

A Fractional-Order Weighted and Self-Adaptive Max-Min Ant System with 3-Opt Algorithm for Traveling Salesman Problem

Shushu Chen^{*}, Yifei Pu

College of Computer Science, Sichuan University, Chengdu, China

Email address:

css_uk@stu.scu.edu.cn (Shushu Chen)

^{*}Corresponding author

To cite this article:

Shushu Chen, Yifei Pu. A Fractional-Order Weighted and Self-Adaptive Max-Min Ant System with 3-Opt Algorithm for Traveling Salesman Problem. *International Journal of Intelligent Information Systems*. Vol. 5, No. 4, 2016, pp. 48-54. doi: 10.11648/j.ijis.20160504.11

Received: May 8, 2016; **Accepted:** May 31, 2016; **Published:** June 21, 2016

Abstract: In this paper, a Fractional-order weighted and Self-adaptive Max-Min Ant System (FS-MMAS) is proposed to make full use of spatial information of the traveling salesman problem. Furthermore, it is realized so that ants can select next city according to the complex topography. Most advanced algorithms based on Ant Colony Optimization can't take advantage of spatial information during the traverse tour, which easily leads to local minimums. Through the multi-scale and self-adaptive search, ants can take advantage of information which contributes to finding the global optimization. Finally, the 3-Opt algorithm is used to improve local solutions. The performance of proposed method was investigated on eight different benchmark problems taken from a literature and proved to be better than other well-known methods in terms of solution quality and robustness.

Keywords: Self-Adaptive, Max-Min Ant System, 3-Opt Algorithm, Fractional-Order, Tsp

1. Introduction

The Traveling Salesman Problem (TSP) is a well-known binomial optimization Problem, which has been widely used in many engineering applications such as the design of hardware devices and radio electronic systems, and computer networks etc [1, 2]. However TSP is regarded as a classical NP-hard problem, whose solution will rapidly expand with the increasing scale of issues. Many algorithms have been proposed in order to solve TSP effectively, such as Simulated Annealing (SA) [3], Genetic Algorithms (GA) [4], Ant Colony Optimization (ACO) [5], Particle Swarm Optimization algorithm (PSO) [6] etc. ACO is an optimization algorithm of swarm intelligence which is inspired by the behavior of ants in the real wild, and was first proposed by Dorigo. M and Gambardella. L. M in 1996 [5]. ACO has rapid convergence, strong robustness and distributed parallel computing architecture. It can also be easily combined with other algorithms. Therefore, ACO spreads rapidly and is widely applied. But the performance of ACO is poor when it comes to solving large scale optimization issues. ACO will have low solution quality,

slow convergence and easily fall into local minimums with the increasing scale of issues.

In order to solve those issues, many algorithms based on ACO were proposed. The Ant Colony System (ACS) [7] differs mainly in its pheromone update function. ACS employs a local pheromone update at the end of every iteration. And only the best ant can update its pheromone trials. ACS has better performance than ACO for TSP. The rank-based Ant System (AS rank) [8] incorporates the concept of ranking into the pheromone update procedure. Those M ants are ranked according to their tour length ($L_1 \leq L_2 \leq \dots \leq L_M$). The pheromone trials are updated among several better ants. And they are weighted so that an increment of the pheromone trials is in direct proportion to their ranking. In addition, the paths traversed by the global-best solution receive an additional amount of pheromone. The Max-Min Ant System (MMAS) algorithm [9] differs from the ACO in two ways: only the best ant can update the pheromone trials in the MMAS, but all ants can do in the ACO; the pheromone trials are limited between the upper and the lower pheromone bounds in the MMAS, but there is no limit in the ACO. MMAS makes full use of

historical information and avoids premature convergence to local minimums. Although these methods improve some aspects, they can't make full use of information during the traverse tour and have poor performance in finding the global optimization. As we know, fractional calculus has nonlinear, non-causal and non-stationary characteristics, which help search multi-scalely and contribute to obtaining the global optimization.

In this paper, a Fractional-order weighted and Self-adaptive method based on the Max-Min Ant System is proposed. Not just the best ant can update its pheromone trails, but also those better ants can do. Furthermore, the search processing is self-adaptive such that ants may select other closer cities in addition to the closest city. Hence, FS-MMAS is more likely to find the global optimization. Finally, the 3-Opt algorithm is used to reduce the probability of falling into local minimums. Better results are achieved with the suggested method compared to other studies in the literature.

