Complex & Intelligent Systems (2023) 9:3951–3970
https://doi.org/10.1007/s40747-022-00932-1

ORIGINAL ARTICLE

A hybrid algorithm based on state-adaptive slime mold model and fractional-order ant system for the travelling salesman problem

Xiaoling Gong[1] · Ziheng Rong[2] · Jian Wang[3] · Kai Zhang[4,5] · Shengxiang Yang[6]

Received: 27 April 2022 / Accepted: 14 November 2022 / Published online: 15 December 2022

Abstract

The ant colony optimization (ACO) is one efficient approach for solving the travelling salesman problem (TSP). Here, we propose a hybrid algorithm based on state-adaptive slime mold model and fractional-order ant system (SSMFAS) to address the TSP. The state-adaptive slime mold (SM) model with two targeted auxiliary strategies emphasizes some critical connections and balances the exploration and exploitation ability of SSMFAS. The consideration of fractional-order calculus in the ant system (AS) takes full advantage of the neighboring information. The pheromone update rule of AS is modified to dynamically integrate the flux information of SM. To understand the search behavior of the proposed algorithm, some mathematical proofs of convergence analysis are given. The experimental results validate the efficiency of the hybridization and demonstrate that the proposed algorithm has the competitive ability of finding the better solutions on TSP instances compared with some state-of-the-art algorithms.

Keywords Ant system (AS) · Slime mold (SM) · Fractional-order calculus · Travelling salesman problem (TSP) · Convergence proof

Introduction

The travelling salesman problem (TSP) is one of the most intensively studied NP-hard [1] combinatorial optimization problems, which requires a shortest path through all cities without repetition [2,3]. Over the years, considerable bio-inspired heuristic algorithms have been applied to address TSPs, e.g., the ant colony optimization (ACO) [4,5], the particle swarm optimization (PSO) [6,7], the artificial bee colony (ABC) algorithm [8], the firefly algorithm (FA) [9], and earthworm optimization [10].

Particularly, the pioneering ACO algorithm, ant system (AS) [4], was originally developed to solve the TSP [5]. On this account, the TSP is used mostly as a benchmark to test ACO variants. ACO is a swarm intelligence algorithm which simulates the foraging behavior of ant colony [11]. With the characteristics of positive feedback and information sharing among ants, it has become one effective method for solving combinatorial optimization problems. A considerable number of its variants have been designed to boost the performance, including the popular Max–Min ant system (MMAS) [12] and the rank-based ant system (ASrank) [13]. Meanwhile, the convergence proofs of AS and MMAS were presented to explain the rationale behind the ACO theoreti-

✉ Jian Wang
wangjiannl@upc.edu.cn

✉ Kai Zhang
zhangkai@upc.edu.cn

✉ Shengxiang Yang
syang@dmu.ac.uk

Xiaoling Gong
gongxiaoling@s.upc.edu.cn

Ziheng Rong
rongziheng64@163.com

[1] College of Control Science and Engineering, China University of Petroleum (East China), Qingdao 266580, China

[2] Pudong Lingang Middle School Affiliated to Shanghai Normal University, Shanghai 201200, China

[3] College of Science, China University of Petroleum (East China), Qingdao 266580, China

[4] School of Petroleum Engineering, China University of Petroleum (East China), Qingdao 266580, China

[5] School of Civil Engineering, Qingdao University of Technology, Qingdao 266520, China

[6] School of Computer Science and Informatics, De Montfort University, Leicester, UK

cally [14]. Recently, two modified ACO algorithms based on fractional calculus have been proposed [15,16], which integrate the long-time memory property of fractional calculus into updating pheromones from a different perspective.

Generally, higher convergence speed and lower risk of trapping into local optima are the main targets of these extensions. In this context, how to make a dynamic balance between the exploration and exploitation of a swarm, has always been an important issue worthy of study and discussion. Considering the various advantages of different algorithms and search techniques, researchers have proposed hybrid intelligence algorithms by combining multiple methods. In [17], a parallel cooperative hybrid algorithm, PACO-3Opt, is developed to address the TSP, where the main idea is generating solutions by ACO in parallel, and then integrating the 3-Opt local search heuristic to modify these solutions. Another hybrid method called PSO-ACO-3Opt [18] utilizes PSO algorithm to adjust the parameters of ACO, and applies the 3-Opt technique to avoid premature convergence of ACO algorithm. These works show the effectiveness of the hybridization experimentally. However, there remains much room for improvement in accelerating convergence and jumping out of local optimal solutions. Moreover, they have made no attempt to provide the theoretical analysis of the convergence.

On the other hand, recently, there have been increased interests in slime mold (SM) model, due to its unique biological mechanism and the intelligent behavior of network design and path finding. An SM is an amoeba-like single-celled organism with a tubular network structure and flowing protoplasm, which can adaptively adjust its body shape to construct a shortest path [19]. During foraging, the tube network, which contains nutrients and chemical signals that circulate throughout the organism, forms by connecting the growing point to food sources. These tubes are disassembled and reorganized according to external conditions, such as the quantity and position of nutrient sources, the terrains, and the light intensity distribution. In this way, the network structure of the organism is optimized to facilitate the efficient access to available nutrients [20]. For instance, when an SM is placed in a maze with two nutrient sources positioned at the entrance and exit, it modifies its body shape dynamically to absorb nutrients efficiently, and eventually connect the entrance and exit (a pair of food sources) with the shortest path [21]. Biological experiments have indicated that the simulation network constructed by SM model is very similar to the real Tokyo, Mexico, and Greece railway networks in terms of transportation performance, robustness, and cost [22–24]. Therefore, a biologically inspired mathematical model of SM has been developed with the positive feedback characteristic, which means the tube diameter increases with the increase of the protoplasm flux, and decreases otherwise [25]. Thereafter, the SM model has been successfully applied to solve combinatorial optimization, including the knapsack problem [26], traffic network optimization [27,28], load-shedding problem [29], supply chain network design [30,31], etc.

This model attracts our attention because apparently, the path optimization ability of SM model makes it a good candidate approach to solving the TSP. We noticed that a recent research has presented a slime mold-ant colony fusion algorithm (SMACFA) to deal with TSP, which uses SM model to determine the connections of some edges before ACO is implemented [32]. This approach is more like the initialization process before ACO, which takes the advantage of SM in shortest path construction. However, in this way, the variety of path selection is limited, and the advantages of SM model are not brought into full play.

The purpose of our work is to develop an effective approach to hybridize the AS algorithm with the SM model to bring the ability of SM model of constructing shortest paths into full play. To this end, a novel hybrid algorithm based on state-adaptive SM model and fractional-order AS (SSMFAS) is developed in this paper for addressing the TSP. Moreover, some convergence proofs are given to ensure the feasibility of the proposed hybrid algorithm theoretically. The main contributions of this paper are summed up as follows.

1. To make the SM model more suited for solving the TSP cooperating with the ant colony, we have developed a state-adaptive multi-entrance-exit SM model where each pair of nodes is regarded as the entrance/exit of an SM subsystem, equipped with two auxiliary strategies. The adaptive conductivity strategy changes the contraction rate smoothly over iterations, so that SM model shifts its state from exploration to exploitation to achieve a dynamic balance between local and global optimization. The maximum–minimum flux strategy is introduced to limit the flux through the tubes within an appropriate range matching the pheromone in the AS.
2. A hybrid algorithm based on state-adaptive SM model and fractional-order AS (SSMFAS) is proposed, where a fractional-order neighborhood probability is used in the node transition of ants to take the information of the neighborhood of candidate cities into account by fractional-order calculus, and a modified pheromone update rule is proposed which adds the protoplasm flux through each pipe as the pheromone on the corresponding edge dynamically to make the ant colony obtain information from the SM model.
3. The convergence properties of SSMFAS are analyzed in this paper. In simple terms, first, it has been proved that the probability of SSMFAS in finding an optimal path arbitrarily approaches to 1 given enough iterations. Second, we have proved that the pheromone trails of the optimal edges increase over iterations after the optimal solution is

found, while the pheromone trails of other edges decrease. In addition, the lower bound of the probability of finding an optimal solution is given.

4. Abundant experiments have been conducted with promising results. First, ablation studies are implemented to validate the rationality of every module of the hybridization as well as the proposed two auxiliary strategies. Subsequently, comparisons with several state-of-the-art algorithms, including ACO-based hybrid algorithms and other heuristic approaches, are made to illustrate the effectiveness of SSMFAS.

The remainder of this paper is organized as follows. In "Background" section, we briefly introduce some algorithms and theories related to our work. The details of SSMFAS are presented in Methodology section. The theoretical convergence properties and the proofs are given in Theoretical convergence proofs section. Experiments section demonstrates the simulation results and the relevant analysis. Finally, Conclusions section summarizes the conclusion.

Background

In this section, we first give the definition of TSP and the notations used in the paper. Then we briefly introduce the framework of the AS algorithm, the mathematical model of SM and the background knowledge related to fractional calculus.

Problem definition and notations

In a TSP, assume the number of cities is $N \in \mathbb{N}^+$ and the cities are labeled as $i = 1, 2, \cdots, N$. The distance between city i and city j is given as L_{ij}. The objective is to find the shortest route visiting each city once and returning to the starting point. Therefore, each solution to the problem is a sequence with length of N, i.e., $\mathbb{C} = (C_1, C_2, \cdots, C_N)$. The cost function of the solution is $S = L_{C_1C_2} + L_{C_2C_3} + \cdots + L_{C_NC_1}$.

For the sake of the reader's convenience, the mathematical notations used in the paper are listed in Table 1.

Ant system algorithm

Route construction and pheromone update are two critical steps of the AS algorithm. The process of the AS is summarized as follows:

1) *Initialization:* The population of ants is set to $M \in \mathbb{N}^+$. Every ant is placed at a random city as a start position.

Table 1 Summary of notations

Notation	Description	Notation	Description
N	Number of cities/nodes	Q_{ij}	Flux through the edge (i, j)
M	Population of ants	p_i	Pressure at node i
M'	Number of elitist ants	D_{ij}	Conductivity of edge (i, j)
S	Length of the solution	ω	Viscosity of the fluid
L_{ij}	Length of edge (i, j)	a_{ij}	Radius of pipe (i, j)
τ_{ij}	Pheromone concentration on edge (i, j)	I_0	Fixed total flux
η_{ij}	Heuristic information on edge (i, j)	γ	Contraction rate of tubes
α	Parameter determining pheromone influence	ν	Fractional order
β	Parameter determining heuristic information influence	μ	Parameter determining the influence of q_{ij}
J_i^m	Set of cities which have not been visited by ant m	δ	Bound parameter
ρ	Evaporation rate of pheromone	λ	A control coefficient
$T_{\max}$	The maximum number of iterations	$\sigma(t)$	Parameter used to adjust the influence of protoplasm flux

The initial value of pheromone on each edge is set to $\tau_0 > 0$.

2) *Route construction:* Each ant m $(m = 1, 2, \ldots, M)$ in the current city i selects the next visiting city j depending on a random proportion rule of the transition probability, which is defined as

$$p_{ij}^m = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{u \in J_i^m} [\tau_{iu}]^\alpha [\eta_{iu}]^\beta}, & \text{if } j \in J_i^m, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where τ_{ij} represents the pheromone concentration on edge (i, j), η_{ij} is called the heuristic information which is calculated as $\eta_{ij} = \frac{1}{L_{ij}}$, where L_{ij} is the length of edge (i, j), α and β are two preset parameters used to balance the effects of heuristic information and the pheromone concentration, J_i^m represents the set of cities which have not been visited by ant m.

