IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 19, 2020, accepted October 26, 2020, date of publication November 3, 2020, date of current version November 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3035584

A Slime Mold-Ant Colony Fusion Algorithm for Solving Traveling Salesman Problem

MEIJIAO LIU[1,3], YANHUI LI[1], ANG LI[1,3], QI HUO[1], NING ZHANG[1,3], NAN QU[2], MINGCHAO ZHU[1], AND LIHENG CHEN[1]

[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[2]College of Resources and Environmental Sciences, Jilin Agricultural University, Changchun 130118, China
[3]School of Optoelectronics, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding authors: Nan Qu (qunan.jlau@gmail.com) and Mingchao Zhu (mingchaozhu@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 11672290, in part by the Science and Technology Development Plan of Jilin Province under Grant 2018020102GX, in part by the Jilin Province, and in part by the Chinese Academy of Sciences Cooperation in Science and Technology High-Tech Industrialization Special Funds Project under Grant 2018SYHZ0004.

ABSTRACT The Ant Colony Optimization (ACO) is easy to fall into the local optimum and its convergence speed is slow in solving the Travelling Salesman Problem (TSP). Therefore, a Slime Mold-Ant Colony Fusion Algorithm (SMACFA) is proposed in this paper. Firstly, an optimized path is obtained by Slime Mold Algorithm (SMA) for TSP; Then, the high-quality pipelines are selected from the path which is obtained by SMA, and the two ends of the pipelines are as fixed-point pairs; Finally, the fixed-point pairs are directly applied to the ACO by the principle of fixed selection. Hence, the SMACFA with fixed selection of high-quality pipelines is obtained. Through the test of the chn31 in Traveling Salesman Problem Library (TSPLIB), the result of path length was 15381 by SMACFA, and it was improved by 1.42% than ACO. The convergence speed and algorithm time complexity were reduced by 73.55 and 80.25% respectively. What's more, under the ten data sets of TSPLIB, SMACFA outperformed other algorithms in terms of the path length, convergence speed and algorithm time complexity by comparison experiments. It is fully verified that the performances of SMACFA is superior to others in solving TSP.

INDEX TERMS Slime mold algorithm, ant colony optimization, high-quality pipelines, fixed selection, travelling salesman problem.

I. INTRODUCTION

The traveling salesman problem (TSP) is a typical combinatorial optimization problem in Non-Deterministic Polynomial problems (NP). TSP [1]–[3] is to find a shortest path by visiting n cities in sequence and returning to the city of departure. There are many significant applications of TSP, such as designing the rational road traffic to reduce congestion, planning logistics to reduce costs, setting up nodes of internet to deal with the information flow, etc. In addition, the welding sequence, cable wiring, data clustering and so on can all be effectively solved by TSP. Some heuristic algorithms have yielded the good results in solving TSP such as Genetic Algorithm (GA) [4], [5], Particle Swarm Optimization (PSO) [6], [7], Lin-Kernighan-Helsgaun Solver (LKH) [8]–[10] and etc. Among them, Ant Colony Optimization (ACO) [7], [11]–[14] has been widely applied in the traditional path planning and has good effect. But it is easy to fall into the local optimal solution and its convergence is slow. In recent years, a new heuristic bionic algorithm called Slime Mold Algorithm (SMA) [15]–[20] has emerged, with a strong and unique ability of path optimization. However, the algorithm is not mature and systematic enough due to staring later. Therefore, a Slime Mold-Ant Colony Fusion Algorithm (SMACFA) for fixing selection of high-quality pipelines is proposed. The key to design is that the high-quality pipelines from SMA are directly applied to ACO. SMACFA will enhance the global optimization ability and improve the speed of convergence in solving TSP.

Currently, ACO is typically used due to good optimization ability. The principle of ACO makes use of the pheromones left in searching for food by ants and continuously updates them, and then the good foraging route with more pheromones will be chosen. ACO was first proposed by Dorigo in 1991 [21]. The algorithm's ontology adopts parallel searching method, so it can be applied to all kinds of

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Ali.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

path optimization problems. However, it is easy to appear the stagnation phenomenon, which will cause the risk of falling into local optimum and the tardiness of convergence speed. In order to solve the shortcomings, many classical improvement methods have emerged [21]–[25]: ant colony system (ACS) [22] which adds updated rule of local pheromones; maximum and minimum ant system (MMAS) [23] which limits the accumulation of pheromone; the best and worst ant system (BWAS) [24] which mainly enhances and weakens the pheromones left by the better path and the worse path; The initial pheromone optimization ant colony system [25] which is mainly realized by the unequal distribution of the initial pheromone. Although the algorithm has many advantages, there are still some shortcomings to improve. On the one hand, the result of ACO may be suboptimal due to the random rule of arbitrarily choosing from some better points. Therefore, it is easy to fall into the local optimal solution. On the other hand, the more accurate optimization path is wanted, the more iterations are needed to set due to the average setting of initial pheromone.

The Slime Mold Algorithm (SMA) [18]–[20], [27], [28] is a newly arisen bionic algorithm which is based on the foraging behavior of slime mold in nature. The slime molds expand toward the food sources and contract away from the food during foraging. Then, they can screen out the shortest path to food by continuously moving and changing. In 2000, Nakagaki [26] and his team discovered the magic optimization ability of slime molds and designed an interesting experiment. They placed slime molds in a maze, put the food at entrance and exit, and made slime molds move freely under foraging behavior. After a period of time, the foraging path was formed and that was the optimal solution of the maze problem. In 2007, Tero *et al.* [27], Nakagaki *et al.* [28] completed the tasks of combining the Poisson's Law and the Kirchhoff's Law to construct road network pipelines which are produced in foraging process of SMA. Eventually, a positive feedback mechanism model of "key pipeline key cultivation" about SMA was abstracted by pheromone flow and pipeline conductivity. In 2008, based on the original research, Tero *et al.* [29] proposed the SMA of multi-food source network. In 2010, Aono *et al.* [30] applied SMA to solve TSP. Subsequently, Vincenzo and Bonifaci [31] verified the feasibility and convergence of SMA. What's more, Chinese researchers such as Southwest University, have proposed a strategy by using SMA to optimize the original pheromone of ACO [32]–[35]. However, SMA started later, so the algorithm is not systematic and mature enough and the stability needs to be improved.

Many hybrid methods of ACO are improved from the perspective of pheromones, such as the unequal distribution of initial pheromones or the setting limit of upper and lower pheromones in path selection [32]–[38]. Pheromones belong to the quantity produced by model; therefore, these improved algorithms of depending on the pheromones in the inherent model cannot get rid of the limitations from model. Meanwhile, ACO randomly generates a set of path points

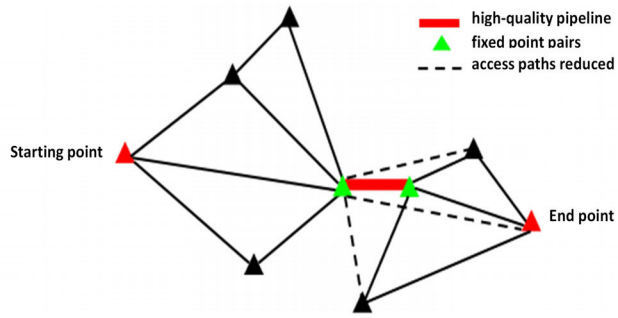


FIGURE 1. Schematic diagram of SMACFA.

to be selected in the point selection processing. So, path points with poor effect are easily selected, which makes the algorithm fall into local optimal. At the same time, as the number of path points increases, the probability of poor path points selected will greatly increase. Therefore, the result of each test will fluctuate greatly, and the probability of good result will decrease, that is, the algorithm has poor robustness. SMA has strong optimization ability and there are many high-quality paths with high flow and short distance. Some paths from SMA are conducive to global optimization due to its good optimization ability, thereby, the high-quality pipelines selection method is proposed.

In this paper, SMA and ACO are combined to design the SMACFA by fixed selection of high-quality pipelines. The schematic diagram is shown in Fig. 1, the high-quality pipelines which are from "key pipeline key cultivation" of SMA are screened out by the relationship among distance, flow and conductivity. The high-quality pipelines are best paths by selecting from optimization result of SMA. Then, the two ends of high-quality pipelines are taken as fixed-point pairs to complete the directional selection in ACO, that is, if the current point is one of the fixed-point pairs, the next point will be directly determined by point pairs. Therefore, the high-quality paths of SMA are applied as fixed path segments in ACO to obtain fusion algorithm. The designed SMACFA can achieve the following advantages:

- The selection of next point could be sub-optimal because of randomness in ACO. But adding the direct selection rule of high-quality pipelines, the probability of the best optimal choice can be increased. At the same time, adding the fixed selection of pipelines will solve local unreasonable selection and improve the algorithm robustness;
- Due to the even distribution of initial pheromones in early stage of ACO, large iterations are needed and the convergence speed is slow. If some point pairs are added in the optimization process, the algorithm will find the next state point earlier. Therefore, the convergence speed will be improved.
- If the fixed selection principle of high-quality pipelines is used, the numbers of points will be reduced. Hence, the algorithm time complexity will be declined.