The rest of the paper is organized as follows: In Section 2, the background knowledge, including ACO, fractional differential approach and 3-Opt algorithms, is given. The proposed method is presented in Section 3. The results obtained in the application are given in Section 4. Finally, in Section 5, we conclude the paper with a summarization of results and emphasize the significance of this study.

2. Materials and Methods

In this section, information about MMAS, fractional differential approach and 3-Opt algorithm is provided.

2.1. Max-Min Ant System

The Max-Min Ant System (MMAS) algorithm was developed by Stutzle, T etc [9]. The algorithm that suggests a solution to the TSP, which is a discrete test issue, by utilizing this attribute of ants, was proposed by Dorigo, M etc [10]. For a set of N towns, the goal of TSP is to find the best route which is shortest and covers each town once. So, it is a closed curve and the starting town is also the ending town. The distance of two towns in TSP is calculated by Euclidean distance, as follows:

$$d_{i,j} = \left[(x_i - x_j)^2 + (y_i - y_j)^2 \right]^{1/2} \quad (1)$$

In Eq. (1), i and j represents the last town and the next town, respectively. $d_{i,j}$ represents the distance from i to j . During one iteration, ants choose the route from town i to town j according to the probability formula as follows:

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}, & \text{if } j \text{ is allowed city} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

In Eq. (2), P_{ij}^k represents the probability of the next town j chosen. $\tau_{ij}(t)$ indicates the amount of pheromones between town i and j . η_{ij} indicates information pertaining to distance

between town i and j whose value is defined as the quantity $1/d_{ij}$. Parameters α and β are used to determine the significance between the amount of pheromones and the inter-city distance. Each ant complete a tour until reaching the end town. Each iteration is defined as the interval in $(t, t + 1)$ where each of the M ants moves once. After one iteration, each of the M ants has left the pheromone trail on a tour where they have gone. So the pheromone trail τ_{ij} will be updated according to the following formula:

$$\tau_{ij}(t + 1) = [(1 - \rho)\tau_{ij}(t) + \sum_1^M \nabla \tau_{ij}^{best}] \tau_{min}^{\tau_{max}} \quad (3)$$

In Eq. (3), ρ is the coefficient of evaporation and receives a value at the interval of $[0-1]$, $\nabla \tau_{ij}^{best}$ indicates the variation of the pheromone trail τ_{ij} and only the shortest tour of the M ants can update its pheromone trail. τ_{max} and τ_{min} represent the upper and lower pheromone bounds respectively. The pheromone trail τ_{ij} is limit between τ_{max} and τ_{min} as follows:

$$\tau_{ij} = \begin{cases} \tau_{min}, & \tau_{ij} < \tau_{min} \\ \tau_{max}, & \tau_{ij} > \tau_{max} \\ \tau_{ij}, & \text{otherwise} \end{cases} \quad (4)$$

2.2. The Basic Definition of Fractional Calculus

There are different definitions for fractional calculus in terms of different angles to analyze the application issues. So far, there isn't a uniform expression of fractional calculus. Among those expressions, there are three classic fractional calculus definitions, which are the definitions from Grünwald-Letnikov, Riemann-Liouville and Caputo. The Grünwald-Letnikov definition of fractional calculus is as follows:

$${}_a^C S_t^v = \lim_{h \rightarrow 0} h^v \sum_{j=0}^{\frac{t-a}{h}} \frac{\Gamma(v+j)}{j! \Gamma(v)} s(t - jh) \quad v \in \mathbb{R}^-, \quad (5)$$

where $\forall v \in \mathbb{R}$, suppose the integral part is $[v]$. when $s(t) \in [a, t]$ ($a < t, a \in \mathbb{R}, t \in \mathbb{R}$), it has $n+1$ order continuous derivative, and the fractional order Grünwald-Letnikov definition can be express as follows [11]:

$${}_a^C D_t^v s(t) = \lim_{h \rightarrow 0} s_h^{(v)}(t) = \lim_{nh=t-a} h^{-v} \sum_{r=0}^n \binom{-v}{r} s(t - rh), \quad (6)$$

where

$$\binom{-v}{r} = \frac{(-v)(-v+1)\dots(-v+r-1)}{r!} \quad (7)$$

2.3. 3-Opt Algorithm

In optimization, 3-Opt is a simple local search algorithm for solving the travelling salesman problem and related network optimization problems [12]. And 3-Opt algorithm is the best one among k-opt algorithms. For a tour route, 3-Opt algorithm selects three different sides from it, and divides the route into three sections. If the original one isn't the shortest route, it will be replaced by the shortest one. Repeat until all possible combinations are traversed.