Every ant keeps moving from one city to another guided by this probability and records the visited cities until it has visited all the cities.

3) *Pheromone update:* At time $(t + 1)$, after each ant completing its tour, the pheromone on edge (i, j) is updated as

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{m=1}^{M} \Delta\tau_{ij}^m(t), \tag{2}$$

where ρ $(0 < \rho < 1)$ denotes the evaporation rate of the pheromone, $\Delta\tau_{ij}^m$ represents the volume of pheromone released by ant m on edge (i, j), which is calculated as

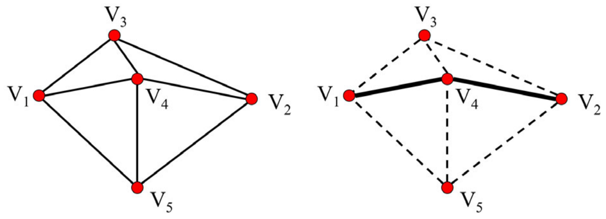
$$\Delta\tau_{ij}^m(t) = \begin{cases} (S^m)^{-1}, & \text{if } (i, j) \text{ is visited by ant } m, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where S^m is the length of solution constructed by ant m.

4) Repeat the above steps until the termination condition is met, and then output the optimal solution.

Slime mold model

The SM model has a tube-liked network, which contains N nodes and $N(N-1)/2$ edges connecting the nodes, indicating the cytoplasmic tubules between the nodes. Each edge has a weight, denoted by Q, which means the flux through the edge. Figure 1 illustrates the path construction process of an SM network. V_1 and V_2 are the entrance and exist nodes, respectively. V_3, V_4 and V_5 are transition nodes. As shown in Fig. 1a, the classic SM model has only one growth point (entrance node), V_1, and one food source (exist node), V_2. V_3 to V_5 are transition nodes. The protoplasmic flux in the tube is similar to Hagen–Poiseuille flux [25,33]. Therefore, the flux through tube (i, j), Q_{ij}, is formulated as



(a) Initial structure **(b)** Final structure

Fig. 1 Penels (**a**) and (**b**) illustrates the path construction process of an SM network with one pair of entrance/exist nodes

$$Q_{ij} = D_{ij} \cdot \frac{p_i - p_j}{L_{ij}}, \tag{4}$$

where L_{ij} is the length of edge (i, j), p_i and p_j denote the pressures at nodes i and j, and D_{ij} is used to describe the conductivity of edge (i, j), which is determined by

$$D_{ij} = \frac{\pi a_{ij}^4}{8\omega}, \tag{5}$$

where ω represents the viscosity of the fluid, and a_{ij} denotes the radius of pipe (i, j).

Following the Kirchhoff's law [34], the total flux should be conserved. Therefore, the outflow and inflow of each internal node must be equal, except for the start and end nodes. Let I_0 be the fixed total flux from the entrance to the exit in the entire network. This conservation law is modeled as

$$\sum_i Q_{ij} = \sum_i \frac{D_{ij}}{L_{ij}} \left(p_i - p_j\right) = \begin{cases} -I_0, \text{ if } j = \text{entrance}, \\ I_0, \text{ if } j = \text{exit}, \\ 0, \text{ otherwise}. \end{cases} \tag{6}$$

Given an initial conductivity D_0 for every edge, and $p_{\text{exit}} = 0$ as a fundamental pressure at the exit node, the linear equation system with sparse symmetric matrix (6) can be solved for the pressure value at each node, as well as the flux Q_{ij} through each edge.

The slime mold adapts itself when foraging in a way that high-flow pipes are thickened, while low-flow pipes shrink and disappear. Since the length L_{ij} is a given constant, the adaptive characteristic of the model can be expressed by the variation of conductivity D_{ij}, modeled as

$$\frac{d}{dt} D_{ij} = f\left(|Q_{ij}|\right) - \gamma D_{ij}, \tag{7}$$

where $f\left(|Q_{ij}|\right)$ is a monotonically increasing function with $f(0) = 0$, which describes the positive effect of increased flow on conductivity, $\gamma > 0$ is the contraction rate of tubes, which denotes the tubes contracts over time.

The discretized expression of (7) is expressed as

$$\frac{D_{ij}^{t+\Delta t} - D_{ij}^{t}}{\Delta t} = f\left(|Q_{ij}^{t}|\right) - \gamma D_{ij}^{t}, \tag{8}$$

where Δt represents a time interval which is typically considered as 1. Therefore, (8) is rewritten as

$$D_{ij}^{t+1} = f\left(|Q_{ij}^{t}|\right) + (1-\gamma) D_{ij}^{t}. \tag{9}$$

From (4) and (9), it is noteworthy that there is a positive feedback cycle between the conductivity and the flux. That is, the conductivity of the edge, D_{ij}, increases when a larger amount of flux, Q_{ij}, passes through it, and this is conducive to the increase of flux further.

Under this mechanism, the network structure of the SM model in Fig. 1a changes to the one in Fig. 1b after a few iterations, where some tubes are disappeared (marked as the dotted lines), while some other tubes are thickened and stabilized (highlighted with bold black lines).

Fractional calculus

Fractional-order calculus is an extension of the integer calculus, where the order extends from an integer to a real number. As a mathematical tool, it has been investigated intensively and applied successfully in various areas [35]. The most popular used definitions of fractional calculus are Grünwald–Letnikov, Riemann–Liouville, Caputo, and Riesz [36–39]. In this work, we employ Grünwald–Letnikov fractional form because it can be discretized. For a differentiable function $g(x)$ in the duration of $[a, x]$, its Grünwald–Letnikov fractional-order differential definition of order ν $(\nu > 0)$ is defined as

$$ {}_{a}^{G-L}D_{x}^{\nu} g(x) = \lim_{H\to\infty} \left\{ \frac{\left(\frac{x-a}{H}\right)^{-\nu}}{\Gamma(-\nu)} \sum_{l=0}^{H-1} \frac{\Gamma(l-\nu)}{\Gamma(l+1)} g\left[x - l\left(\frac{x-a}{H}\right)\right] \right\}, \tag{10}$$

where ${}_{a}^{G-L}D_{x}^{\nu}$ denotes fractional differential operator, $\Gamma(\alpha) = \int_0^\infty e^{-x} x^{\alpha-1} dx$ is the Gamma function, $\frac{x-a}{H}$ is the sampling step length.

Let $\Delta x = \frac{x-a}{H}$. In the duration of $[x - H\Delta x, x]$, (10) is simplified as

$$\begin{aligned} {}_{a}^{G-L}\mathrm{Diff}_{x}^{\nu} g(x) &= (\Delta x)^{-\nu} \sum_{l=0}^{H-1} \frac{\Gamma(l-\nu)}{\Gamma(-\nu)\Gamma(l+1)} g(x - l\Delta x) \\ &= (\Delta x)^{-\nu} \left[g(x) + \sum_{l=1}^{H-1} \frac{\Gamma(l-\nu)}{\Gamma(-\nu)\Gamma(l+1)} g(x - l\Delta x) \right], \end{aligned} \tag{11}$$

when $\nu = 1$, it is obvious that (11) can be revised as

$$ {}_{a}^{G-L}\mathrm{Diff}_{x}^{1} g(x) = (\Delta x)^{-1} \left[g(x) - g(x - \Delta x) \right], \tag{12}$$

where ${}_{a}^{G-L}\mathrm{Diff}_{x}^{1} g(x)$ is the first-order difference.

By comparing (11) with (12), we observe that the fractional-order difference includes more history information, which is generally regarded as the property of long-term memory.

Methodology

In this section, we elaborate on the proposed SSMFAS, including the improved SM model and details of key procedures.

Figure 2 shows the overall procedure of the proposed SSMFAS. After the initialization, a state adaptive multi-entrance-exit SM model is developed, where the positive feedback mechanism is used to update the flux and the conductivity. It consists of two strategies. The adaptive contraction strategy shifts the state of SM from exploration to exploitation. The maximum–minimum flux strategy limits the flux by the upper and lower bounds of AS algorithm, making it accessible for linking with AS algorithm. For the ant colony part, when the ants constructing solutions, the fractional-order neighborhood transition probabilistic rule which uses the neighboring information is introduced. After that, the flux of SM model is added to pheromone update rule as a part of adaptive pheromone. This loop continues until the stopping criterion is satisfied.

State-adaptive multi-entrance-exit SM model

We have introduced the classic single-entrance-exit SM model in Background section. However, it may not be suitable for solving the TSP directly since there is no fixed pair of entrance/exit in the TSP [40]. Therefore, we propose a state-adaptive multi-entrance-exit SM model where each pair of nodes is considered as a single-entrance-exit SM subsystem to address the TSP.

Let the number of the nodes in the improved SM model be N, which corresponds to the number of cities in AS. Accordingly, there are $N(N-1)/2$ single-entrance-exit SM

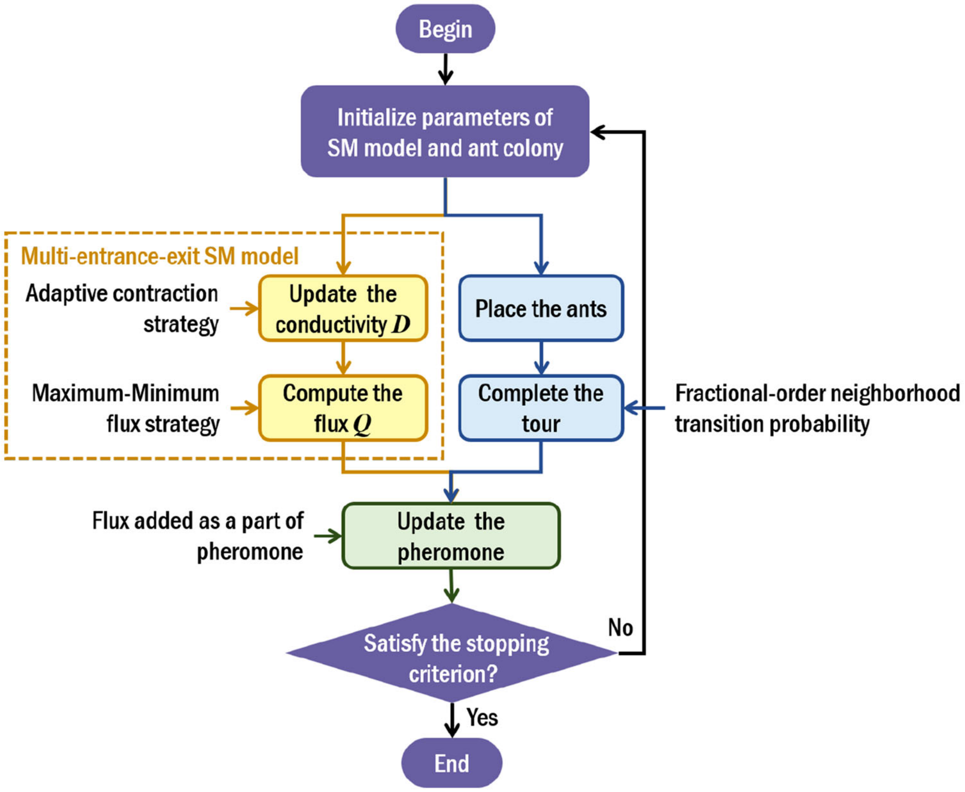

Begin
Initialize parameters of SM model and ant colony
Multi-entrance-exit SM model
Adaptive contraction strategy
Update the conductivity D
Maximum-Minimum flux strategy
Compute the flux Q
Place the ants
Complete the tour
Fractional-order neighborhood transition probability
Flux added as a part of pheromone
Update the pheromone
Satisfy the stopping criterion?
No
Yes
End

Fig. 2 The flowchart of the proposed SSMFAS

subsystems since each pair of nodes is considered as the entrance/exit. Given the pressure at exit node as $p_{\text{exit}} = 0$, the pressure at any other node in the nth ($n = 1, 2, \cdots, N(N-1)/2$) sub-network is derived from

$$\sum_{i \neq j} \frac{D_{ij}^n}{L_{ij}} \left(p_i^n - p_j^n \right) = \begin{cases} \dfrac{-2I_0}{N(N-1)}, & \text{for } j = \text{entrance}, \\ \dfrac{2I_0}{N(N-1)}, & \text{for } j = \text{exit}, \\ 0, & \text{otherwise}, \end{cases} \tag{13}$$

where I_0 denotes the total flux in the model, which is fixed during the optimization process.