The structure of rest is as follows: the second part describes two basic algorithms of ACO and SMA; the third part

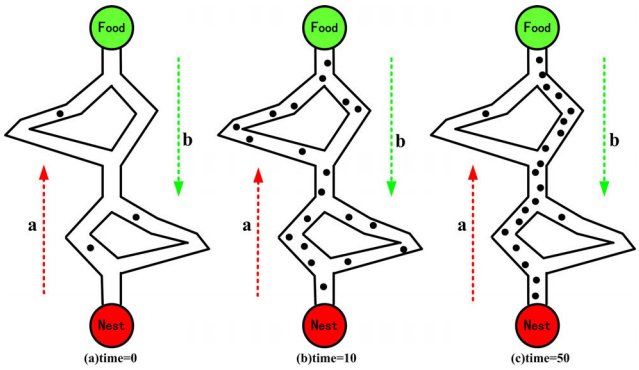


FIGURE 2. The description of ant colony foraging behavior.

describes SMACFA, which is selecting high-quality pipelines through SMA and using the selected pipelines as fixed pipelines in ACO; the fourth part completes the simulation and result analysis; the fifth part is conclusion and summary.

II. ANT COLONY OPTIMIZATION AND SLIME MOLD ALGORITHM

A. ACO

ACO is a bionic heuristic algorithm, which imitates the foraging behavior of ants. In the biological world, if ants find the food source, they will secrete a chemical substance called pheromone on the way from food to nest. Each individual of colony mainly uses the pheromone to exchange information and to find foraging path. As shown in Fig. 2, the basic principle [7], [11]–[14], [21]–[25] is: if the food is found by ants, they will leave pheromones on the path to inform other companions when they return to nest. In the processing of foraging, ants will find food by the pheromones left on path and the pheromones of each path will be accumulated by the number of ants passed. The higher pheromone concentration of path segment is existed, the larger probability is selected. Under the influence of pheromones, the foraging behavior form a positive feedback and ant colonies will gradually select a short path from the anthill to food.

Based on the foraging principle of ant colony, ACO is formed and applied to solve TSP. The specific rules are as follows [11], [14], [22], [24]:

1) The m ants should be placed on n points, and the selection rule of next point is based on the state movement probability. The formula which calculates the state movement probability of the Kth ant from the current point i to the next position point j is illustrated as:

$$P_{ij}^{k} = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{j\in allowed}\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}, & j = allowed \\ 0, & otherwise \end{cases} \tag{1}$$

$$\eta_{ij} = 1/L_{ij} \tag{2}$$

where $\tau_{ij}(t)$ represents the pheromone concentration of the path from point i to point j at time t. α and β are the weight coefficient and the *allowed* is the set of all points which are not accessed. η_{ij} represents the expectation degree from point i to point j.

2) During the optimization process, the points which have been traversed by ants, should be recorded in tabu list. When the ants select the next path point, they not select the points which have been visited. Therefore, the ant does not repeatedly traverse to the selected path point.

3) When the ants finish visiting all the points, they will save the pheromones left in path optimization. Then, the pheromones will be updated according to the updated pheromone formula:

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{3}$$

$$\Delta\tau_{ij}(t) = \sum\nolimits_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \tag{4}$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} 1/L_k, & the\ kth\ ant\ at\ time\ t \\ 0, & otherwise \end{cases} \tag{5}$$

where ρ represents the volatility coefficient of pheromone and the range of ρ is usually $0 < \rho < 1$.

4) The processing of iteration will be completed according to the above method until the requirements are met. Then, the optimal path will be obtained.

B. SMA

The bionic model of SMA [39]–[42] is derived from the foraging behavior. By the expansion behavior, slime molds cover the whole road network firstly. Then, in the foraging process, slime molds continuously move towards food by expansion, and withdraw from the redundant paths which are far away from food by contracting behavior. Finally, the high-quality paths are retained and a path to food source is formed. SMA of the pipeline model can be abstracted by foraging instinct of slime molds. It is shown in Fig. 3, the principle of the model is as following [32]–[35], [41]: Firstly, slime molds build the pipeline network in all directions of food source. Then, according to the length of path, obstacles, and other external environmental factors, slime molds will select the foraging path by the principle of seeking advantages and avoiding disadvantages. The movement of slime molds on the path is constantly changing and the flows of every path is gradually formed. Finally, the pipeline structure which is composed of the optimal path is obtained by flows.

The TSP can be realized by the pipeline model of SMA. The rules are as follows [17], [18], [43], [44]:

1) Variable initialization needs to be completed and the distance L_{ij} between point i and j is required to calculate. The distance calculation formula is:

$$L_{ij} = \left(\left(x_i - x_j\right)^2 - \left(y_i - y_j\right)^2\right)^{1/2} \tag{6}$$

where x_i and x_j represent the abscissa of point i and j, y_i and y_j represents the ordinate of point i and j. L_{ij} is the length of pipe (i, j).

2) Two points as the entry point and the exit point are selected, and the pressure of each point should be calculated according to Kirchhoff's law. The calculation formula of pressure is given as:

$$\sum\nolimits_{i}\frac{D_{ij}}{L_{ij}}\left(P_i - P_j\right) = \begin{cases} -I_0, & for\ j = 1 \\ I_0, & for\ j = 2 \\ 0, & otherwise \end{cases} \tag{7}$$

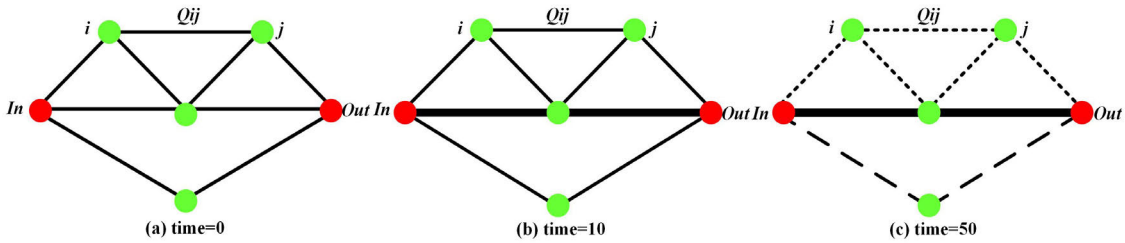


FIGURE 3. Schematic diagram of pipeline model about slime molds.

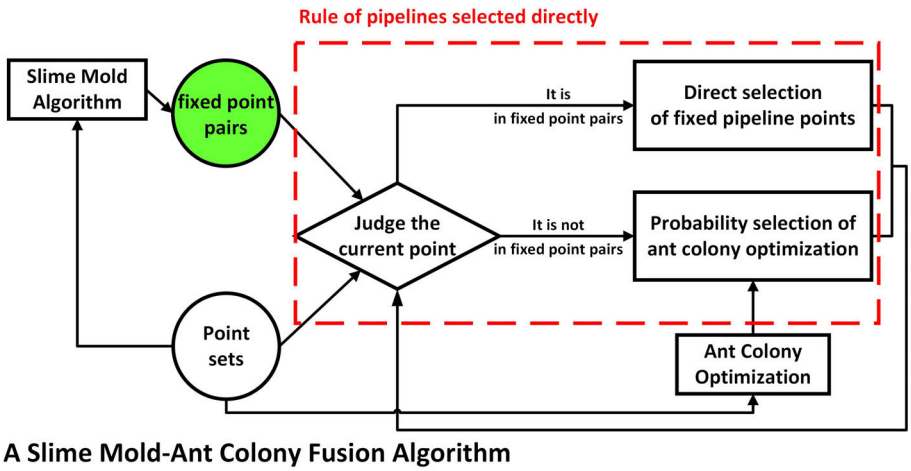


FIGURE 4. Schematic diagram of algorithm principle by SMACFA.

where the D_{ij} is the conductivity of pipeline (i, j) and L_{ij} is the distance of pipeline (i, j). The pressure value can be calculated by setting a point of pressure as the reference, such as setting $j = 2$ and $P_2 = 0$ as reference.