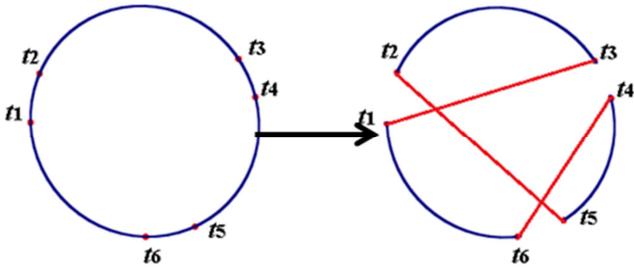


Fig. 1. 3-Opt algorithm representation.

1. Initialize a tour route as (a_1, a_2, \dots, a_n) .
2. Then select three random cities as t_1, t_3, t_5 in fig. 1; $t_1 \in (a_1, a_2, \dots, a_{n-2})$, $t_3 \in (a_2, a_3, \dots, a_{n-1})$, $t_5 \in (a_1, a_2, \dots, a_n)$, and $t_2 = t_1 + 1$, $t_4 = t_3 + 1$, $t_6 = t_5 + 1$, Break edges $\{(t_1, t_2), (t_3, t_4), (t_5, t_6)\}$ then use 3-Opt algorithm to improve the tour (a_1, a_2, \dots, a_n) .
3. If $d(t_1, t_3) + d(t_2, t_5) + d(t_4, t_6) < d(t_1, t_2) + d(t_3, t_4) + d(t_5, t_6)$, a new tour route $(t_1, t_3, \dots, t_2, t_5, \dots, t_4, t_6)$ replaces the original tour route as (a_1, a_2, \dots, a_n) .
4. Repeat 2 and 3 until all possible combinations are traversed.

3. Proposed Method and Analysis (FS-MMAS)

In general, ants can select the next city by the probability formula. If ants select a wrong city, the pheromone trials of the wrong route will be updated and increased. As a result, the algorithm falls into local minimums which is not what we expect. In this paper, a self-adaptive method is proposed. Ants select not only the best city but also other probable cities. The probability formula is changed as follows:

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}, & \text{if } j \text{ is allowed city} \\ S, & \text{if } S \text{ is allowed city} \end{cases} \quad (8)$$

In Eq. (8), the probability formula is improved based on the MMAS algorithm. Firstly, each ant of M ants select the next town j which has the highest ratio of P_{ij}^k . Then it will choose other towns S which is defined as follows:

$$S \in \{x | d_{ix} \leq \lambda * d_{ij}, \text{ if } x \in \text{allowed}\} \quad (9)$$

In Eq. (9), λ is the coefficient and receives a value at the interval of [1-2]. As a result, a few of tours may appear with the varied topography. In this way, the new method can be self-adaptive. Ants may select other closer cities, which contributes to the probability of finding the global optimization. In the MMAS algorithm, only the best ant can update its pheromone trials of the traverse route which it goes through. But MMAS can't take advantage of spatial information after one iteration. In this paper, a fractional-order weighted method is proposed. All the traverse routes of the ants are sorted according to their length, such as $(L_1 \leq L_2 \leq \dots \leq L_M)$. And a few of them which have shorter tour length update their pheromone trials as follows:

$$\tau_{ij}(t+1) = [(1-\rho)\tau_{ij}(t) + \sum_{k=1}^r w(k) \nabla \tau_{ij}^k(t, t+1)]_{\tau_{min}}^{\tau_{max}}, \quad (10)$$