Then the flux $q_{ij}^n(t)$ through the tube (i, j) in the nth subsystem at time t is computed as

$$q_{ij}^n(t) = \frac{D_{ij}^n(t)}{L_{ij}} \left(p_i^n(t) - p_j^n(t) \right). \tag{14}$$

The total flux Q_{ij} through the tube (i, j) is calculated by the sum of the flux in each subsystem, expressed as

$$Q_{ij} = \sum_{n=1}^{N(N-1)/2} |q_{ij}^n|. \tag{15}$$

According to (9), the conductivity of tube (i, j) at time $(t + 1)$ in the nth sub-network is updated by:

$$D_{ij}^n(t+1) = f\left(|q_{ij}^n(t)| \right) + (1-\gamma) D_{ij}^n(t), \tag{16}$$

where γ is the contraction rate of tubes and $f(\cdot)$ is a monotonically increasing function.

To make the improved model more applicable to the TSP, instead of using the two classical types of $f(\cdot)$ [25], we introduce a function as

$$f(|q_{ij}^n(t)|) = \mu \cdot |q_{ij}^n(t)|, \tag{17}$$

where μ is a parameter which controls the influence of q_{ij}. Its value is optimized by orthogonal experiments in Experiments section.

Adaptive contraction strategy

Specifically, as the intention of integrating SM model with AS is to improve the global searching and convergence ability, we introduce an adaptive strategy, which changes the contraction rate of tubes, r, dynamically over iterations, to make the SM model maintain a dynamic balance between convergence speed and global searching ability.

More specifically, at the beginning of the iteration when the global optimal has not been found, the algorithm should

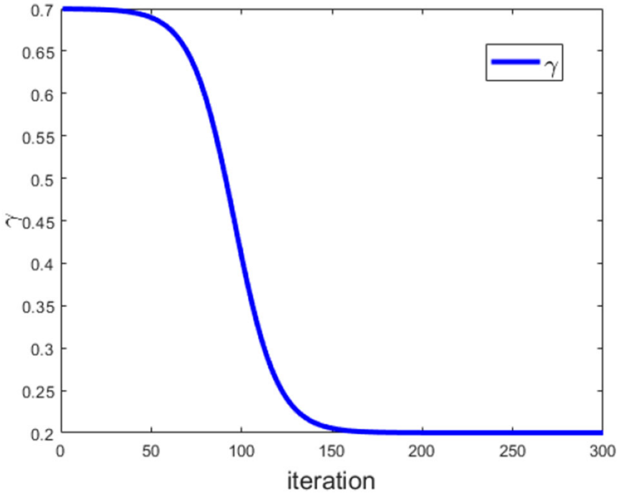


Fig. 3 The variation of contraction rate γ over iterations

be more sensitive to various solutions, which is represented by a larger value of the contraction rate of tubes. Subsequently, the contraction rate of tubes should decrease with the number of iterations increasing, for the sake of accelerating the convergence of the algorithm.

To achieve this goal, we propose a smooth decreasing function within a range of (a, b), which is defined as

$$\gamma(t) = a\left[1 - \frac{1}{\frac{a}{a-b} + (c \cdot e)^{T'-t}}\right], \tag{18}$$

where T' is an inflection point when the algorithm shifts from the period of exploration to exploitation, and c is a coefficient to adjust the speed of the transition.

The variation of the contraction rate $\gamma(t)$ over iterations is plotted in Fig. 3, with $a = 0.7$, $b = 0.2$, and $T' = \frac{T_{\max}}{3}$. It shows that the value of γ is higher in the early search, which makes the conductivities of the tubes in SM sensitive to the flow, leading to the piping connections changing rapidly. In this way, the SM is in a more active state to look for various solutions at this stage. As the number of iterations increases, the value of the contraction rate decreases, which means that the conductivity changes slowly to form the optimal path gradually. In other words, the SM changes from active state to stable state over successive iterations.

Maximum–minimum flux strategy

Furthermore, there is another important point need to be consider when we try to combine the SM model with AS using the flux of SM model as a kind of pheromone. That is whether flux and pheromone, the two variables belonging to different models, keep similar orders of magnitude.

We notice that one simple but quite efficient mechanism of MMAS [12] is limiting the values of pheromone to the maximum and minimum ranges. Inspired by this, we consider an upper bound and a lower bound to limit the flux through each tube to a suitable range. More specifically, to better integrate with AS algorithm, the bounds should be related to the pheromone in AS, which is set to

$$Q_{ij}(t) \in \delta \cdot [\tau_{\min}, \tau_{\max}], \tag{19}$$

where $\tau_{\min}$ and $\tau_{\max}$ are the upper and lower bounds of the pheromone, respectively, $\delta > 0$ is the weight coefficient, which is set based on some trial experiments in the algorithm.

The pseudo-code of the state adaptive multi-entrance-exit SM model is outlined in Algorithm 1.

Algorithm 1 State Adaptive Multi-entrance-exit SM Model

```
Input: The length matrix $L$;
       the maximum number of iterations $T_{max}$;
       the number of cites $N$;
Output: The total flux $Q_{ij}(t)$ at time $t$;
Initialize the conductivity matrix $D_0^n$; the flow of the entire network $I_0$;
for $t = 0$ to $T_{max}$ do
    for $n = 1$ to $(N - 1)$ do
        Calculate $p^n(t)$ at each node based on (13);
        Obtain $q_{ij}^n(t)$ according to (14);
        Update the conductivity $D_{ij}^n$ using (16);
    end for
    Calculate $Q_{ij}(t)$ using (15);
end for
```

Fractional-order neighborhood transition probability

In general ACO, an ant selects a movement using only the information between two nodes, which brings about a higher risk of trapping into local optima, as the best node for the next step is not always the best choice for the total path. Recently, a fractional-order ant colony algorithm (FACA) [16] has been proposed which calculates the probabilities of next few steps, and combines them with fractional-order calculus to get a more predictable probability. Similarly, another way of combined probability has been proposed in our previous work [15]. However, these algorithms need to calculate the transition probabilities of a few more nodes ahead of the current node. Therefore, the computational complexity is multiplied by the number of steps that the algorithm counts in.

To overcome this limitation, our goal is to take full account of the neighbor information without increasing too much computational cost. Here, we propose a fractional-order neighborhood transition probability, which combines the moving probability of the candidate node with those of its neighboring nodes to form a new transition probability. Specifically, when an ant m $(m = 1, 2, \dots, M)$ is at the ith node, the transition probability to the jth node at the tth iteration is defined as

$$
{}^{\nu}p_{ij}^{m}(t) = \begin{cases} \dfrac{p_{ij}^{m}(t) + \sum_{k=1}^{N'-1} |\frac{\Gamma(k-\nu)}{\Gamma(-\nu)\Gamma(k+1)}| p_{in_k}^{m}(t)}{\sum_{k=0}^{N'-1} |\frac{\Gamma(k-\nu)}{\Gamma(-\nu)\Gamma(k+1)}|}, & \text{if } j \in J_i^m(t), \\ 0, & \text{otherwise.} \end{cases} \tag{20}
$$

where ν is the fractional order, $p_{ij}^{m}(t)$ is calculated by (1), $(N' - 1)$ is the number of nodes considered in the neighborhood of the node j, n_k means the unvisited node which ranks k in descending order of Euclidean distance from node j, the denominator part is used to normalize this combination of probabilities.

For the candidate node j, the proposed fractional-order neighborhood transition probability considers the information in the neighborhood of j. This modification only needs the values of probabilities of the candidate cities for city i, which have been calculated already. Furthermore, it is logically reasonable that $|\frac{\Gamma(k-\nu)}{\Gamma(-\nu)\Gamma(k+1)}|$ is a monotonically decreasing function of k, meaning the effect of its information decreases as the distance from the neighbor node to j increases.

Pheromone update

Algorithm 2 SSMFAS Algorithm

```
Input: The length Matrix $L$, maximum number of iterations $T_{max}$;
Output: The best solution;
Set iteration $t = 0$;
Initialize the parameters;
Place ants in starting cities randomly;
while $t \le T_{max}$ do
    Each ant completes the tour using fraction-order neighborhood transition probability (20);
    Local optimization and solutions sorting;
    Use Algorithm 1 to get $Q_{ij}(t)$;
    Update the pheromone on each edge according to (22);
    $t = t + 1$;
end while
Local optimization;
Return the best solution.
```

When all ants have finished their loops at the tth iteration, the pheromone trails are updated. The pheromone update rule of SSMFAS consists of two parts: the pheromone trails left by ant colony and the protoplasm flux of the SM model.

For the component of pheromone trails, similar to a rank-based version of the AS (AS$_{\text{rank}}$) [13], only the pheromone trails of some elite ants with ranks count. First, we sort the solutions constructed by M ants at the tth iteration in ascending order, written as

$$
S^1(t) \le S^2(t) \le \dots \le S^M(t). \tag{21}
$$

Then, only the M' $(1 \le M' \le M)$ best ants are used to update the pheromone trials and their effects are based on the ranks of their solutions. We use a nonlinear decreasing function with 1 as the largest weight instead of the smallest used in AS$_{\text{rank}}$, since the flux in SM model is also added as the pheromone in this procedure.

The pheromone update policy of SSMFAS is given by

$$
\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{m=1}^{M'} e^{-\lambda(m-1)} \cdot \Delta\tau_{ij}^{m}(t) + \sigma(t) Q_{ij}(t), \tag{22}
$$

and

$$
\tau_{\min} \le \tau_{ij} \le \tau_{\max} \quad \forall \tau_{ij}, \tag{23}
$$

where M' $(1 \le M' < M)$ is the number of chosen ants, λ denotes a control coefficient which is set to 0.1 in the experiments, $\sigma(t)$ is a control parameter used to adjust the influence of protoplasm flux, which is defined as

$$
\sigma(t) = 1 - \frac{1}{1 + e^{\frac{T_{\max}}{4} - t}}. \tag{24}
$$

Similar to (18), $\sigma(t)$ is a decreasing function within the range of (0, 1), which means the effect of SM model decreases with the number of iterations increasing. The reason for this is that the addition of the flux of SM model changes the pheromone distribution of the ant colony, and thus influences the edge selections of ants. At the beginning of the iteration, the algorithm needs to use the flux of SM model as part of information (pheromone) to guide ants to find some good edges. However, when the path search in SM model converges gradually, the algorithm reduces the effect of the flux to prevent excessive pheromone at some edges which would make it difficult for ants to find other solutions and thus fall into the local minimum. The adaptive hybridization of the two intelligent algorithms is conducive to better solutions.