3) Q_{ij} is the flow of pipeline (i, j), the flow value between point i and j is calculated by the relationship among the pressure difference $(P_i - P_j)$, the conductivity D_{ij} and the length L_{ij}. The calculation formula of Q_{ij} is as follows:

$$Q_{ij} = \frac{D_{ij}}{L_{ij}} \left(P_i - P_j\right) \tag{8}$$

The formula of updating the conductivity is expressed as:

$$\frac{dD_{ij}}{dt} = \frac{Q_{ij}}{1 + Q_{ij}} - D_{ij} \tag{9}$$

The iteration formula is obtained as:

$$D_{ij}(n+1) = \left(\frac{\left|Q_{ij}(n)\right|}{1 + \left|Q_{ij}(n)\right|} - D_{ij}(n)\right) * \Delta t + D_{ij}(n) \tag{10}$$

4) Using iterative method, the formula (7) is needed to repeat, (8) and (10) until the iteration termination condition is met. The iteration termination condition is as follows:

$$\left|D_{ij}(n+1) - D_{ij}(n)\right| \leq \delta \tag{11}$$

According to the final pipeline flows, the optimal path L_{best} is obtained by selecting the pipeline with the largest flow. The formula for selecting the optimal path is defined as:

$$Q_{inest} = \max\left\{\left|Q_{i1}\right|, \left|Q_{i2}\right|, \cdots, \left|Q_{in}\right|\right\} \tag{12}$$

where Q_{inest} represents the largest pipeline flow value from the current point i to the next point (next). After a selection circle, point j selected will become the current point i in the next selection process. According to formula (12), the path optimization enters the next selection and the point i is put in L_{best} in sequence.

III. SLIME MOLD-ANT COLONY FUSION ALGORITHM

In the process of optimization, the pheromones of ACO are evenly distributed at the beginning. So, when the number of path points is large and dense, the convergence speed is slow and the algorithm is easy to fall into local optimum. Additionally, the optimization mechanism of ACO is selecting the next point by random. It appears that some path segments with long distance are selected. Hence, the suboptimal solutions are usually got rather than optimal results, and a lot of optimization time is spent. According to these deficiencies, SMA and ACO are combined to design the SMACFA by fixing selection of high-quality pipelines in this paper. As illustrated in Fig. 4, the designed key is to extract some high-quality path segments from SMA and directly apply them to ACO as fixed paths. On the one hand, SMACFA prevents the algorithm from falling into the local optimum and improves the probability of the optimal path; On the other hand, it reduces the number of path points and the complexity for improving the search efficiency.

The selection rule of high-quality pipelines is as shown in Fig. 5. The shortest distance and the result of SMA are combined, that is, the path segments with shortest distance

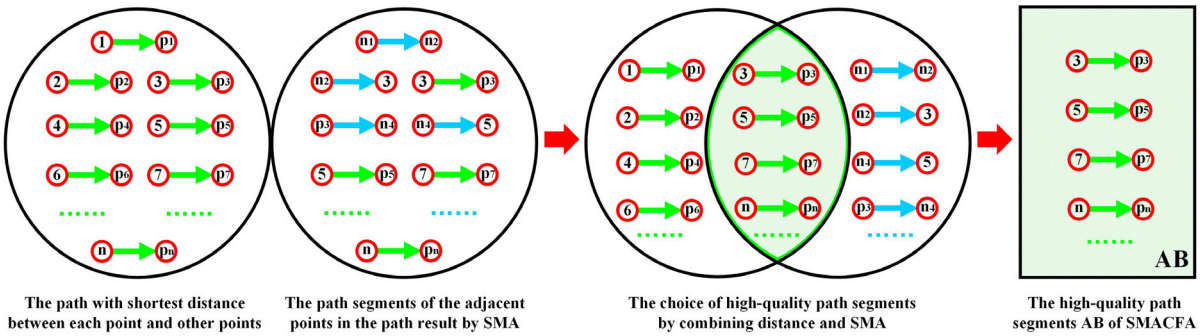


FIGURE 5. The selection mechanism of high-quality pipelines.

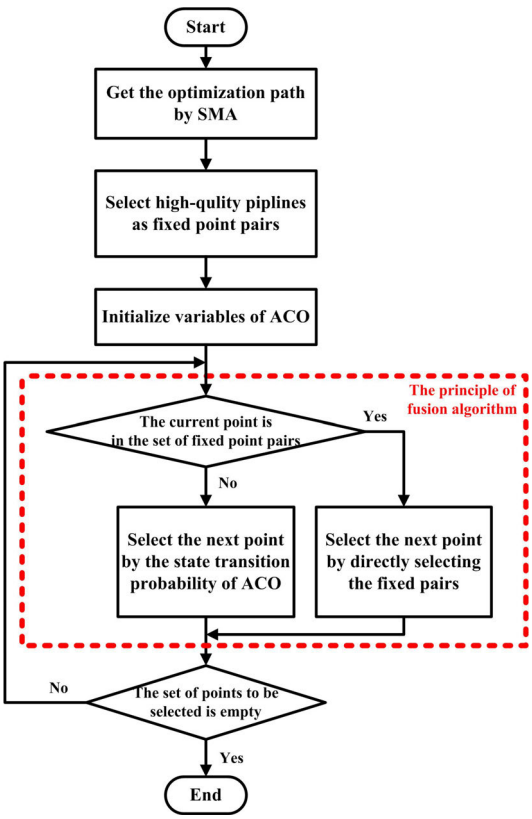


FIGURE 6. Flow chart of SMACFA.

in SMA are selected. Then these path segments are directly applied to ACO to solve the local optimal problem. The times of the random selection of ACO are reduced and the influences of the flow factor and distance factor are added in the process of optimization. So, the optimization ability and convergence speed of the algorithm are obviously improved.

The basic principle of SMACFA is shown in Fig. 4. Firstly, a path is obtained by using the "pipeline cultivation" method of SMA. Then, the relationship among distance, flow and conductivity is combined to screen out the high-quality pipelines. At the same time, the two ends of pipeline are taken as the fixed-point pair to form the set L_{AB}. The fixed-point pairs in L_{AB} are applied to complete the directional selection in the optimization process of ACO, that is, the next point is directly determined when a point is selected in L_{AB}.

In other words, the rule of fusion algorithm is equivalent to automatically match points by mapping relationship between set A and set B, where, set A and set B are the sets combined with the end points of the fixing point pairs.

SMACFA combines the fixed-point pairs from SMA and the state movement probability of ACO. According to the principle of SMACFA, the algorithm flowchart is as shown in Fig. 6 and the steps for solving TSP are as follows:

1) The path L_{best} needs to be obtained by SMA. According to the relationship among the distance, flow and conductivity, the high-quality pipelines with large flow and short distance between two points are selected from L_{best}. The two ends of the high-quality pipelines are taken as point pairs and put into the set L_{AB}. The selection formula of the fixed-point pairs is as follows:

$$L_{AB} = \left\{(i,j) \left| Q_{ij} > Q_\eta, D_{ij} > D_\eta \ , \ i \in L_{best}\,(u), \right. \right. \\ \left. j \in L_{best}\,(u+1)\,, u \in [1,\ n) \right\} \quad (13)$$

where Q_η and D_η are respectively the limited values of flow and distance, i is the Uth value in L_{best} and j is the $(U+1)$th value in L_{best}. L_{AB} is the set which is consist of both ends of high-quality pipelines.

2) The intersection set AB is calculated by set A and set B. The intersection points in AB are removed and directly put into the tabu table as the points which have been visited. It will prevent the intersection points from being selected by state movement probability method and affecting the direct selection rule of fixed pipelines.

3) The current point i will be judged whether it is in L_{AB} or not. If i is in L_{AB}, the next point will be selected by the fixed selection rule of high-quality pipelines. Otherwise, the next point will be visited by state movement probability of ACO. In summary, the selection formula of next point is obtained as:

$$J = \begin{cases} J_{L_{AB}}, & j = allowed\ and\ i \in L_{AB} \\ \max\left\{p_{ij}^k\right\}, & j = allowed\ and\ i \notin L_{AB} \end{cases} \quad (14)$$

4) According to the rule of state transition probability in ACO, the selection from the current point to the next point is carried out. The probability calculation formula

TABLE 1. Steps of slime mold-ant colony fusion Algorithm.

