is more effective than the single algorithm in solving combinatorial optimization problems.

Table 4. The results of introduction of pheromone into HAIOA

Example	Known	No pheromone introduced			No pheromone introduced		
		Best	Value	Error	Best	Value	Error
A-n55-k9	1073	1075	1079	0.56%	1074	1076	0.28%
A-n60-k9	1354	1368	1387	2.44%	1363	1376	1.62%
A-n80-k10	1763	1791	1822	3.35%	1779	1806	2.44%
B-n67-k10	1032	1082	1088	5.43%	1046	1081	4.75%
B-n78-k10	1221	1239	1255	2.78%	1228	1248	2.21%
E-n51-k5	521	532	562	7.87%	525	530	1.73%
E-n76-k7	682	698	714	4.69%	695	712	4.40%
M-n101-k10	820	830	848	3.41%	829	845	3.05%
M-n200-k17	1373	1378	1407	2.48%	1355	1395	1.60%
P-n50-k8	631	638	648	2.69%	633	644	2.06%
P-n101-k4	681	704	719	5.58%	699	715	4.99%
G-n262-k25	6119	5824	5976	--	5788	5942	--

From Table 4, we can see that the precision of HAIOA after introducing pheromone has been improved, although the amplitude of the increase is not particularly significant, it still shows that it can guide the algorithm to converge to the optimal solution. It also directly shows that pheromones have certain guiding significance for the subsequent TSA and can accelerate the convergence speed of TSA.

In order to prove that HAIOA in this paper can not only reflect its advantages on the basis of comparison with itself, but also compare with other improved hybrid ant colony algorithms in reference [14], the results are shown in Table 5 and Table 6.

Table 5. Comparison with other improved hybrid ant colony algorithms

Example	Known	HAIOA			Other algorithms [14]		
		Best	Value	Error	Best	Value	Error
A-n32-k5	784	797.87180	800.53534	2.11%	806.08590	812.65976	3.66%
A-n33-k6	742	742.69326	743.48043	0.20%	742.69326	744.34502	0.32%
A-n36-k5	799	802.13180	809.24323	1.28%	817.21146	823.54400	3.07%
A-n37-k6	949	957.02979	957.12351	0.86%	963.00752	971.26462	2.35%
A-n39-k6	831	835.25177	837.57087	0.79%	840.78543	850.19784	2.31%
A-n44-k6	937	938.18128	948.82120	1.26%	948.24541	951.91445	1.59%
A-n63-k9	1616	1643.95232	1653.1508	2.30%	1653.3444	1665.2142	3.05%
A-n80-k10	1763	1779.17694	1805.88780	2.43%	1856.2821	1896.7304	5.27%

It can be seen from Table 5 that HAIOA is not very advantageous when the scale of problem is small, but when the scale of problem increases gradually, HAIOA can show its advantages better. Compared with the algorithm in reference [14], the relative error of the algorithm in this paper will be smaller. The results show that HAIOA can achieve good results in solving large scale problems. In order to show the effectiveness of HAIOA, a comparison is made with the reference [16] hybrid algorithm, including several different heuristic algorithms, and the computational time of the algorithm is also compared. The results are shown in Table 6.

Table 6. HAIOA comparison with other algorithms

Example	Path points	Best	Objective function value				Computing time(s)			
			PSO	Chen	ACPSO	HAIOA	PSO	Chen	ACPSO	HAIOA
A-n33-k5	32	661	661	661	661	662	11	32	9	3
B-n45-k5	44	751	751	751	751	753	17	134	14	8
B-n78-k10	77	1221	1239	1239	1229	1229	41	429	38	18
P-n101-k4	100	681	686	694	683	692	99	978	402	67
F-n135-k7	134	1162	1184	1215	1170	1175	178	1526	1062	114

Table 6 shows that when the number of path points is 32 and 44, respectively, the other three algorithms obtain the optimal solution, HAIOA, although it does not get the optimal solution, the gap from the known optimal solution is basically negligible, and the computing time is the least. When the number of path points is 77, ACPSO has the same optimal solution as this algorithm, but the algorithm in this paper takes less time, and when the number of path points is 100 and 134, respectively. Compared with the other three algorithms, HAIOA still has better performance although it has no better solution than them, but it is also within the acceptable range, and the computation time is much shorter than the other three algorithms.