The general framework of SSMFAS is shown in Algorithm 2.

Theoretical convergence proofs

We analyze the convergence properties of SSMFAS in this section from several aspects, including the pheromone values and the probability of finding the optimal solution.

Proposition 1 *For SSMFAS, the pheromone value $\tau_{ij}(t)$ on an arbitrary edge (i, j) at any time t satisfies that*

$$\lim_{t\to\infty} \tau_{ij}(t) \le \tau_{\max} = \frac{\Delta_{max}}{\rho}, \tag{25}$$

where Δ_{max} is the maximum value of pheromone (including flux) added to an arbitrary edge, determined as

$$\Delta_{max} = \sum_{m=1}^{M'} \frac{e^{-\lambda(m-1)}}{S_{min}} + Q_{max} \ge 0, \tag{26}$$

where S_{min} is the theoretical shortest solution, Q_{max} is the upper bound of flux.

Proof Based on (22), the maximum pheromone value on edge (i, j) at time 1 is derived as

$$\tau_{ij}^{\max}(1) = (1-\rho)\tau_{ij}(0) + \Delta_{\max}. \tag{27}$$

Accordingly, the maximum pheromone value on edge (i, j) at time 2 is obtained by

$$\begin{aligned}\tau_{ij}^{\max}(2) &= (1-\rho)\tau_{ij}(1) + \Delta_{\max} \\ &= (1-\rho)^2\tau_{ij}(0) + (1-\rho)\Delta_{\max} + \Delta_{\max}.\end{aligned} \tag{28}$$

In the same manner, the maximum pheromone value on edge (i, j) at iteration t is derived as

$$\tau_{ij}^{\max}(t) = (1-\rho)^t\tau_{ij}(0) + \sum_{i=1}^{t}(1-\rho)^{t-i}\Delta_{\max}. \tag{29}$$

As the pheromone evaporation rate, ρ, satisfies $0 < \rho < 1$, based on the formula of summation for geometric sequence, we obtain that

$$\begin{aligned}\lim_{t\to\infty} \tau_{ij}^{\max}(t) &= \lim_{t\to\infty}\left[(1-\rho)^t\tau_{ij}(0) + \frac{1-(1-\rho)^t}{\rho}\Delta_{\max}\right] \\ &= \frac{\Delta_{\max}}{\rho}.\end{aligned} \tag{30}$$

Thus, we have

$$\lim_{t\to\infty} \tau_{ij}(t) \le \lim_{t\to\infty} \tau_{ij}^{\max}(t) = \frac{\Delta_{\max}}{\rho}. \tag{31}$$

Thus, the proof is completed. □

Proposition 2 *On the premise that an optimal solution $S_{\min}$ is found at the t^*th iteration, for the pheromone value, $\tau_{ij}(t)$, it holds that*

$$\lim_{t\to\infty} \tau_{ij}(t) = \frac{\Delta\tau_{max}}{\rho}, \tag{32}$$

where edge $(i, j) \in S_{\min}$, $\Delta\tau_{\max} = (S_{\min})^{-1}$ is the maximum pheromone concentration added by ants.

Proof The pheromone updating expression (22) can be rewritten as

$$\tau_{ij}(t+1) = \tilde{\tau}_{ij}(t) + \sigma(t)Q_{ij}(t), \tag{33}$$

where

$$\tilde{\tau}_{ij}(t) = (1-\rho)\tau_{ij}(t) + \sum_{m=1}^{M'} e^{-\lambda(m-1)} \cdot \Delta\tau_{ij}^m(t). \tag{34}$$

Thus, we have

$$\lim_{t\to\infty} \tau_{ij}(t) = \lim_{t\to\infty} \tilde{\tau}_{ij}(t-1) + \lim_{t\to\infty} \sigma(t-1)Q_{ij}(t-1). \tag{35}$$

It is concluded from (24) that

$$\lim_{t\to\infty} \sigma(t-1) = 0. \tag{36}$$

Since $Q_{ij}(t-1)$ has an upper bound according to (19). It holds that

$$\lim_{t\to\infty} \sigma(t-1)Q_{ij}(t-1) = 0. \tag{37}$$

Next, we only need to prove $\lim_{t\to\infty} \tilde{\tau}_{ij}(t-1) = \frac{\Delta\tau_{\max}}{\rho}$.

On the premise that an optimal solution S_{min} is found at the t^*th iteration, the value of pheromone trail on each edge of the optimal path at the (t^*+1)th iteration is given as

$$\tilde{\tau}_{ij}(t^*+1) = (1-\rho)\tau_{ij}(t^*) + \Delta\tau_{max}. \tag{38}$$

It can be concluded from (38) that

$$\tilde{\tau}_{ij}(t^*+t) = (1-\rho)^t\tau_{ij}(t^*) + \sum_{i=1}^{t}(1-\rho)^{t-i}\Delta\tau_{\max}. \tag{39}$$

Thus, by taking the limits of both sides of (39), we have

$$\begin{aligned}&\lim_{t\to\infty} \tilde{\tau}_{ij}(t^*+t) \\ &= \lim_{t\to\infty}\left[(1-\rho)^t\tau_{ij}(t^*) + \sum_{i=1}^{t}(1-\rho)^{t-i}\Delta\tau_{max}\right] \\ &= \lim_{t\to\infty}\left[(1-\rho)^t\tau_{ij}(t^*) + \frac{1-(1-\rho)^t}{\rho}\Delta\tau_{\max}\right] \\ &= \frac{\Delta\tau_{\max}}{\rho}.\end{aligned} \tag{40}$$

This then completes the proof. □

Theorem 1 *Let $P^*(t)$ be the probability of SSMFAS finding an optimal route S_{min} at least once in the first t steps. Supposing that t is sufficiently large, for an arbitrarily small positive number $0 < \varepsilon < 1$, it holds that*

$$P^*(t) \geq 1 - \varepsilon, \tag{41}$$

and

$$\lim_{t \to \infty} P^*(t) = 1. \tag{42}$$

Proof Since the pheromone concentration on each edge is bounded by $\tau_{ij}(t) \in [\tau_{\min}, \tau_{\max}]$, the minimum value of the probability of choosing an optimal edge, ${}^{\nu}p_{\min}$ has a lower bound given by

$${}^{\nu}p_{\min} \geq \frac{p_{\min} + \sum_{k=1}^{N'-1} |\frac{\Gamma(k-\nu)}{\Gamma(-\nu)\Gamma(k+1)}| p_{\min}}{\sum_{k=1}^{N'-1} |\frac{\Gamma(k-\nu)}{\Gamma(-\nu)\Gamma(k+1)}|}, \tag{43}$$

where $p_{\min}$ is reached theoretically when the pheromone values are $\tau_{\min}$ on the optimal edges, and $\tau_{\max}$ on all the other edges, which is written as

$$p_{\min} = \frac{\tau_{\min}^{\alpha} \eta_{\min}^{\beta}}{\tau_{\min}^{\alpha} \eta_{\min}^{\beta} + (N-1)\tau_{\max}^{\alpha} \eta_{\max}^{\beta}}. \tag{44}$$

Accordingly, the probability of finding an optimal solution $S_{\min}$ is expressed as

$$({}^{\nu}p_{\min})^{(N-1)} > 0. \tag{45}$$

Thus, the maximum probability of not finding the optimal solution $S_{\min}$ at the tth iteration is expressed as

$$\tilde{P}(t) = [1 - ({}^{\nu}p_{\min})^{(N-1)}]^t. \tag{46}$$

Therefore, the probability of SSMFAS searching out the optimal route $S_{\min}$ at least once in the first t steps is given as

$$\begin{aligned} P^*(t) &\geq 1 - \tilde{P}(t) \\ &= 1 - [1 - ({}^{\nu}p_{\min})^{(N-1)}]^t. \end{aligned} \tag{47}$$

On the premise that t is sufficiently large, we conclude that there exists an arbitrarily small ε such that

$$P^*(t) \geq 1 - \varepsilon. \tag{48}$$

Also, we obtain that

$$\lim_{t \to \infty} P^*(t) = 1. \tag{49}$$

Therefore, the proof is completed. □

Theorem 2 *For $\forall t > t^* + t_0$, where $t_0 = \frac{1-\rho}{\rho}$, it holds that*

$$\tau_{ij}(t) > \tau_{kl}(t), \quad \forall(i,j) \in S_{\min} \text{ and } \forall(k,l) \notin S_{\min}. \tag{50}$$

Proof Under the worst circumstance that the pheromone concentration is $\tau_{ij}(t^*) = \tau_{\min}$ on optimal edge $(i, j) \in S_{\min}$, and $\tau_{kl}(t^*) = \tau_{\max}$ on non-optimal edge $(k,l) \notin S_{\min}$ at time t^*, $\tau_{ij}(t^* + t')$ is expressed as

$$\begin{aligned} \tau_{ij}(t^* + t') &= (1-\rho)^{t'} \tau_{ij}(t^*) + \sum_{i=0}^{t'-1} (1-\rho)^i \Delta_{\max} \\ &= (1-\rho)^{t'} \tau_{\min} + \sum_{i=0}^{t'-1} (1-\rho)^i \Delta_{\max} \\ &> t'(1-\rho)^{t'-1} \Delta_{\max}, \end{aligned} \tag{51}$$

and $\tau_{kl}(t^* + t')$ is given as

$$\begin{aligned} \tau_{kl}(t^* + t') &= \max\left\{\tau_{\min}, (1-\rho)^{t'} \tau_{kl}(t^*)\right\} \\ &= \max\left\{\tau_{\min}, (1-\rho)^{t'} \tau_{\max}\right\}. \end{aligned} \tag{52}$$

Let $\tau_{kl}(t^* + t')$ be $(1-\rho)^{t'} \tau_{\max}$ for our purpose here. Therefore, $\tau_{ij}(t^* + t') > \tau_{kl}(t^* + t')$ holds when

$$t'(1-\rho)^{t'-1} \Delta_{\max} > (1-\rho)^{t'} \tau_{\max}, \tag{53}$$

which is equal to

$$t' > \frac{\tau_{\max}(1-\rho)}{\Delta_{\max}} = \frac{1-\rho}{\rho} = t_0. \tag{54}$$

Therefore, the proof is completed. □

Proposition 3 *After the optimal solution $S_{\min}$ has been discovered, the value of pheromone $\tau_{kl}(t)$ on any non-optimal edge $(k,l) \notin S_{\min}$ decreases as the number of iterations increases. It holds that*

$$\lim_{t \to \infty} \tau_{kl}(t) = \tau_{\min}. \tag{55}$$

Proof After the t^*th iteration when the optimal path is found, the pheromone on the non-optimal edges evaporates without addition. The pheromone on edge (k,l) at the $(t^* + t')$th iteration is derived as

$$\tau_{kl}(t^* + t') = \max\left\{\tau_{\min}, (1-\rho)^{t'} \tau_{kl}(t')\right\}. \tag{56}$$

Thus, when $t \to \infty$, $\tau_{kl}(t) \to \tau_{\min}$. □

Proposition 4 *Supposing the optimal solution $S_{\min}$ is found at the t^*th iteration, after $(t^* + t_0)$ iterations, it holds that*

$$\tau_{kl}(t) = \tau_{\min} \quad \forall (k, l) \notin S_{\min}, \tag{57}$$

where $t \geq t^* + t_0$ and $t_0 = \frac{ln(\tau_{\min}) - ln(\tau_{\max})}{ln(1-\rho)}$.