SMACFA to solve the TSP
Import: path point sets
Output: shortest path
(a) Initialization
Step 2 Use SMA to get the flow Q_{ij} and conductivity D_{ij}
Step 3 Get the optimal path L_{best} of SMA by flows
Step 4 Select the high-quality pipelines and get the set L_{AB} of fixed-point pairs
(c) Select the next point by fixed pipelines rule
Step 5 Calculate state transition probability by ACO
Step 6 Use the rule of fixed selection to determine next point when the current point is in L_{AB} Otherwise, use state transition probability of ACO to select the next point
Step 7 Modify tabu table and update pheromones of ant colony algorithm
Step 8 Get back to step 5, iterate until the iteration conditions are met
(d) Obtain the result
Step 9 Get and output the optimal path $L_{Length}best$

is defined as:

$$P_{ij}^{k} = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{j\in allowed}\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}, & j = allowed \\ 0, & otherwise \end{cases} \tag{15}$$

And the pheromone updating formula is:

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{16}$$

The corresponding point of the current point is found by the fixed-point pairs in L_{AB}. According to the fixed selection rule of high-quality pipelines, the corresponding point is directly taken as the next selected point. The fixed selection formula is expressed as:

$$J_{L_{AB}} = \begin{cases} L_{AB}(u,2), & point\ i\ is\ L_{AB}\ (u,1) \\ L_{AB}(u,1), & point\ i\ is\ L_{AB}\ (u,2) \end{cases} \tag{17}$$

5) After the selection from the current point to the next point, the point selected will be the new current point to complete the next selection cycle by formula (14). And then, the Ncth path result $L_{Length}(Nc)$ is obtained when all points are continuously visited by formula (13) - (17). Furthermore, when iteration termination condition is met by the iteration, all the paths $L_{Length}(Nc)$ are obtained and the formula is given as:

$$L_{Length} = \left\{L_{Length}(1), \cdots, L_{Length}(Nc_\max)\right\} \tag{18}$$

where the Nc is the number of the current cycle and the $Nc_\max$ is the maximum of iterations.

Finally, the optimal path $L_{Length}best$ with the shortest path is selected from all the paths obtained and the formula is as follows:

$$L_{Length}best = \min\left\{L_{Length}\right\} \tag{19}$$

According to the above description, the steps of SMACFA are summarized as shown in Table 1.

TABLE 2. Parameters and configurations of the algorithms.

Algorithm parameters	
GA	Population size $M = 50$ Crossover probability $\alpha = 0.8$ Mutation probability $\beta = 0.5$ Iterations $Nc = 200$
PSO	Population size $M = 50$ Inertia weight decreases linearly from 0.9 to 0.4 Individual-best acceleration factor $C1 = 2$ Global-best acceleration factor $C2 = 2$ Iterations $Nc = 200$
SMA	Pressure value of starting point is generated randomly from 0.1 to 50 Index of conductivity deviation $\delta = 10^{-6}$ Iterations $Nc = 200$
ACO	Population size $M = 50$ Pheromone trail factor $\alpha = 1$ Heuristic information factor $\beta = 5$ Evaporation rate $\rho = 0.5$ Iterations $Nc = 200$
SMACFA	Pressure value of starting point is generated randomly from 0.1 to 50 Index of conductivity deviation $\delta = 10^{-6}$ Population size $M = 50$ Pheromone trail factor $\alpha = 1$ Heuristic information factor $\beta = 5$ Evaporation rate $\rho = 0.5$ Iterations $Nc = 200$

TABLE 3. The high-quality fixed pipelines.

No.	Fixed Pipeline AB	
1	2	4
2	8	9
3	9	10
4	12	13
5	14	12
6	15	14
7	18	3
8	19	17
9	20	24
10	21	22
11	26	28
12	28	27
13	30	31

IV. SIMULATION AND RESULT ANALYSIS

Through the Traveling Salesman Problem Library (TSPLIB) which is included the large data sets for TSP, the simulation tests of GA, PSO, LKH, SMA, ACO and SMACFA etc. were carried out on the server in turn by matlab. The parameters and configuration information of the relevant algorithms are shown in Table 2. Through the simulations of data points and comparisons of results, the experimental results of the path length, the convergence and the algorithm time complexity are analyzed to verify the effectiveness and superiority of the SMACFA designed.

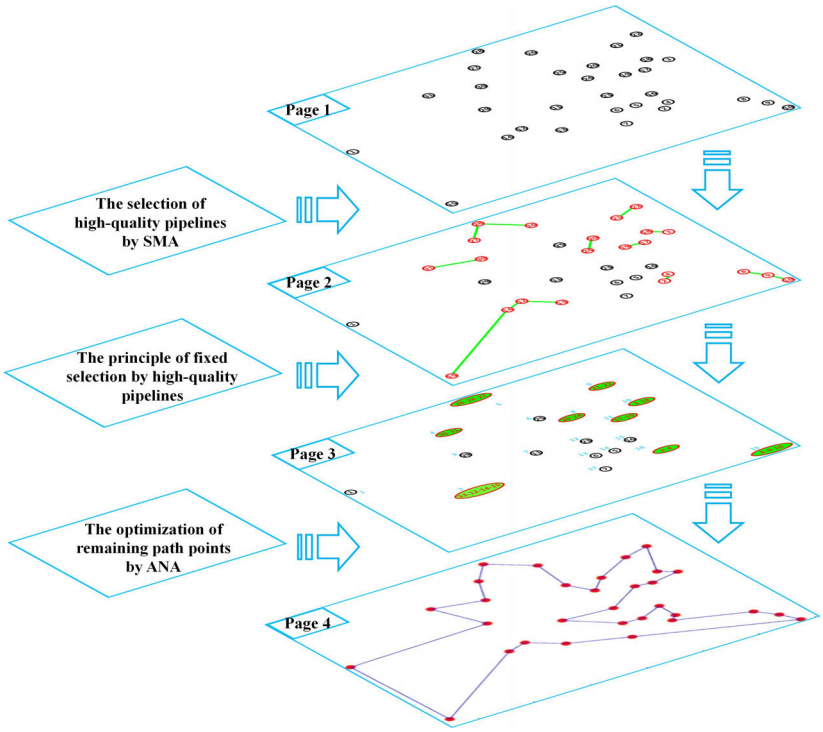


FIGURE 7. The schematic diagram of SMACFA on chn31 data.

TABLE 4. The test results of path length and convergence on chn31 data.

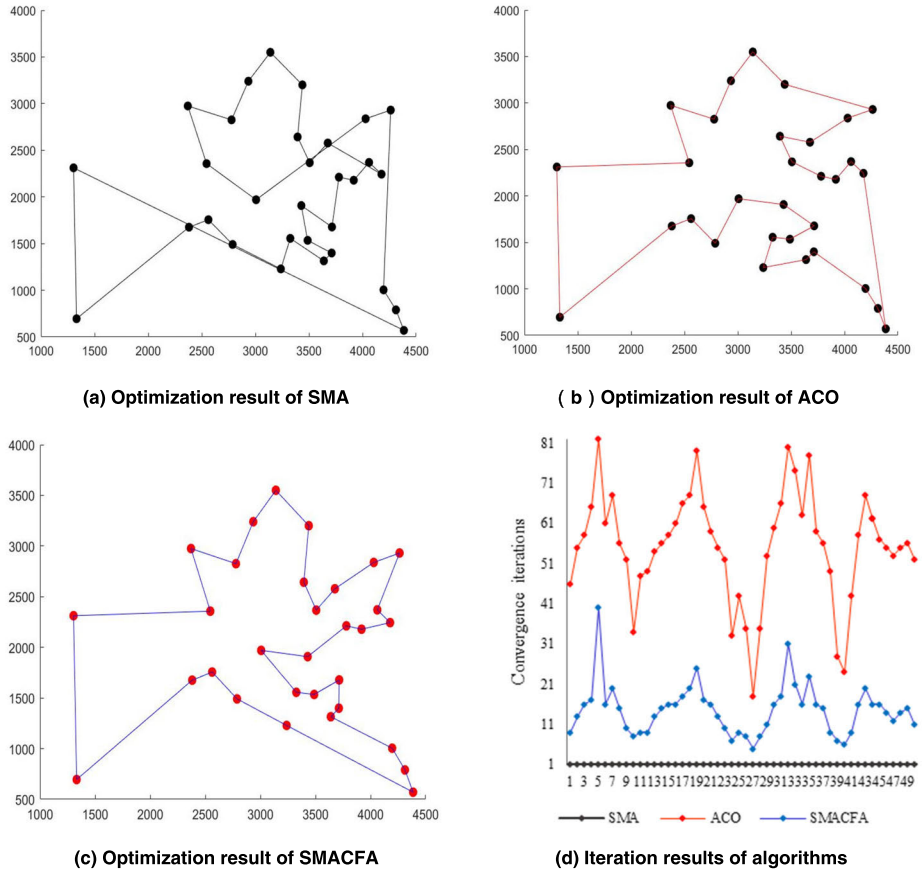
Name	SMA	ACO	**SMACFA**	**Increased percentage** (between SMACFA and ACO)
Optimal result	19282	15602	**15381**	**1.42%**
Average result	19282	15866.4	15492.8	**2.35%**
Standard deviation	-	191.47	99.68	**47.94%**
Wilcoxon rank sum test	6	39	15/16	p>=0.05
Average λ -branching factor	1	8.6	**3.2**	**62.79%**
Average convergence iteration	1	55.2	14.6	**73.55%**
Standard deviation	-	13.82	6.32	**54.27%**
Wilcoxon rank sum test	1	610	230/210	**p<=0.05**
Number of points	31	31	18	**41.94%**
Number of points	n	n	n-nAB (nAB~n/3)	**33.33%**
Algorithm time complexity	n^3	n^4	n^4 *16/81	**80.25%**