5 Conclusion

According to the characteristics of VRP, this chapter synthesizes ACA and TSA, combines the global search ability of the former with the local search ability of the latter, and uses ACA stage to produce the initial solution as the initial solution of TSA. In this way, not only the global search ability of ACA is utilized, but also the problem of slow convergence caused by the lack of pheromones in the early stage of ACA is avoided, thus speeding up the convergence speed of hybrid algorithm. In this case, It can not only guarantee the high efficiency of the hybrid algorithm, but also obtain the global optimum, and finally obtain a better path planning effect. The experimental results show that this method is an effective VRP research method.

Acknowledgments

The work was supported by National Natural Science Foundation of China (61179032), the Special Scientific Research Fund of Food Public Welfare Profession of China (201513004-3), the Guiding

Scientific Research Project of Hubei Provincial Education Department (2017078), the Humanities and Social Sciences Fund Project of Hubei Provincial Education Department (17Y071).

References

[1] G.B. Dantzig, J.H. Ramser, The truck dispatching problem, Management Science 6(1)(1959) 80-91.

[2] J.M. Wu, Vehicle routing optimization problem of logistics distribution, Computer Simulation 28(7)(2011) 357-360.

[3] H.F. Zhang, Application of computer ant colony algorithm in logistics distribution routing optimization, [dissertation] Jiangsu, China: Jiangsu University, 2009.

[4] Y. Shen, J.Y. Potvin, J.M. Rousseau, S. Roy, A computer assistant for vehicle dispatching with learning capability, Annals of Operations Research 61(1)(1995) 189-221.

[5] S. Kirkpatrick, C.D. Gelatt, Optimization by simulation annealing, Science 220(4598)(1983) 671-680.

[6] M. Gendreau, A. Hertz, G. Laporte, A tabu search heuristics for the vehicle routing problem, Management Science 40(1994) 1276-1290.

[7] F. Glover, Tabu search: part, ORSA Journal on Computing 1(3)(1989) 135-206.

[8] L.M. Cambardella, M. Dorigo, Solving symmetric and asymmetric TSPs by ant colonies, in: Proc. International Conference on Evolutionary Computation, 1996.

[9] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society 26(1)(1996) 29-41.

[10] K.M. Salama, A. A. Freitas, Classification with cluster-based bayesian multi-nets using ant colony optimisation, Swarm and Evolutionary Computation 18(2014) 54-70.

[11] K.V. Narasimha, E. Kivelevitch, B. Sharma, M. Kumar, An ant colony optimization technique for solving min–max multi-depot vehicle routing problem, Swarm and Evolutionary Computation 13(2013) 63-73.

[12] S.R. Balseiro, I. Loiseau, J. Ramonet, An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows, Computers & Operations Research 38(6)(2011) 954-966.

[13] D. Karaboga, C. Ozturk, A novel clustering approach: artificial bee colony (ABC) algorithm, Applied Soft Computing 11(2011) 652-657.

[14] J.S. Zhang, Research on vehicle problem of logistics distribution based on improved ant colony algorithm, [dissertation] Liaoning, China: Liaoning Technical University, 2014.

[15] Q.Y. Wang, Solving algorithm of vehicle routing problem with soft time windows, Journal of Beijing Information Science and Technology University (4)(2013) 57-59.

[16] X.J. Chen, The application of the hybrid ant colony algorithm on capacitated vehicle routing problem, [dissertation] Anhui, China: Hefei University of Technology, 2016.