Proof Now suppose that t' is the first time such that

$$(1-\rho)^{t'} \tau_{kl}(t') \leq \tau_{\min}. \tag{58}$$

It is derived that

$$t' \geq \frac{ln(\tau_{\min}) - ln(\tau_{\max})}{ln(1-\rho)} = t_0. \tag{59}$$

This then implements the proof. □

Corollary 1 *Suppose that the first optimal solution S_{min} is found at the t^*th iteration. For the probability of ant m constructing the optimal route S_{min} at the tth iteration, $P_m^*(t)$, it holds that*

$$\lim_{t \to \infty} P_m^*(t) \geq C[1 - f(\tau_{max}, \tau_{min})], \tag{60}$$

where C is a constant and $f(\cdot)$ is a function of τ_{max} and τ_{min}.

Proof Known that

$$\lim_{t \to \infty} P_m^*(t) = [\lim_{t \to \infty} {}^{\nu}p_{ij}^*(t)]^{N-1}, \tag{61}$$

where ${}^{\nu}p_{ij}^*(t)$ is the transition probability for edge $(i, j) \in S_{\min}$ after the first optimal solution $S_{\min}$ is found, determined by

$$\begin{aligned} {}^{\nu}p_{ij}^*(t) &= \frac{1}{F}\left(p_{ij}^*(t) + \sum_{k=1}^{N'-1} |\frac{\Gamma(k-\nu)}{\Gamma(-\nu)\Gamma(k+1)}| p_{in_k}(t)\right) \\ &\geq \frac{1}{F} p_{ij}^*(t), \end{aligned} \tag{62}$$

where F is a constant defined as

$$F = \sum_{k=1}^{N'-1} |\frac{\Gamma(k-\nu)}{\Gamma(-\nu)\Gamma(k+1)}|, \tag{63}$$

and $p_{ij}^*(t)$ satisfies that

$$\lim_{t \to \infty} p_{ij}^*(t) = \frac{\tau_{\max}^{\alpha} \eta_{ij}^{\beta}}{\tau_{\max}^{\alpha} \eta_{ij}^{\beta} + (N-1)\tau_{\min}^{\alpha} \eta_{kl}^{\beta}}. \tag{64}$$

Accordingly, the probability of ant m finding the optimal solution $S_{\min}$ at the tth iteration satisfies

$$\begin{aligned} \lim_{t \to \infty} P_m^*(t) &= [\lim_{t \to \infty} {}^{\nu}p_{ij}^*(t)]^{N-1} \\ &\geq \left(\frac{\lim_{t \to \infty} p_{ij}^*(t)}{F}\right)^{N-1} \\ &\geq \left(\frac{1}{F}\right)^{N-1} \left(\frac{\tau_{\max}^{\alpha} \eta_{ij}^{\beta}}{\tau_{\max}^{\alpha} \eta_{ij}^{\beta} + (N-1)\tau_{\min}^{\alpha} \eta_{kl}^{\beta}}\right)^{N-1} \\ &\geq \left(\frac{1}{F}\right)^{N-1} \left(1 + \frac{(N-1)\tau_{\min}^{\alpha} \eta_{kl}^{\beta}}{\tau_{\max}^{\alpha} \eta_{ij}^{\beta}}\right)^{-(N-1)}. \end{aligned} \tag{65}$$

Based on the binomial expansion theorem, we derive that

$$\lim_{t \to \infty} P_m^*(t) \geq \left(\frac{1}{F}\right)^{N-1} \left(1 - \frac{(N-1)^2 \tau_{\min}^{\alpha} \eta_{kl}^{\beta}}{\tau_{\max}^{\alpha} \eta_{ij}^{\beta}}\right). \tag{66}$$

Clearly, this then results in the consequence, (66), by setting $C = \left(\frac{1}{F}\right)^{N-1}$ and $f(\tau_{\max}, \tau_{\min}) = \frac{(N-1)^2 \tau_{\min}^{\alpha} \eta_{kl}^{\beta}}{\tau_{\max}^{\alpha} \eta_{ij}^{\beta}}$. □

The proofs of the above four propositions, two theorems and one corollary show the convergence properties of SSMFAS. Proposition 1 reveals that the pheromone value on any edge has an upper bound. Proposition 2 proves that after the optimal path has been discovered, the pheromone concentrations on edges of the optimal route increase to the maximum with the increase of iterations. Theorem 1 demonstrates that the probability of finding an optimal path tends to 1 with the increase of iterations. Theorem 2 illustrates that finite times later after the optimal solution is discovered, the value of pheromone on any edge of the optimal route keeps larger than that on any other edge. Proposition 3 indicates that after the optimal path has been discovered, the value of pheromone on the edge that does not belong to the optimal route approaches the minimum value with the increasing number of iterations. Proposition 4 specifically declares that finite times later after the optimal solution is discovered, the value of pheromone on the edge that does not belong to the optimal route maintains the minimum. Corollary 1 exhibits that the probability of finding an optimal solution has a lower bound.

Experiments

This section presents extensive experimental results of the proposed SSMFAS, as well as some advanced peer methods. The experimental setup is given first. Then, the convergence curves of SSMFAS on different TSP instances are

Table 2 Orthogonal array $L_{16}(4^3)$

ParaCom	δ	μ	ν	ParaCom	δ	μ	ν
1	4	3	1	9	3	1	4
2	2	1	3	10	2	2	2
3	4	1	2	11	3	2	1
4	4	4	4	12	2	4	1
5	2	3	4	13	1	2	4
6	3	4	2	14	3	3	3
7	1	1	1	15	4	2	3
8	1	3	2	16	4	3	1

drawn. Subsequently, the effectiveness of each component and strategy is verified by an ablation study. Finally, various comparisons have been conducted with several advanced algorithms on small-scale TSP instances and larger-scale TSP instances to demonstrate the competitive performance of SSMFAS.

Experimental setup

All experiments have been implemented in MATLAB 2019a environment in Window 10 on the PC equipped with an Intel Core i5 processor and 8GB of RAM.

Evaluation metrics

TSP instances obtained from TSPLIB of Heidelberg University[1] are used in the experiments. The number in each instance's name represents the number of cities. Each algorithm is tested 20 runs for one instance with random initialization, and each run includes 300 iterations.

The minimal, maximal and average solutions acquired from 20 runs are recorded as the crucial indicators to evaluate the overall performance of algorithms.

The standard deviation (SD) is adopted in parameter tuning to evaluate the dispersion of the solutions, which is computed as

$$\mathrm{SD} = \sqrt{\frac{1}{R}\sum_{i=1}^{R}(S_i - S_{\mathrm{AVE}})^2}, \tag{67}$$

where R ($R = 20$) is the number of runs, S_i and S_{AVE} are the solution in the ith run and the average solution of 20 runs, respectively. SD is considered as a measure of the robustness of algorithms since it is sensitive to data outliers. The smaller SD indicates that the algorithm is more robust.

Table 3 Results of the OED

ParaCom	δ	μ	ν	Ave
1	1.00	1.05	0.9	428.20
2	0.75	0.30	0.9	427.90
3	0.25	0.30	0.8	427.60
4	0.25	1.05	1.0	428.85
5	0.75	0.80	1.0	428.00
6	0.50	1.05	0.8	428.95
7	1.00	0.30	0.7	428.60
8	1.00	0.80	0.8	426.85
9	0.50	0.30	1.0	429.00
10	0.75	0.55	0.8	427.15
11	0.50	0.55	0.7	428.30
12	0.75	1.05	0.7	427.70
13	1.00	0.55	1.0	429.15
14	0.50	0.80	0.9	427.10
15	**0.25**	**0.55**	**0.9**	**426.50**
16	0.25	0.80	0.7	427.25
z1	428.2	428.28	427.96	–
z2	427.69	427.78	427.64	–
z3	428.34	**427.30**	**427.43**	–
z4	**427.55**	428.43	428.75	–

Table 4 Results of the additional experiments

Instance	δ	μ	ν	Ave
eil51	0.25	0.80	0.9	426.8
	0.25	0.55	0.9	**426.5**
eil76	0.25	0.80	0.9	542.1
	0.25	0.55	0.9	**539.8**
st70	0.25	0.80	0.9	679.45
	0.25	0.55	0.9	**678.05**
rat99	0.25	0.80	0.9	1218.65
	0.25	0.55	0.9	**1214.20**
kroA100	0.25	0.80	0.9	**21305.80**
	0.25	0.55	0.9	21309.75

The relative error (ER) is used to reflect the reliability of the results, which is defined as

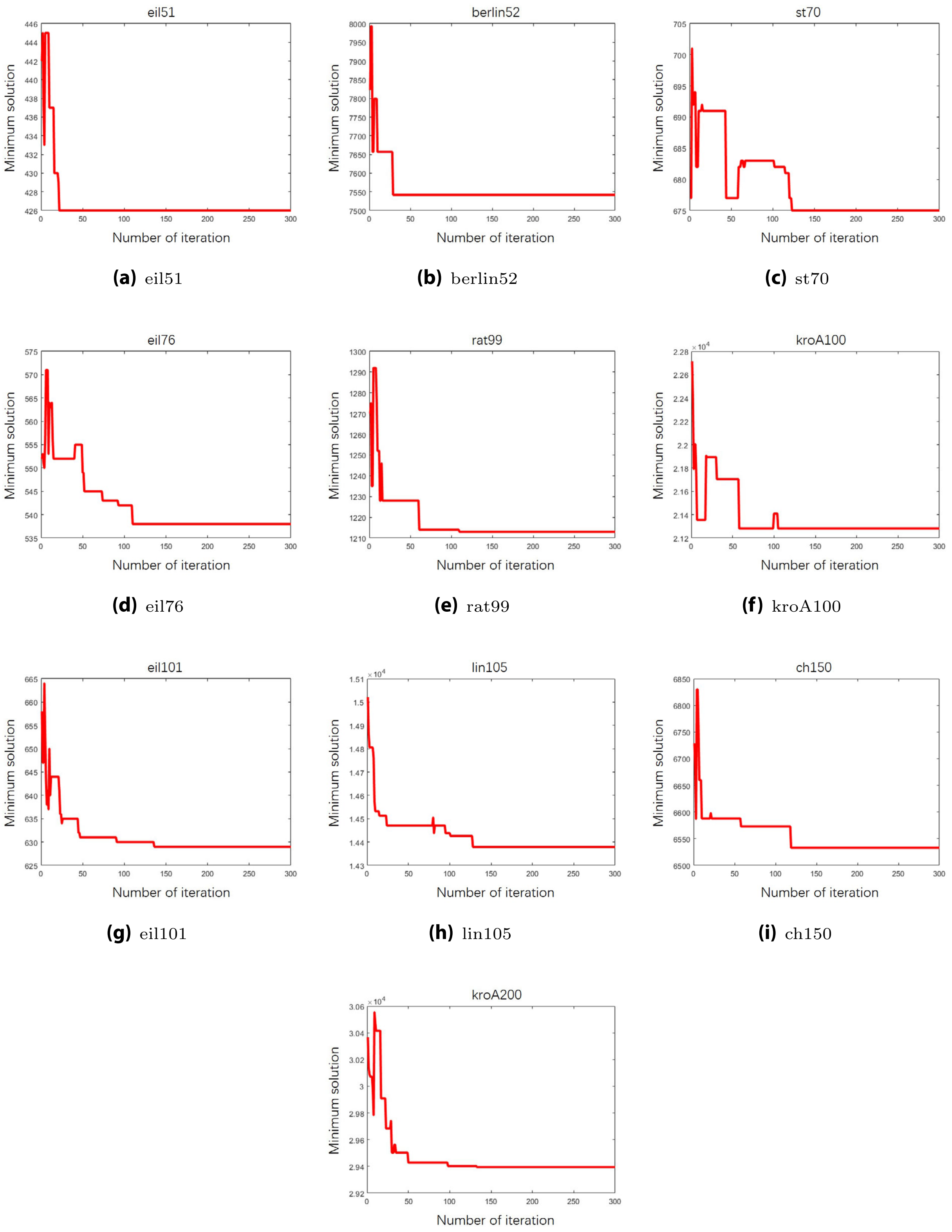
$$\mathrm{ER} = \frac{S_{\mathrm{AVE}} - S_{\mathrm{MIN}}}{S_{\mathrm{MIN}}} \times 100\%, \tag{68}$$

where S_{AVE} and S_{MIN} are the average and minimal solutions of 20 runs, respectively.