A. RESULT OF OPTIMIZATION

Chn31 are selected as data points from TSPLIB data set, which contains the latitude and longitude coordinates of 31 cities in China. In the data chn31, the high-quality fixed pipelines are selected by SMA and shown in Table 3. Therefore, the path point is reduced from 31 to 18 when ACO is used to optimize. The schematic diagram of SMACFA optimization on chn31 data is shown in Fig. 7. Based on SMACFA designed in this paper, the comparative results are obtained in Table 4. It can be seen that in Table 4, the shortest path length is 15381 which is significantly longer than 15602 obtained by ACO, and the convergence speed and algorithm time complexity are respectively improved 62.79 and 80.25%. Fig. 8 is the simulation diagrams of optimization result and convergence.

B. ANALYSIS OF PATH LENGTH

The sub-optimal solutions are always obtained rather than optimal solutions by ACO. The reason for this problem is that partial selections of path segment are not optimal, so the fixed selections will effectively solve the local optimal solution of ACO. As shown in Table 5 and Fig. 9, in the suboptimal ACO result of 15889, point 12 is selected as the next point after point 15. In the final result 15602 of the ACO, the next point of optimum selection after point 15 is point 14, and the distance from point 15 to other points is longer



(a) Optimization result of SMA

(b) Optimization result of ACO

(c) Optimization result of SMACFA

(d) Iteration results of algorithms

FIGURE 8. Simulation results of SMACFA on chn31 data.

TABLE 5. Optimization results of path point on chn31 data.

Name	Planning result of path point	Length
ACO	1 15 14 12 13 11 23 16 5 6 7 2 4 8 9 10 3 18 17 19 24 25 20 21 22 26 28 27 30 31 29	15602
ACO (Suboptimal)	1 15 12 14 13 11 23 16 5 6 7 2 4 8 9 10 3 18 17 19 24 25 20 21 22 26 28 27 30 31 29	15889
SMA	1 15 14 12 13 7 6 2 4 5 23 16 19 17 18 3 20 24 25 26 28 27 30 31 29 11 21 22 8 9 10	19282
SMACFA	1 **15** **14** **12** 13 7 10 9 8 2 4 16 5 6 11 23 19 17 3 18 22 21 20 24 25 26 28 27 30 31 29	**15381**

than that from point 14. The segment between point 14 and point 15 is a fixed path pipeline, and it is the path selected by SMA with the shortest distance than other points. There are many high-quality paths in the result of SMA, however, they are always not chosen by ACO due to its random rule. Therefore, under the effect of the fixed high-quality pipeline pairs (in Table 3), SMACFA directly applies the fixed pipelines to ACO, the obtained result by SMACFA is 15381. At the same time, the results and average path length are gained as Fig. 10, the average path results of SMA, ACO and SMACFA are respectively 19282, 15866.4 and 15492.8.

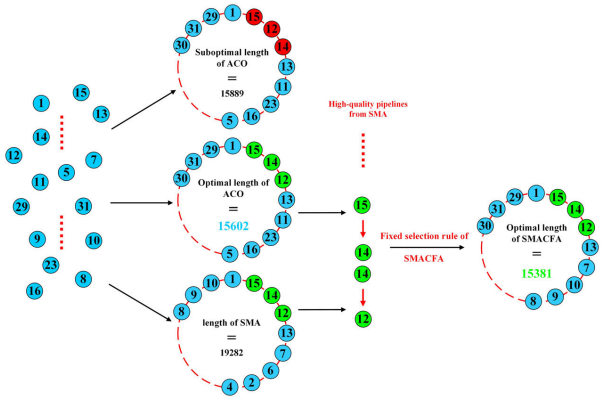


FIGURE 9. Optimization results and analysis of path point on chn31 data.

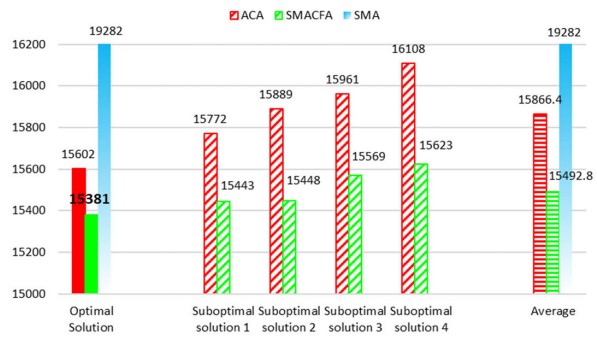


FIGURE 10. Path solutions and average results obtained by each algorithm.

It can be known that in Table 4, compared with the total path lengths of ACO, the result of SMACFA is reduced by 221. In addition, ACO itself is good and the rank sum results of ACO and SMACFA meet the similar performance requirements, so the good character of ACO is not destroyed by improving. What's more, the path optimal value and average value of SMACFA are increased by 1.42 and 2.35%. Therefore, it is evidenced that optimization result of SMACFA is better than SMA and ACO.

In addition, robustness is critical to heuristic algorithms. Through the principle of fixed selection of pipelines, SMACFA greatly reduces the number of points to optimized by the state transition probability. So, the randomness of algorithm is reduced and the probability of poor points being selected is reduced. Moreover, the algorithm's robustness will be significantly improved. As shown in Table 4, with the standard deviation of average λ-branching factor reduced by 54.27%, the randomness is greatly reduced. Meanwhile, the standard deviation of the path length is reduced by 47.94%. Therefore, the fluctuation of the algorithm is greatly reduced and the robustness is increased.

C. ANALYSIS OF CONVERGENCE AND TIME COMPLEXITY

With chn31 data, each algorithm was tested for 50 times to obtain the distribution of convergence iteration in different ranges and convergence results in Fig. 11 and Table 4. In ACO, due to the selection of each point all has randomness, the convergence speed is slower. SMACFA will directly select

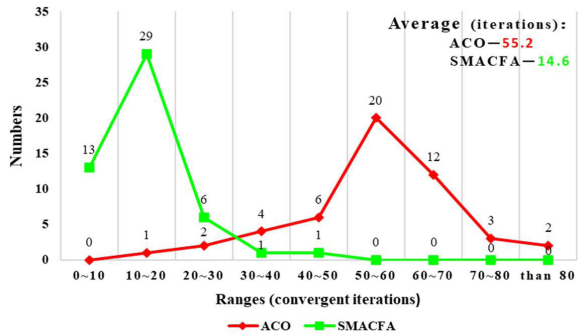


FIGURE 11. The convergence distribution of being tested 50 times.

some path points (such as Table 3) by the fixed pipeline selection principle, and the number of fixed-point pairs is about one third of total path points. Because the actual path points to be selected are reduced, the number of random is reduced, the iterations of convergence are reduced and the convergence speed of algorithm is improved.

As shown in Fig. 11 and Table 4, the average iterations were 14.6 when it achieved convergence by SMACFA, while ACO needed 55.2. At the same time, the convergence iterations of SMACFA are mostly concentrated in the range of 10-20, while ACO is in the range of 50-60. Moreover, iterations of getting to converge is analyzed by the Wilcoxon Rank Sum Test of statistics. As shown in Table 4, the statistical results of the rank sum show that ACO and SMACFA have large differences in the number of convergence iterations, and the fewer iterations of SMACFA are needed. So, the convergence performance of SMACFA is far better than ACO.