Where

$$w(k) = \binom{v'}{k} = \frac{v'(v'+1)\dots(v'+k-1)}{k!} \quad (11)$$

$$\nabla \tau_{ij}^k(t, t+1) = \begin{cases} \frac{Q}{L^k}, & \text{if } (i, j) \in \text{route performed by the } k\text{th ant} \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

In Eq. (10), the difference coefficient of the fractional order $w(k)$ is imported as a weight of ants which is determined by the tour length of the ant. So, MMAS is improved to be a multi-scale method through the nonlinear weighting. r represents the number of ants which are selected to update their pheromone trials. k is a coefficient and varies from 1 to r . $\nabla \tau_{ij}^k$ is the quantity of pheromone laid on path (i, j) by the k th ant between time t and $t+1$. In Eq. (11), v' is a fractional-order degree and is set to a positive number. Besides, $\nabla \tau_{ij}^k(t, t+1)$ can be calculate as Eq. (12), Q is a constant and is the tour length of the ant and L^k indicates the length of the tour of the k th ant. Referring to Eq. (10), suppose that FS-MMAS could converge to the global optimization after t iterations and Q is set to 1. Hence, Eq. (10) can be expressed as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \frac{1}{L^{best}} \times \sum_{k=1}^r w(k) \quad (13)$$

In Eq. (13), L^{best} represents the value of the global optimization. So, the maximum of the pheromone trial τ_{max} can be express as follows:

$$\tau_{max} = \frac{1}{\rho} \times \frac{\sum_{k=1}^r w(k)}{L^{best}} \quad (14)$$

Suppose that the number of cities is set to N . Referring to Eq. (14), the minimum of the pheromone trial τ_{min} can be express as follows:

$$\tau_{min} = \frac{\tau_{max}}{n} = \frac{1}{n\rho} \times \frac{\sum_{k=1}^r w(k)}{L^{best}} \quad (15)$$

Algorithm: FS-MMAS

1. In the initialization processing, $\tau(i, j) = \tau_0$.
2. for $1 \leq k \leq \text{iterator}_{max}$ do
3. for $1 \leq s \leq M$ do
4. M ants are randomly placed in N cities.
5. The tabu of route visited is set null.
6. while (the root of s ant has next town) do
7. The s ant select next town according to P_{ij}^k .
8. Then select other probable towns x which satisfy $d_{ix} \leq \lambda * d_{ij}$.
9. Add the j and x to the tabu.
10. end while
11. end for
12. Update the pheromone trials
13. end for

4. Experimental Results

The accuracy and performance of the proposed method is

measured by standard deviation and average tour length on eight test problems taken from TSPLIB [13]. During the experiment, the number of ants is the same as the number of cities. In Eq. (8), many extra routes may be obtained but less than the number of ants. ρ is set to 0.3 and the value of $[\alpha, \beta]$ is set as $[1,5]$ in all experiments. Q is a constant that is set to 1. FS-MMAS is operated throughout 1500 iterations and each

TSP problem is repeated 50 times.

Fig. 2 shows that how the 3-Opt algorithm works. Lengths of the route are improved by the 3-Opt algorithm. 3-Opt algorithm is a local optimization algorithm, which can remove local cross-paths.