Parameter settings

For SSMFAS, α, β and ρ are set to 1, 5 and 0.1, respectively, referring to [41]. N' and M' are set to 8. The upper and lower

[1] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.



(a) eil51 **(b)** berlin52 **(c)** st70

(d) eil76 **(e)** rat99 **(f)** kroA100

(g) eil101 **(h)** lin105 **(i)** ch150

(j) kroA200

Fig. 4 The values of solutions of SSMFAS on ten TSP instances over iterations

bounds of pheromone are computed by $\tau_{\max} = \sum_{m=1}^{M'} \mid \frac{\Gamma(m-\nu-1)}{\Gamma(-\nu)\Gamma(m)} \mid \frac{1}{S^*}$ and $\tau_{\min} = 2\tau_{\max}/N$, respectively, where S^* is the approximate estimated shortest route length.

The fractional order, ν, the coefficient in conductivity update rule, μ, and the weight coefficient in maximum–minimum flux strategy, δ, are searched to get appropriate values, since the performance of SSMFAS is sensitive to them. To determine an appropriate combination of values of these parameters, four alternative values are tested for each parameter, i.e., $\nu \in \{0.7, 0.8, 0.9, 1.0\}$, $\mu \in \{0.3, 0.55, 0.8, 1.05\}$, and $\delta \in \{0.25, 0.50, 0.75, 1.00\}$. Accordingly, a four-level and three-factor orthogonal experimental design (OED) is utilized to choose a good combination of values of the parameters.

Let $L_a(b^c)$ represent the orthogonal array, where a is the number of tests performed, b and c are the level and number of parameters, respectively. Therefore, an orthogonal array $L_{16}(4^3)$ is procured, where 16 typical combinations are shown in Table 2.

One small-sized TSP instance, eil51, is used for this experiment. The average solutions of 16 combinations are listed in Table 3, where the best solutions are marked in bold font. We note that $z_1 - z_4$ represent the average of the results from 20 runs employing some certain parameters with specific values. For example, as $\delta = 1.00$ is obtained by calculating $(428.2 + 428.6 + 426.85 + 429.15)/4 = 428.2$, the result 428.2 is placed in the corresponding position (z_1, δ). From Table 3, it is clear that the best result is obtained when the parameter combination is $\delta = 0.25$, $\mu = 0.8$, and $\nu = 0.9$.

Then, the optimal combination obtained by orthogonality, $\{0.25, 0.80, 0.9\}$, is compared with the parameter combination that obtains the smallest average result, $\{0.25, 0.55, 0.9\}$, on 5 TSP instances. The results are recorded in Table 4 where the best solutions are emphasized in bold font. It is obvious that the combination of $\{0.25, 0.80, 0.9\}$ wins on 4 of 5 instances. Therefore, we set $\delta = 0.25$, $\mu = 0.8$, and $\nu = 0.9$ in later experiments.

Convergence property

The convergence curves of SSMFAS on 10 TSP instances are displayed in Fig. 4. It can be seen that the minimum solutions are reached within 150 iterations for all instances, which verifies the convergence of the algorithm experimentally.

Effectiveness of hybridization

An ablation experiment is designed to prove the effectiveness of the hybridization of AS, SM model and fractional-order difference components. Different versions of SSMFAS is developed for comparison. By setting the fractional order to $\nu = 1$, SSMFAS degrades into an integer-order version, marked as SSMAS. By setting the parameter to $\sigma(t) = 0$, SSMFAS degrades into a version without SM model involved, marked as FAS. By setting both $\nu = 1$ and $\sigma(t) = 0$, SSMFAS degrades into a version without the fractional calculus and SM model, marked as AS.

Table 5 The average solutions of AS, FAS, SSMAS, and SSMFAS on different TSP instances

Instance	BKS	AS	FAS	SSMAS	SSMFAS
eil51	426	430.00	427.85	428.00	**426.50**
berlin52	7542	7556.2	**7542.0**	**7542.0**	**7542.0**
st70	675	679.05	679.10	678.65	**678.05**
eil76	538	546.55	541.50	542.60	**539.45**
rat99	1211	1219.10	1216.40	1215.25	**1214.20**
kroA100	21282	21349.30	21312.75	21315.25	**21309.75**
eil101	629	647.85	637.65	641.50	**635.95**
lin105	14379	14404.10	14398.10	14399.20	**14390.50**
ch150	6528	6570.80	6566.40	6563.55	**6562.90**
kroA200	29368	29558.75	29511.50	29527.20	**29506.30**

The average solutions as well as the best known solutions (BKS) are reported in Table 5, where the best solutions are marked in bold font. SSMFAS achieves the best solutions on all instances compared to other versions, which verifies the success of hybridization in SSMFAS. Furthermore, the results also show that both FAS and ASMAS perform better than AS, while there is no significant difference between them.

On the other hand, Table 6 presents the computation time of AS, FAS, SSMAS, and SSMFAS on different TSP instances. As can be seen from the table, insertions of the fractional calculation and SM model make the calculation time longer. This may be due to the long-term memory character of fractional-order difference is more computationally intensive, along with the fact that parallel computation has not been adopted in SM module. This limitation will be a major issue for future improvement.

Strategy evaluation

Two strategies proposed in the paper, adaptive conductivity strategy and maximum–minimum flux strategy, are evaluated separately to verify the effectiveness. The results are summarized in Table 7, where SSMFAS^0 represents the version without two strategies, SSMFAS^1 represents the version with the adaptive conductivity strategy but without the maximum–minimum flux strategy, and SSMFAS^2 is the opposite version to SSMFAS^1. The best results in each case are emphasized in bold font.

From the data in Table 7, it is apparent that SSMFAS achieves the best performances in terms of all evaluation metrics on the four TSP instances and has significant advan-

Table 6 The computation time (second) of AS, FAS, SSMAS, and SSMFAS on different TSP instances

Instance	AS	FAS	SSMAS	SSMFAS
eil51	13.78	25.14	28.19	37.44
berlin52	15.09	24.63	26.29	35.43
st70	30.63	53.29	57.94	76.19
eil76	38.18	59.81	85.89	92.38
rat99	76.13	126.05	129.92	187.58
kroA100	78.45	126.23	132.87	201.32
eil101	83.25	128.4	136.64	209.86
lin105	90.39	151.62	169.14	225.57
ch150	259.89	443.75	473.6	649.9
kroA200	606.25	1030.02	1073.89	1504.68

tages than the others. It validates that the two specifically designed strategies improve the optimization capability and robustness of SSMFAS significantly.

Second, the respective positive effects of the two strategies are evaluated by comparing SSMFAS^1 and SSMFAS^2 with SSMFAS^0, respectively. SSMFAS^1 is superior to SSMFAS^0 in all TSP instances, which illustrates that the adaptive conductivity strategy has a positive impact on SSMFAS for solving TSPs. The comparison between SSMFAS^2 and SSMFAS^0 draws a similar conclusion.

Therefore, the employment of the two auxiliary strategies is beneficial to improve the performance of SSMFAS.

Comparison to other ACO-based hybrid approaches

To further investigate the efficiency of SSMFAS, several advanced ACO-based hybrid algorithms are used for comparison, including fractional-order ant colony algorithm (FACA) [16], heterogeneous adaptive ACO with 3-Opt local search (HAACO) [41], parallel cooperative hybrid ACO with 3-Opt local search [17], new hybrid PSO and ACO with 3-Opt [18], and two variants of MMAS algorithm [41] which are the standard MMAS incorporated 3-Opt and greedy, and only 3-Opt, respectively.

Table 8 Common parameter settings

Algorithm	α	β	ρ	Maximum Iteration
SSMFAS	1	5	0.1	300
FACA [16]	1	5	0.2	300
HAACO [41]	1	5	0.1	1000
PACO-3Opt [17]	[0,2]	[0,2]	0.1	1000
PSO-ACO-3Opt [18]	[0,2]	[0,2]	0.1	1000
MMAS^1 [41]	1	5	0.1	1000
MMAS^2 [41]	1	5	0.1	1000

We note that like the proposed SSMFAS, the six compared algorithms also include the 3-Opt local search procedure. The results of competitors are taken from the corresponding references. For a fair comparison, we set the values of the common parameters in our algorithm to be the same as those in [41]. Table 8 presents the common parameter settings of the algorithms. Other specific parameters are set following the original setting.

Table 7 The performance of using or not using two strategies separately

Instance	Version	Min	Max	Ave	SD	ER(%)
eil51	SSMFAS^0	**426**	431	428.15	1.27	0.5047
	SSMFAS^1	**426**	430	427.60	1.04	0.3873
	SSMFAS^2	**426**	429	427.70	1.08	0.3991
	SSMFAS	**426**	**428**	**426.50**	**0.76**	**0.1174**
eil76	SSMFAS^0	**538**	548	542.05	3.19	0.7528
	SSMFAS^1	**538**	548	540.45	2.30	0.2695
	SSMFAS^2	**538**	545	541.30	2.25	0.6134
	SSMFAS	**538**	**544**	**539.45**	**1.67**	**0.2695**
st70	SSMFAS^0	676	691	681.90	4.43	0.8728
	SSMFAS^1	**675**	685	680.00	3.39	0.7407
	SSMFAS^2	676	684	680.05	2.65	0.5991
	SSMFAS	**675**	**682**	**678.05**	**2.19**	**0.4519**
kroA100	SSMFAS^0	**21282**	21569	21371.45	91.17	0.4203
	SSMFAS^1	**21282**	21480	21346.25	67.50	0.3019
	SSMFAS^2	**21282**	**21379**	21315.70	40.76	0.1583
	SSMFAS	**21282**	**21379**	**21309.75**	**38.77**	**0.1304**

Table 9 The minimal solutions and the corresponding ranks of different methods on 10 small-scale TSP instances