Besides, the situation of average λ-branching factor, which is the important convergence measure indicator of heuristic algorithm, is shown in Table 4. The less λ-branching factor is, the less the point need to be tested. The average λ-branching factor of SMACFA is 3.2, and far less than 8.6 of ACO. That is to say, the optimization speed of SMACFA is faster than that of ACO, and the convergence property is stronger. Therefore, the convergence speed is decreased significantly by SMACFA.

The algorithm time frequency T(n) (the number of statements executed by the algorithm) is the same as the algorithm time complexity O(f(n)), that is, T(n)=O(f(n)). Therefore, the algorithm time complexity can also measure the algorithm executing speed. As shown in Table 6 and Fig. 12, the algorithm time complexity of SMA is n^3, and ACO's is n^4. The actual path point of the fusion algorithm is n-nAB and the nAB value is usually about one third of n. Therefore, the complexity of SMACFA is 16/81 times of ACO (that is, $n^4 * 16/81$). Therefore, the algorithm speed of SMACFA will be improved about 80.25%.

D. COMPARISON AND ANALYSIS WITH OTHER ALGORITHMS

In this paper, the other improved ACO algorithms, which are included on the elite ant system (ACO-1), the max-min

TABLE 6. The results of algorithm time complexity.

Name	Setting	Complexity Formula	General Rule	Algorithm Time Complexity
SMA	n : the number of path points	O (f(n))=time(SMA) $= n^3$	/	Time(SMA) = n^3
ACO	n : the number of path points m : the number of ant T : iterations	O (f(n))=time(ACO) =n*(n-1)*m*T/2	m=n; T=k*n; n>>k>=1	Time(ACO) = n^4
SMACFA	n : the number of path points nAB : the number of fixed pipelines AB	O (f(n))=time(SMACFA) $= n^3 + (n - nAB)^4$	nAB =n/3	Time(SMACFA) = $\boldsymbol{n^4 * 16/81}$
Percent Increase	**Up**=(Time(ACO)-Time(SMACFA)) / Time(ACO)=**80.25%**			

TABLE 7. The comparison between SMACFA and ACO (pheromone improvement) under TSPLIB.

Name	ACO and improved ACOs					SMACFA				Increased Percentage		
	$\bar{\lambda}$	Complexity	ACO-1 Length	ACO-2 Length	ACO-3 Length	n_{AB}	$\bar{\lambda}$	Complexity	Length	$\bar{\lambda}$	Complexity	(mean) Length
Berlin52	6	52^4	7682	7791	7748	21	5	52^3+31^4	7572	16.67%	85.45%	2.17%
Pr76	13	76^4	120508	122325	121051	34	8	76^3+42^4	114668	38.46%	89.36%	5.46%
Lin105	17	105^4	15275	15310	15302	42	12	105^3+63^4	15055	29.41%	86.09%	1.58%
Gr120	19	120^4	1824	1843	1833	50	13	120^3+70^4	1805	31.58%	87.59%	1.54%
Ch130	21	130^4	6689	6716	6733	50	14	130^3+80^4	6403	33.33%	84.89%	4.61%
Ch150	24	150^4	6918	7089	7089	60	17	150^3+90^4	6776	29.17%	86.37%	3.63%
Gr202	33	202^4	539	551	546	88	22	202^3+114^4	532	33.33%	89.36%	2.44%
Tsp225	34	225^4	4346	4440	4365	94	23	225^3+131^4	4239	32.35%	80.06%	3.29%
A280	48	280^4	3002	3128	3071	160	32	280^3+120^4	2884	33.33%	96.26%	5.94%
Pcb442	75	442^4	62764	63035	61725	244	53	442^3+198^4	60677	29.33%	95.75%	2.92%

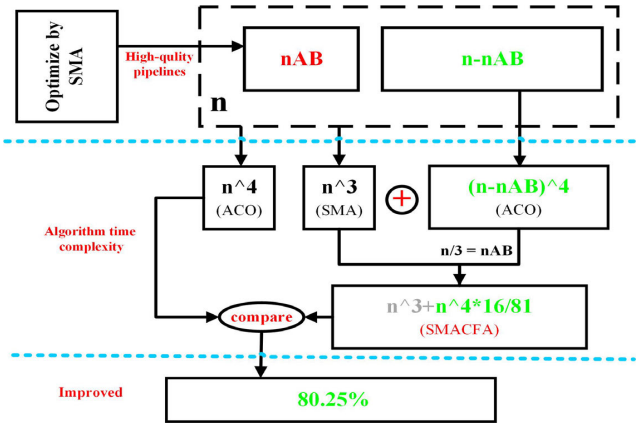


FIGURE 12. The schematic diagram of algorithm time complexity.

ant system (ACO-2) and the initial pheromone optimization ant colony system (ACO-3), are tested and compared with SMACFA to further verify the performance of SMACFA. Under the data of berlin52, lin105, ch150, tsp225 and etc., the test results are shown in Table 7.

As shown in Table 7 and Fig. 13, the path lengths of SMACFA are improved separately 2.17, 5.46, 1.58, 1.54, 4.61, 3.63, 2.44, 3.29, 5.49 and 2.92% than the other improved ACO algorithms by using ten sets of TSPLIB data to test. At the same time, the average λ- branching factor of SMACFA becomes small, so the convergence speed is accelerated. In addition, due to the low complexity and the high-quality path of SMA, SMACFA saves a lot of algorithm time complexity for the optimization part of the ACO rule, and it is greatly reduced the algorithm time complexity of SMACFA. Furthermore, as shown in Fig. 13, the more points are tested, the more performances of SMACFA on path length, convergence speed and algorithm time complexity are improved. It is adequately proved that the performance of SMACFA is better than the other improved ACO algorithms.

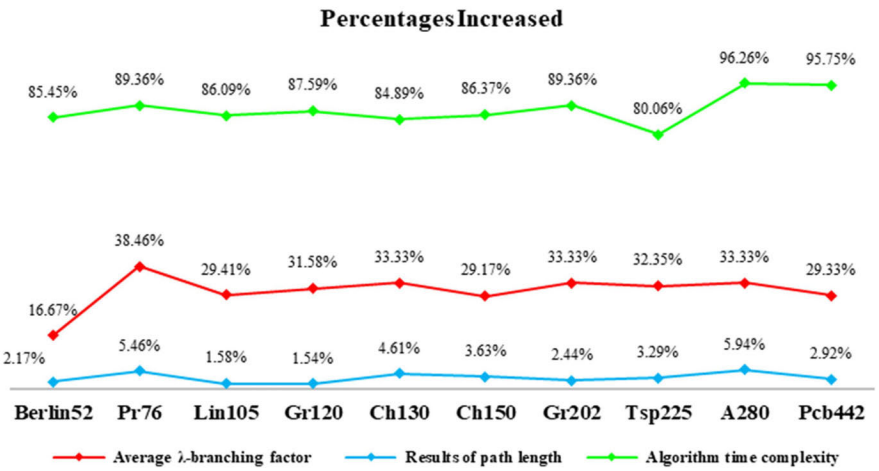


FIGURE 13. The comparison of SMACFA and the improved ACO algorithms.

TABLE 8. The comparison of algorithm performance under TSPLIB date sets.

Type	Basic Algorithm			Fusion Algorithm						This Article	
Name	GA	PSO	LKH	GA-ACO		PSO-ACO		LKH -ACO		SMACFA	
				$n_{AB}/\bar{\lambda}$	Result	$n_{AB}/\bar{\lambda}$	Result	$n_{AB}/\bar{\lambda}$	Result	$n_{AB}/\bar{\lambda}$	Result
Berlin52	9949	8606	7542	16/6	7806	16/6	7878	21/5	7694	21/5	7572
Pr76	157604	152946	108159	5/13	123465	19/11	117480	29/9	115270	34/8	114668
Lin105	30168	23752	14379	9/16	15378	18/14	15188	49/12	15206	42/12	15055
Gr120	6917	4245	1667	4/19	1824	10/18	1849	43/14	1821	50/13	1805
Ch130	33270	20309	6110	1/21	6644	5/20	6690	48/15	6680	50/14	6403
Ch150	14245	17290	6528	5/24	7010	8/25	7088	59/18	6958	60/17	6776
Gr202	2418	1624	549	4/32	570	8/31	551	71/25	543	88/22	532
Tsp225	11350	15022	3916	3/34	4218	7/34	4320	90/27	4362	94/23	4239
A280	26108	14161	2579	6/45	3092	14/44	3111	158/32	3019	160/32	2884
Pcb442	689033	407987	50778	3/71	62669	11/67	62948	274/53	61266	244/53	60677
Time complexity	n^3	n^3	n^5	$k_1 * n^4 + n^3$		$k_2 * n^4 + n^3$		$k_3 * n^5 + n^3$		$\boldsymbol{k * n^4 + n^3}$	
	(In general , $k \approx k_3 \approx k_1/3 \approx k_2/3 < 1$)										

Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are the basic bionic algorithms with good optimization ability like ACO. Lin-Kernighan-Helsgaun Solver (LKH) is the heuristic algorithm with the best optimization results in solving TSP so far. In order to further verify the rationality and the superiority of SMACFA in this paper, GA, PSO and LKH are respectively contrasted to SMACFA under the ten groups of data sets. Besides this, the good paths from the GA, PSO and LKH are combined with ACO respectively through the rule of the high-quality pipeline selection to optimize, and their results are compared with SMACFA to verify good performance of SMACFA. The specific comparison results of algorithms are shown in Table 8.

It can be seen that from the Table 8, path optimization results of SMACFA are significantly better than GA and PSO, and close to the current optimal results which are obtained by LKH. Moreover, for SMACFA, since the complexity of SMA algorithm is n^3 and the selected high-quality pipelines is about $n/3$, the complexity of SMACFA algorithm is about$n^4 *$ 16/81. The algorithm time complexity of LKH is n^5, so the optimization time and convergence speed are slow. And the larger the data points are, the more time is spent due to high exponential relation. Thus, while SMACFA path result is slightly weaker than the current optimal algorithm LKH, the algorithm time complexity is much better than LKH.

Furthermore, the good paths of GA, PSO and LKH were extracted and applied to ACO respectively. According to the

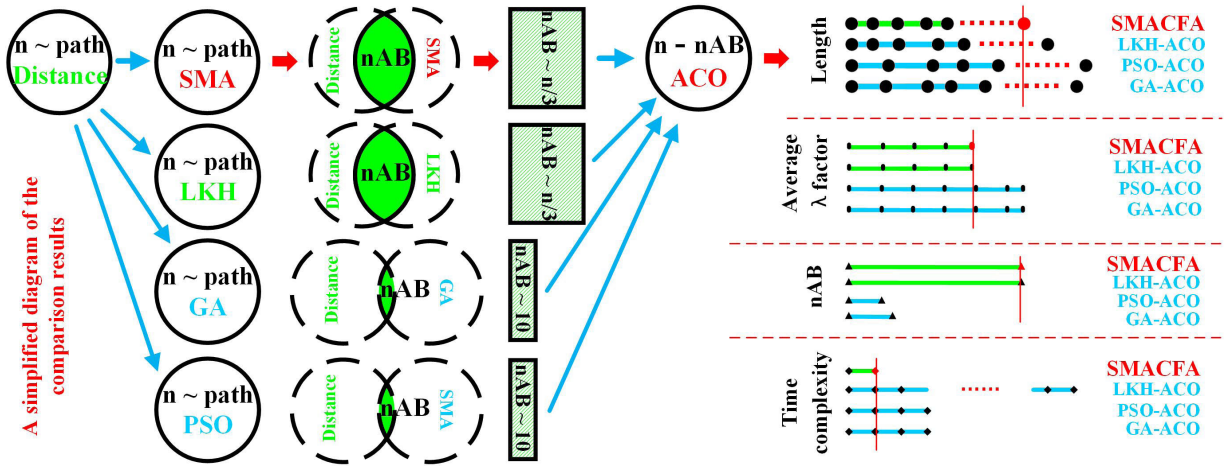


FIGURE 14. The schematic diagram of comparison results.

direct selection rule of SMACFA to test and compare in these algorithms, the test results were shown in Table 8 and Fig. 14. GA, PSO and LKH all used the high-quality pipelines rule to optimize, but the results are not as good as SMACFA due to long path lengths to be get. What's more, the numbers of good pipelines obtained by GA and PSO are very small and the average λ- branching factors of them are high; therefore, the algorithm time complexity and convergence are weaker than SMACFA. For LKH, although the numbers of selected pipelines of LKH is similar to SMACFA, the optimization results are not ideal. And the algorithm time complexity of LKH itself is large, it takes a lot of time to select high-quality pipelines. So LKH still cannot get rid of the disadvantage of large algorithm time complexity and not better than SMACFA.

Based on the results of the comparative experiment analysis of the above algorithms, it can be concluded that the SMACFA designed in this paper has certain advantages than other algorithms in terms of path optimization results, algorithm time complexity and algorithm convergence speed and fully proved the superiority of SMACFA.

According to the above analysis and results, it is easy to reach the conclusion that the path length and the average path length are improved by SMACFA than other algorithms. At the same time, the path points are reduced about a third by the fixed selection principle of high-quality pipelines. It makes the number of convergence iterations and the algorithm time complexity obviously reduced. Therefore, the performance superiority of SMACFA has been fully proved.

V. CONCLUSION

In this paper, SMACFA for the fixed selection rules of the high-quality pipelines is proposed to prevent algorithm from falling into the local optimum and speed up the convergence speed in ACO. In summary, the high-quality paths of SMA were directly applied to ACO to achieve the fixed selection rules of SMACFA. The test results and the contrast experiment under the TSPLIB data, show that the three indexes of the optimization ability, the convergence speed and the algorithm time complexity are all improved. It is verified the effectiveness and superiority of SMACFA in solving the TSP.

The selection of parameter values has impact on the performance of SMACFA. At present, some new heuristic algorithms for function optimization appear constantly, such as, Henry Gas Solubility Optimizer, Equilibrium optimizer, Marine Predator Algorithm etc. These algorithms search aims randomly and update their position by the mechanism about particle concentrations and can be applicable to optimization problems of mathematical functions. So, they can be taken into consideration to optimize SMACFA parameters. The subsequent research will focus on selecting appropriate methods to optimize the parameter values and improve the algorithm performance.

REFERENCES

[1] H. Shi, L. Zheng, and G. Liu, "Research on TSP based on ant colony algorithm," in *Proc. IEEE Int. Conf. Inf. Autom.*, Aug. 2015, pp. 2048–2051.

[2] A. Ouaarab, B. Ahiod, and X.-S. Yang, "Random-key cuckoo search for the travelling salesman problem," *Soft Comput.*, vol. 19, no. 4, pp. 1099–1106, Apr. 2015.

[3] A. Khanra, T. Pal, M. K. Maiti, and M. Maiti, "Multi-objective four dimensional imprecise TSP solved with a hybrid multi-objective ant colony optimization-genetic algorithm with diversity," *J. Intell. Fuzzy Syst.*, vol. 36, no. 2, pp. 1–19, 2019.

[4] F. Chen, P. Deng, T. Ding, and W. Liang, "Research on ant colony optimization Tabu search and genetic fusion algorithm," in *Proc. 2nd Int. Conf. Robot. Autom. Sci. (ICRAS)*, Jun. 2018, pp. 1–5.

[5] R. Bernardino and A. Paias, "Heuristic approaches for the family traveling salesman problem," *Int. Trans. Oper. Res.*, vol. 28, no. 1, pp. 262–295, Jan. 2021.

[6] I. Khan, S. Pal, and M. K. Maiti, "A hybrid PSO-GA algorithm for traveling salesman problems in different environments," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 27, no. 05, pp. 693–717, Oct. 2019.

[7] M. Yu, "A solution of TSP based on the ant colony algorithm improved by particle swarm optimization," *Discrete Continuous Dyn. Syst.*, vol. 12, nos. 4–5, pp. 979–987, 2019.

[8] D. Applegate, W. Cook, and A. Rohe, "Chained Lin-Kernighan for large traveling salesman problems," *INFORMS J. Comput.*, vol. 15, no. 1, pp. 82–92, Feb. 2003.

[9] K. Helsgaun, "Solving arc routing problems using the Lin-Kernighan-Helsgaun algorithm," *Ice Trans. Inf. Syst.*, vol. e95, no. 4, pp. 1031–1043, 2015.