Instance	BKS	SSMFAS	FACA [16]	HAACO [41]	PACO-3Opt [17]	PSO-ACO-3Opt [18]	MMAS^1 [41]	MMAS^2 [41]
eil51	426	**426** (3.5)	**426** (3.5)	**426** (3.5)	**426** (3.5)	**426** (3.5)	**427** (7)	**426** (3.5)
berlin52	7542	**7542** (4)	**7542** (4)	**7542** (4)	**7542** (4)	**7542** (4)	**7542** (4)	**7542** (4)
st70	675	**675** (2.5)	**675** (2.5)	**675** (2.5)	**676** (5.5)	**676** (5.5)	**675** (2.5)	682 (7)
eil76	538	**538** (4)	**538** (4)	**538** (4)	**538** (4)	**538** (4)	**538** (4)	**538** (4)
rat99	1211	**1211** (2)	**1211** (2)	**1211** (2)	1213 (6)	1224 (7)	**1212** (4.5)	**1212** (4.5)
kroA100	21282	**21282** (2)	21379 (6.5)	**21282** (2)	**21282** (2)	21301 (4)	21315 (5)	21379 (6.5)
eil101	629	**629** (2)	**629** (2)	**630** (4)	**629** (2)	631 (6)	631 (6)	631 (6)
lin105	14379	**14379** (4)	**14379** (4)	**14379** (4)	**14379** (4)	**14379** (4)	**14379** (4)	**14379** (4)
ch150	6528	**6528** (1.5)	**6528** (1.5)	6566 (5.5)	6570 (7)	**6538** (3)	6554 (4)	6566 (5.5)
kroA200	29368	**29380** (1)	29464 (2.5)	29483 (4)	29533 (7)	29464 (2.5)	29485 (5)	29488 (6)
Average rank		**2.65**	**3.25**	3.55	4.5	4.35	4.6	5.1

Performance evaluation on small-scale problems

First, SSMFAS are evaluated on 10 Small-scale TSP instances. Tables 9 and 10 present the minimal and average solutions of SSMFAS and other algorithms, respectively. For the purpose of analyzing the performance of the algorithms, the ranks of the solutions are computed and listed in the parentheses in those tables. We note that if any values are tied, we compute their average rank. Furthermore, the average ranks of each algorithm on the 10 instances are also computed and reported in the last lines of tables. The best solutions in each case are written in bold and highlighted in gray, and the second-best results are also written in bold font.

As shown in Table 9, the minimal solutions of SSMFAS on all TSP instances are the shortest over seven algorithms. As a result, SSMFAS obtains the smallest average rank of 2.65, and FACA with an average rank of 3.25 comes next. And, perhaps more tellingly, in terms of the average solutions, Table 10 indicates that SSMFAS outperforms other algorithms with an average rank of 2.20, followed by PACO-3Opt with 3.15.

Now we check whether the solutions constructed by these algorithms are significantly different. A common statistical method, Analysis of Variance (ANOVA) [42], is utilized to test if there is significant differences in these algorithms. The null-hypothesis is tested and the generated p-value is $9.95e-8$, which is less than the significance level of 0.1. Thus, the null-hypothesis is rejected. The Wilcoxon signed-ranks test [43] is then used to verify the significant improvement of the proposed SSMFAS in pairs. The results are reported in Table 11. As can be seen, the generated p-values are all less than the significance level of 0.1 except for FACA. This may due to the fact that the two algorithms are tied on two instances. More specifically, Fig. 5 shows the shape of distribution of ranks in terms of average solution. The distribution of SSMFAS is more concentrated with the smallest median, which clarifies the effectiveness and robustness of the proposed SSMFAS.

Performance evaluation on larger-scale problems

Furthermore, to illustrate the ability of the proposed SSMFAS to handle large-scale TSPs, 11 lager-scale TSP instances from TSPLIB whose numbers of cities are between 400 and 800 are used in this subsection.

Tables 12 and 13 present the minimal and average solutions of SSMFAS and its competitors, respectively. The best solutions in each case is highlighted in bold. It can be seen from the data in Table 12 that SSMFAS wins on 8 of 11 instances. From the average solutions in Table 13 we can see

Table 10 The average solutions and the corresponding ranks of different methods on 10 small-scale TSP instances

Instance	BKS	SSMFAS	FACA [16]	HAACO [41]	PACO-3Opt [17]	PSO-ACO-3Opt [18]	MMAS^1 [41]	MMAS^2 [41]
eil51	426	426.5 (3)	427.4 (4)	427.5 (5)	**426.35** **(1)**	**426.45** **(2)**	429.4 (7)	428.5 (6)
berlin52	7542	**7542** **(3.5)**	**7542** **(3.5)**	**7542** **(3.5)**	**7542** **(3.5)**	7543.2 7	**7542** **(3.5)**	**7542** **(3.5)**
st70	675	**676.65** **(2)**	680.1 (5)	**676.5** **(1)**	677.85 (3)	678.2 (4)	683.8 (6)	685.2 (7)
eil76	538	**539.45** **(2)**	541.0 (4)	542 (5)	539.85 (3)	**538.3** **(1)**	542.8 (6)	543.5 (7)
rat99	1211	**1213.0** **(1.5)**	**1213.0** **(1.5)**	1214.1 (3)	1217.1 (5)	1227.4 (7)	1216.9 (4)	1219.4 (6)
kroA100	21282	**21309.75** **(1)**	21379.0 (4)	21364.2 (3)	**21326.8** **(2)**	21445.1 (5)	21528.3 (7)	21513.7 (6)
eil101	629	635.9 (5)	**630.6** **(2)**	632.5 (3)	**630.55** **(1)**	632.7 (4)	640.4 (6)	640.9 (7)
lin105	14379	**14379** **(1)**	14392.4 (3)	14411.8 (5)	14393 (4)	**14379.15** **(2)**	14429.2 (6)	14433 (7)
ch150	6528	**6560.5** **(2)**	**6537.0** **(1)**	6578.8 (4)	6601.4 (6)	6563.95 (3)	6603.9 (7)	6581 (5)
kroA200	29368	**29506.3** **(1)**	29680.5 (5)	**29633.2** **(2)**	29644.05 (3)	29646.05 (4)	29799.9 (7)	29760.3 (6)
Average rank		**2.20**	3.30	3.45	**3.15**	3.90	5.95	6.05

Table 11 The p-values of the Wilcoxon signed-ranks test

	FACA [16]	HAACO [41]	PACO-3Opt [17]	PSO-ACO-3Opt [18]	MMAS^1 [41]	MMAS^2 [41]
p-value	0.3125	0.0547	0.0547	0.0840	0.0039	0.0039
win(+)/tie(≈)	(≈)	(+)	(+)	(+)	(+)	(+)

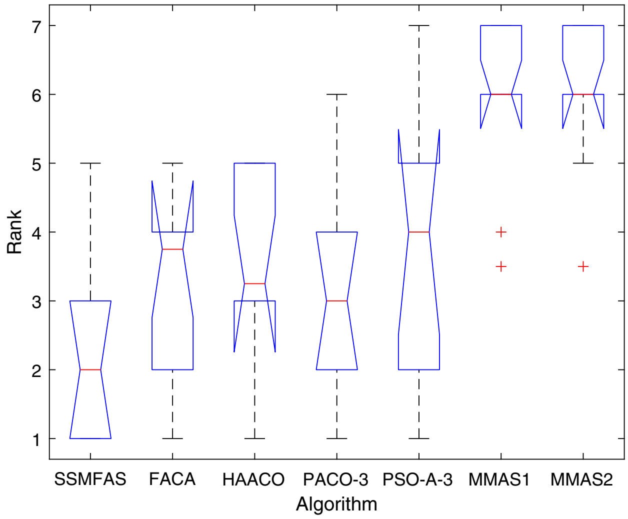


Fig. 5 Box-plot of the ranks of average solutions of 7 algorithms

that SSMFAS obtains the best results on 9 of 11 instances. The results demonstrate the effectiveness of SSMFAS for solving larger-scale TSPs.

It is worth mentioning that the superiority of the proposed SSMFAS becomes more obvious compared with the competitors when the problem size becomes larger. This may illustrate the proposed SSMFAS has good robustness and adaptability to large-scale problems.

Comparison to other heuristic approaches

In addition to ACO-based hybrid methods, some advanced heuristic methods are involved in the comparison, including a discrete stochastic population-based optimization algorithm (DJAYA) [44], an approach of improvement heuristics based on 2-opt operators by deep reinforcement learning [45], and an open source software for combinatorial optimization, Google OR-Tools [46], which includes 2-opt and LKH

Table 12 The minimal solutions of different methods on 11 larger-scale TSP instances

Instance	BKS	SSMFAS	HAACO [41]	PACO-3Opt [17]	PSO-ACO-3Opt [18]
rd400	15281	**15511**	15603	15578	15594
fl417	11861	**11933**	11960	11972	11947
pr439	107217	**108481**	108730	108482	108530
pcb442	50778	51976	51780	**51962**	52131
d493	35002	35918	–	**35735**	35789
u574	36905	38094	–	37981	**37818**
rat575	6773	**6871**	–	7003	6987
p654	34643	**34756**	–	35045	35052
d657	48912	**49561**	–	50206	50291
u724	41910	**42565**	–	42764	43172
rat783	8806	**8939**	–	9111	9128

HAACO lacks results on some instances because it was only performed on the first four TSP instances in [41]

Table 13 The average solutions of different methods on 11 larger-scale TSP instances

Instance	BKS	SSMFAS	HAACO [41]	PACO-3Opt [17]	PSO-ACO-3Opt [18]
rd400	15281	**15530.5**	15644.2	15613.9	15691.3
fl417	11861	**11947.65**	11979.5	11987.4	11980.4
pr439	107217	**108535**	108950.6	108702	108965.4
pcb442	50778	**52171**	52179.8	52202.4	52368.1
d493	35002	35971.4	–	**35841**	35973.8
u574	36905	38195.2	–	**38030.7**	38112.9
rat575	6773	**6923**	–	7012.4	7018.6
p654	34643	**34779.1**	–	35075	35098.2
d657	48912	**49587.8**	–	50277.5	50475.5
u724	41910	**42580.67**	–	43122.5	43300.3
rat783	8806	**8950**	–	9127.3	9138.1

HAACO lacks results on some instances because it was only performed on the first four TSP instances in [41]

Table 14 Performance of SSMFAS and other heuristic approaches on TSP instances

Instance	BKS	SSMFAS	DJAYA [44]	[45]	OR-Tools [45]
eil51	426	**426.5**	440.15	427	439
berlin52	7542	**7542**	7580.3	7974	7944
st70	675	**676.65**	702.3	680	683
eil76	538	**539.45**	573.17	552	548
rat99	1211	**1213**	–	1388	1284
kroA100	21282	**21309.75**	21735.31	23751	21960
eil101	629	635.9	677.37	**635**	650
lin105	14379	**14379**	–	16156	15363
ch150	6528	**6560.5**	6638.63	6597	6733
kroA200	29368	**29506.3**	–	32522	29874

(Lin–Kernighan–Helsgaun) [47] as improvement heuristics [45,48]. The results are reported on Table 14, where the best result in each instance is highlighted in bold. We note that the results of SSMFAS are the average solutions of 20 runs, and the results of competitors are taken directly from the corresponding references.

As can be seen from Table 14, SSMFAS outperforms other approaches on 9 of 10 instances. It further verifies the superiority of the proposed SSMFAS in addressing TSPs.

Conclusion

In this paper, we proposed a novel algorithm called SSMFAS combining a state adaptive multi-entrance-exit SM model and a fractional-order based AS algorithm for solving TSPs.

The adaptive conductivity strategy is developed to shift the state of SM in different periods. The maximum–minimum flux strategy limits the upper and lower bounds of the flux to match with the concentration of pheromone of AS algorithm. Fractional-order neighborhood transition probability which uses the neighbor information is introduced for path construction by ants to improve the performance. Varying degrees of flux in SM model are added as pheromone in the pheromone update process to provide more information.