[10] A. Bevilaqua, D. Bevilaqua, and K. Yamanaka, "Parallel island based memetic algorithm with Lin–Kernighan local search for a real-life two-echelon heterogeneous vehicle routing problem based on Brazilian wholesale companies," *Appl. Soft Comput.*, vol. 76, pp. 697–711, Mar. 2019.
[11] M. Mavrovouniotis and S. Yang, "Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors," *Appl. Soft Comput.*, vol. 13, no. 10, pp. 4023–4037, Oct. 2013.
[12] W. Zhang, X. Gong, G. Han, and Y. Zhao, "An improved ant colony algorithm for path planning in one scenic area with many spots," *IEEE Access*, vol. 5, pp. 13260–13269, 2017.
[13] R. Moeini and M. H. Afshar, "Hybridizing ant colony optimization algorithm with nonlinear programming method for effective optimal design of sewer networks," *Water Environ. Res.*, vol. 91, no. 4, pp. 300–321, Apr. 2019.
[14] A. Siemiński and M. Kopel, "Solving dynamic TSP by parallel and adaptive ant colony communities," *J. Intell. Fuzzy Syst.*, vol. 37, no. 1, pp. 1–12, 2019.
[15] V. Evangelidis, M.-A. Tsompanas, G. C. Sirakoulis, and A. Adamatzky, "Slime mould imitates development of roman roads in the Balkans," *J. Archaeol. Sci., Rep.*, vol. 2, pp. 264–281, Jun. 2015.
[16] M. Becker, "On the quality of graphs generated by swarm algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 3734–3739.
[17] S. Meyer-Nieberg, "Coordinating a team of searchers: Of ants, swarms, and slime molds," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2017, pp. 27–34.
[18] X. Zhang and S. Mahadevan, "A bio-inspired approach to traffic network equilibrium assignment problem," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1304–1315, Apr. 2018.
[19] T. Takaoka, M. Sato, T. Otake, and T. Asaka, "Novel routing method using slime mold algorithm corresponding to movement of content source in content-oriented networks," *J. Signal Process.*, vol. 23, no. 4, pp. 173–176, 2019.
[20] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.
[21] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proc. Eur. Conf. Artif. Life (ECAL)*, 1991, pp. 134–142.
[22] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
[23] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generat. Comput. Syst.*, vol. 16, no. 8, pp. 889–914, Jun. 2000.
[24] O. C. García, I. J. F. de Viana González, and F. H. Triguero, "Analysis of the best-worst ant system and its variants on the TSP," *Mathware Soft Comput.*, vol. 9, no. 3, pp. 177–192, 2002.
[25] Q. F. Qiu and X. J. Xie, "Theoretical analysis on initial pheromone values for ACO," *Fuzzy Eng. Oper. Res.*, vol. 147, pp. 339–349, 2012.
[26] T. Nakagaki, H. Yamada, and Á. Tóth, "Intelligence: Maze-solving by an amoeboid organism," *Nature*, vol. 407, no. 6803, p. 470, 2000.
[27] A. Tero, R. Kobayashi, and T. Nakagaki, "A mathematical model for adaptive transport network in path finding by true slime mold," *J. Theor. Biol.*, vol. 244, no. 4, pp. 553–564, Feb. 2007.
[28] T. Nakagaki, M. Iima, T. Ueda, Y. Nishiura, T. Saigusa, A. Tero, R. Kobayashi, and K. Showalter, "Minimum-risk path finding by an adaptive Amoebal network," *Phys. Rev. Lett.*, vol. 99, no. 6, Aug. 2007, Art. no. 068104.
[29] A. Tero, K. Yumiki, R. Kobayashi, T. Saigusa, and T. Nakagaki, "Flow-network adaptation in physarum amoebae," *Theory Biosci.*, vol. 127, no. 2, pp. 89–94, May 2008.
[30] M. Aono, M. Hara, K. Aihara, and T. Munakata, "Amoeba-based emergent computing: Combinatorial optimization and autonomous eta-problem solving," *Int. J. Unconventional Comput.*, vol. 6, no. 2, pp. 89–108, 2010.
[31] V. Bonifaci, "Physarum can compute shortest paths: A short proof," *Inf. Process. Lett.*, vol. 113, nos. 1–2, pp. 4–7, Jan. 2013.
[32] Y. Liu, Y. Lu, C. Gao, Z. Zhang, and L. Tao, "A multi-objective ant colony optimization algorithm based on the physarum-inspired mathematical model," in *Proc. 10th Int. Conf. Natural Comput. (ICNC)*, Aug. 2014, pp. 303–308.
[33] Z. Zhang, C. Gao, Y. Liu, and T. Qian, "A universal optimization strategy for ant colony optimization algorithms based on the physarum-inspired mathematical model," *Bioinspiration Biomimetics*, vol. 9, no. 3, Mar. 2014, Art. no. 036006.
[34] Z. Zhang, C. Gao, Y. Lu, Y. Liu, and M. Liang, "Multi-objective ant colony optimization based on the physarum-inspired mathematical model for bi-objective traveling salesman problems," *PLoS ONE*, vol. 11, no. 1, Jan. 2016, Art. no. e0146709.
[35] Y. Liu, C. Gao, Z. Zhang, Y. Wu, M. Liang, L. Tao, and Y. Lu, "A new multi-agent system to simulate the foraging behaviors of physarum," *Natural Comput.*, vol. 16, no. 1, pp. 15–29, Mar. 2017.
[36] G. Qin and J. Yang, "An improved ant colony algorithm based on adaptively adjusting pheromone," *Inf. Control*, vol. 31, pp. 197–198, 2002.
[37] F. Dahan, K. El Hindi, H. Mathkour, and H. AlSalman, "Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem," *Sensors*, vol. 19, no. 8, p. 1837, Apr. 2019.
[38] E. Nechita, G. C. Crişan, L. B. Iantovics, and Y. Huang, "On the resilience of ant Algorithms. Experiment with adapted MMAS on TSP," *Mathematics*, vol. 8, no. 5, p. 752, May 2020.
[39] A. Tero, R. Kobayashi, and T. Nakagaki, "Physarum solver: A biologically inspired method of road-network navigation," *Phys. A, Stat. Mech. Appl.*, vol. 363, no. 1, pp. 115–119, Apr. 2006.
[40] Y.-P. Gunji, T. Shirakawa, T. Niizato, M. Yamachiyo, and I. Tani, "An adaptive and robust biological network based on the vacant-particle transportation model," *J. Theor. Biol.*, vol. 272, no. 1, pp. 187–200, Mar. 2011.
[41] A. I. Adamatzky, "Route 20, autobahn 7, and slime mold: Approximating the longest roads in USA and Germany with slime mold on 3-D Terrains," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 126–136, Jan. 2014.
[42] T. Umedachi, S. Horikiri, R. Kobayashi, and A. Ishiguro, "Enhancing adaptability of amoeboid robot by synergetically coupling two decentralized controllers inspired by true slime mold," *Adapt. Behav.*, vol. 23, no. 2, pp. 109–121, 2015.
[43] J. Jones and A. Adamatzky, "Computation of the travelling salesman problem by a shrinking blob," *Natural Comput.*, vol. 13, no. 1, pp. 1–16, Mar. 2014.
[44] A. Adamatzky, "Physical maze solvers. All twelve prototypes implement 1961 Lee algorithm," *Emergent Comput.*, vol. 24, pp. 489–504, 2016.

MEIJIAO LIU received the bachelor's degree from Jilin University, Changchun, China, in 2016. She is currently pursuing the Ph.D. degree with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun. Her current research interests include kinematic dynamics of robots and path and trajectory planning of crawling robots.

YANHUI LI received the bachelor's and M.E. degrees from Beihang University, Beijing, China, in 2008 and 2011, respectively. His current research interests include robot motion planning and motion control.

ANG LI received the bachelor's degree from the Dalian University of Technology, Dalian, China, in 2010, and the M.E. degree from Jilin University, in 2014. His current research interests include robot intelligent technology and high-performance computing research.

QI HUO received the bachelor's degree and the M.E. degree in traffic information engineering and control from Harbin Engineering University, Harbin, China, in 2011 and 2014, respectively. His current research interests include kinematics, dynamics, and control of robots.

NING ZHANG received the bachelor's degree from Northeastern University, Qinhuangdao, China, in 2017. She is currently pursuing the Ph.D. degree with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China. Her research interests include image processing, deep learning, and remote sensing image super-resolution.

NAN QU received the Ph.D. degree from Jilin University, Changchun, China, in 2008. She is currently an Associate Professor with the College of Resources and Environmental Sciences, Jilin Agricultural University, Changchun. Her current research interests include intelligent algorithm and chemometrics.

MINGCHAO ZHU received the bachelor's degree in automation from Jilin University, Changchun, China, in 2003, and the Ph.D. degree in control theory and control engineering from Jilin University, in 2009. He is currently a Researcher with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun. His current research interests include kinematics, dynamics, and control of robots.

LIHENG CHEN received the bachelor's degree from Jilin University, Changchun, China, in 2002, the M.E. degree from Jilin University, in 2005, and the Ph.D. degree from the Chinese Academy of Sciences, in 2008. He is currently a Researcher with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun. His current research interest includes thermal control technology of space optical remote sensor.