The convergence properties of the SSMFAS have been verified. A multitude of experiments are carried out on TSP instances. First, the effect of the hybridization of AS, SM, and fractional-order calculus is verified. Then, the effectiveness of two auxiliary strategies is demonstrated. At last, comparisons with several state-of-the-art algorithms illustrate the competitive performance of SSMFAS.

In the future work, we plan to optimize the combination of SM model and AS algorithm, and improve the computation cost through some parallel strategies, to address some dynamic multi-objective TSPs. More broadly, further research could also be conducted to explore the potential of combining other path planning methods and metaheuristic algorithms.

Acknowledgements This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFA0708700; in part by the National Natural Science Foundation of China under Grant 62173345; in part by the Major Scientific and Technological Projects of China National Petroleum Corporation (CNPC) under Grant ZD2019-183-008; in part by the Fundamental Research Funds for the Central Universities under Grant 22CX03002A; and in part by the Source Innovation Scientific and Incubation Project of Qingdao, China under Grant 2020-88.

Declarations

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

References

1. Karp RM (1972) Reducibility among combinatorial problems. Springer, Boston, MA, pp 85–103. https://doi.org/10.1007/978-1-4684-2001-2_9
2. Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman & Co., USA
3. Punnen AP (2007) The traveling salesman problem: applications, formulations and variations. Springer, Boston, MA, pp 1–28. https://doi.org/10.1007/0-306-48213-4_1
4. Dorigo M (1992) Optimization, learning and natural algorithms. PhD thesis, Politecnico Di Milano, Italy
5. Dorigo M, Stützle T (2003) The ant colony optimization metaheuristic: algorithms, applications, and advances. Springer, Boston, MA, pp 250–285. https://doi.org/10.1007/0-306-48056-5_9
6. Wang K-P, Huang L, Zhou C-G, Pang W (2003) Particle swarm optimization for traveling salesman problem. In: Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No.03EX693), vol. 3, pp. 1583–15853. https://doi.org/10.1109/ICMLC.2003.1259748
7. Marinakis Y, Marinaki M (2010) A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem. Comput Oper Res 37(3):432–442. https://doi.org/10.1016/j.cor.2009.03.004
8. Zhong Y, Lin J, Wang L, Zhang H (2017) Hybrid discrete artificial bee colony algorithm with threshold acceptance criterion for traveling salesman problem. Inf Sci 421:70–84. https://doi.org/10.1016/j.ins.2017.08.067
9. Pandit D, Zhang L, Chattopadhyay S, Lim CP, Liu C (2018) A scattering and repulsive swarm intelligence algorithm for solving global optimization problems. Knowl-Based Syst 156:12–42. https://doi.org/10.1016/j.knosys.2018.05.002
10. Kanna SKR, Sivakumar K, Lingaraj N (2021) Development of deer hunting linked earthworm optimization algorithm for solving large scale traveling salesman problem. Knowl-Based Syst 227:107199. https://doi.org/10.1016/j.knosys.2021.107199
11. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press, USA
12. Stutzle T, Hoos H (1997) Max-min ant system and local search for the traveling salesman problem. In: Proceedings of 1997 IEEE international conference on evolutionary computation (ICEC '97), pp. 309–314. https://doi.org/10.1109/ICEC.1997.592327
13. Bullnheimer B, Hartl RF, Strauß C (1997) A new rank based version of the ant system. a computational study. Working Papers SFB "Adaptive information systems and modelling in economics and management science" 1, SFB Adaptive Information Systems and Modelling in Economics and Management Science, WU Vienna University of Economics and Business, Vienna. https://epub.wu.ac.at/616/
14. Stutzle T, Dorigo M (2002) A short convergence proof for a class of ant colony optimization algorithms. IEEE Trans Evol Comput 6(4):358–365. https://doi.org/10.1109/TEVC.2002.802444
15. Gong X, Rong Z, Gao T, Pu Y, Wang J (2019) An improved ant colony optimization algorithm based on fractional order memory for traveling salesman problems. In: 2019 IEEE symposium series on computational intelligence (SSCI), pp. 1516–1522. https://doi.org/10.1109/SSCI44817.2019.9003009
16. Pu Y-F, Siarry P, Zhu W-Y, Wang J, Zhang N (2022) Fractional-order ant colony algorithm: a fractional long term memory based cooperative learning approach. Swarm Evol Comput 69:101014. https://doi.org/10.1016/j.swevo.2021.101014
17. Gulcu S, Mahi M, Baykan O, Kodaz H (2018) A parallel cooperative hybrid method based on ant colony optimization and 3-Opt

algorithm for solving traveling salesman problem. Soft Comput 22:1669–1685. https://doi.org/10.1007/s00500-016-2432-3
18. Mahi M, Ömer Kaan Baykan Kodaz H (2015) A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem. Appl Soft Comput 30:484–490. https://doi.org/10.1016/j.asoc.2015.01.068
19. Nakagaki T, Iima M, Ueda T, Nishiura Y, Saigusa T, Tero A, Kobayashi R, Showalter K (2007) Minimum-risk path finding by an adaptive amoebal network. Phys Rev Lett 99:068104. https://doi.org/10.1103/PhysRevLett.99.068104
20. Adamatzky A, Martinez GJ (2013) Bio-imitation of Mexican migration routes to the USA with slime mould on 3D terrains. J Bionic Eng 10(2):242–250. https://doi.org/10.1016/S1672-6529(13)60220-6
21. Nakagaki T, Yamada H, Toth A (2000) Maze-solving by an amoeboid organism. Nature 407:470–470. https://doi.org/10.1038/35035159
22. Tero A, Takagi S, Saigusa T, Ito K, Bebber DP, Fricker MD, Yumiki K, Kobayashi R, Nakagaki T (2010) Rules for biologically inspired adaptive network design. Science 327(5964):439–442. https://doi.org/10.1126/science.1177894
23. Adamatzky A, Martinez GJ, Chapa-Vergara SV, Asomoza-Palacio R, Stephens CR (2011) Approximating Mexican highways with slime mould. Nat Comput 10:1195–1214. https://doi.org/10.1007/s11047-011-9255-z
24. Tsompanas M-AI, Sirakoulis GC, Adamatzky AI (2016) Physarum in silicon: the Greek motorways study. Nat Comput 15:279–295. https://doi.org/10.1007/s11047-014-9459-0
25. Tero A, Kobayashi R, Nakagaki T (2007) A mathematical model for adaptive transport network in path finding by true slime mold. J Theor Biol 244(4):553–564. https://doi.org/10.1016/j.jtbi.2006.07.015
26. Zhang X, Huang S, Hu Y, Zhang Y, Mahadevan S, Deng Y (2013) Solving 0–1 knapsack problems based on amoeboid organism algorithm. Appl Math Comput 219(19):9959–9970. https://doi.org/10.1016/j.amc.2013.04.023
27. Gao C, Zhang X, Yue Z, Wei D (2020) An accelerated physarum solver for network optimization. IEEE Trans Cybern 50:2168–2267. https://doi.org/10.1109/TCYB.2018.2872808
28. Xu S, Jiang W, Deng X, Shou Y (2018) A modified physarum-inspired model for the user equilibrium traffic assignment problem. Appl Math Model 55:340–353. https://doi.org/10.1016/j.apm.2017.07.032
29. Gao C, Chen S, Li X, Huang J, Zhang Z (2017) A physarum-inspired optimization algorithm for load-shedding problem. Appl Soft Comput 61:239–255. https://doi.org/10.1016/j.asoc.2017.07.043
30. Zhang X, Gao C, Deng Y, Zhang Z (2016) Slime mould inspired applications on graph-optimization problems. Springer, Cham, pp 519–562. https://doi.org/10.1007/978-3-319-26662-6_26
31. Zhang X, Chan FTS, Adamatzky A, Mahadevan S, Yang H, Zhang Z, Deng Y (2017) An intelligent physarum solver for supply chain network design under profit maximization and oligopolistic competition. Int J Prod Res 55(1):244–263. https://doi.org/10.1080/00207543.2016.1203075
32. Liu M, Li Y, Li A, Huo Q, Zhang N, Qu N, Zhu M, Chen L (2020) A slime mold-ant colony fusion algorithm for solving traveling salesman problem. IEEE Access 8:202508–202521. https://doi.org/10.1109/ACCESS.2020.3035584
33. Cai J, Perfect E, Cheng C-L, Hu X (2014) Generalized modeling of spontaneous imbibition based on Hagen-Poiseuille flow in tortuous capillaries with variably shaped apertures. Langmuir 30(18):5142–5151. https://doi.org/10.1021/la5007204
34. Tanyi EK, Burton BT, Narimanov EE, Noginov MA (2016) Thermal radiation of er doped crystals: studying the range of applicability of the kirchhoff's law. In: Conference on Lasers and Electro-Optics, pp. 2–30. Optica Publishing Group, USA. http://opg.optica.org/abstract.cfm?URI=CLEO_SI-2016-JW2A.30
35. Zhang H, Pu Y-F, Xie X, Zhang B, Wang J, Huang T (2021) A global neural network learning machine: coupled integer and fractional calculus operator with an adaptive learning scheme. Neural Netw 143:386–399. https://doi.org/10.1016/j.neunet.2021.06.021
36. Oldham KB, Spanier J (eds) (1974) The fractional calculus theory and applications of differentiation and integration to arbitrary order. Academic Press, USA
37. Samko SG, Kilbas A, Marichev O (eds) (1993) Fractional Integrals and Derivatives: Theory and Applications. Gordon and Breach Science Publishers, USA
38. Podlubny I (ed) (1998) Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications. Academic Press, USA
39. Agrawal OP (2007) Fractional variational calculus in terms of Riesz fractional derivatives. J Phys Math Theor 40(24):6287–6303
40. Qian T, Zhang Z, Gao C, Wu Y, Liu Y (2013) An ant colony system based on the physarum network. In: Tan Y, Shi Y, Mo H (eds) Advances in swarm intelligence. Springer, Berlin, Heidelberg, pp 297–305
41. Tuani AF, Keedwell E, Collett M (2020) Heterogenous adaptive ant colony optimization with 3-opt local search for the travelling salesman problem. Appl Soft Comput 97(part B):106720. https://doi.org/10.1016/j.asoc.2020.106720
42. Fisher RA (1956) Statistical methods and scientific inference. Oliver and Boyd, London
43. Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1(6):80–83. https://doi.org/10.2307/3001968
44. Gunduz M, Aslan M (2021) Djaya: a discrete jaya algorithm for solving traveling salesman problem. Appl Soft Comput 105:107275. https://doi.org/10.1016/j.asoc.2021.107275
45. da Costa PRdO, Rhuggenaath J, Zhang Y, Akcay A (2020) Learning 2-opt heuristics for the traveling salesman problem via deep reinforcement learning. In: Pan SJ, Sugiyama M (eds.) Proceedings of The 12th Asian Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 129, pp. 465–480. PMLR. https://proceedings.mlr.press/v129/costa20a.html
46. Perron L, Furnon V OR-Tools. https://developers.google.com/optimization/
47. Helsgaun K (2000) An effective implementation of the Lin-Kernighan traveling salesman heuristic. Eur J Oper Res 126(1):106–130. https://doi.org/10.1016/S0377-2217(99)00284-2
48. Bello I, Pham H, Le QV, Norouzi M, Bengio S (2017) Neural combinatorial optimization with reinforcement learning. arXiv:1611.09940