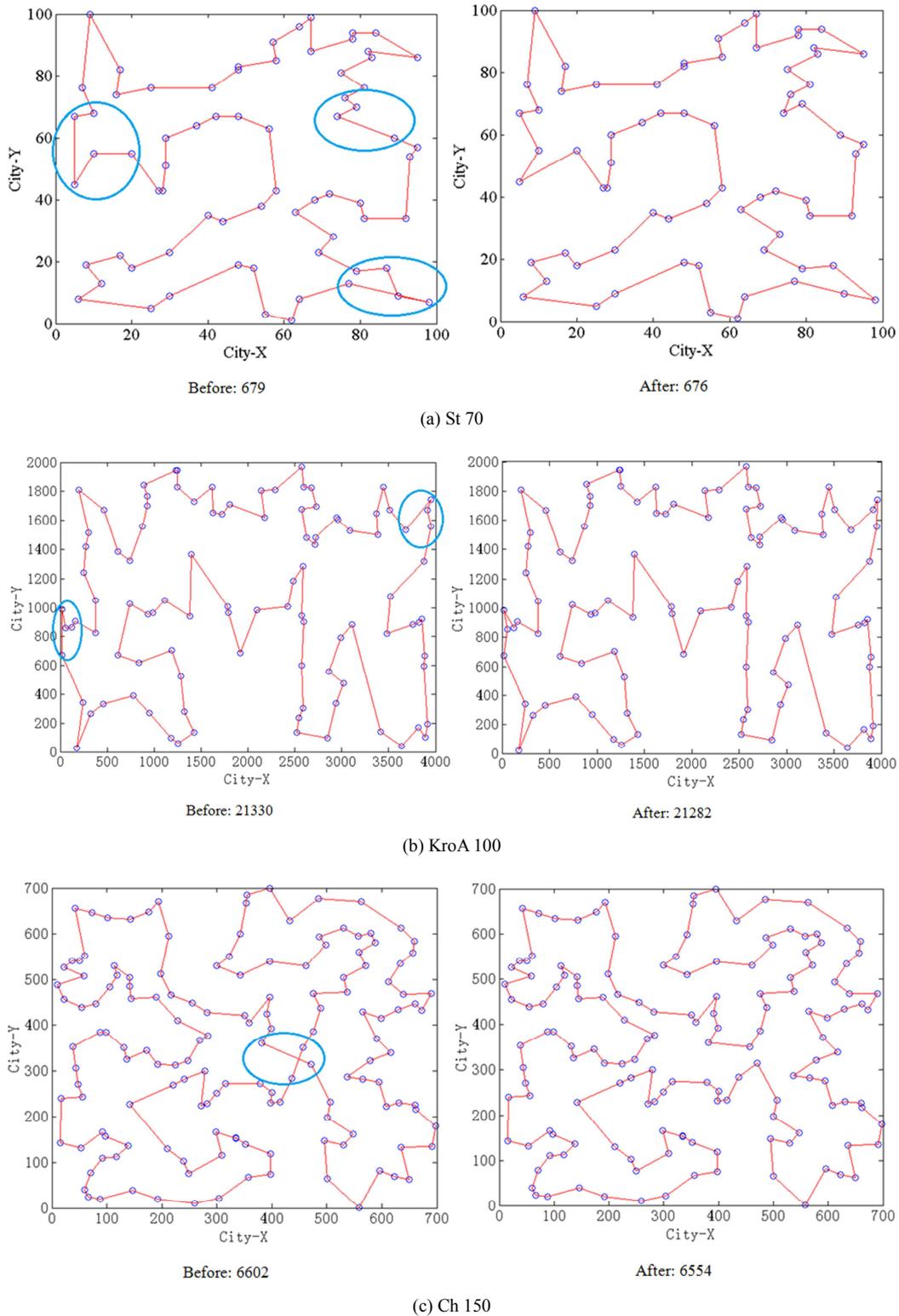


Fig. 2. Before and After the execution of 3-Opt algorithm.

In Table 1, Opt represents the global optimization. Best and Worst represent the shortest and longest route obtained after iteration. Average represent the average route of 50 runs. Prob. indicates the probability of the global optimization obtained by FS-MMAS. SD is the standard deviation and Error (%) represents the percentage relative error of results obtained by 50 runs.

Table 1. The results obtained by FS-MMAS for TSP.

Problem	Opt	Best	Prob.	Worst	Average	SD	Error (%)
Eil51	426	426	0.24	427	426.76	0.43	0.18
Berlin52	7,542	7,542	1.00	7,542	7,542.00	0.00	0.00
Eil76	538	538	0.74	545	538.74	1.65	0.14
St70	675	675	0.20	685	676.98	1.88	0.29
Kroa100	21,282	21,282	0.25	21,387	21,336.05	43.40	0.25
Eil101	629	629	0.40	644	633.20	5.07	0.67
Lin105	14,379	14,379	0.94	14,479	14,385.00	23.75	0.04
Ch150	6,528	6,533	0.02	6,590	6,555.97	7.94	0.43

Table 2. The comparison of FS-MMAS and other algorithms in the literature. OPT is the global optimization; Avg. is the average tour length; SD is the standard deviation and Error (%) is the percentage relative error.

Method	Problem OPT	Eil51 426	Berlin52 7542	Eil76 538	St70 675	Kroa100 21,282	Eil101 629	Lin105 14,379	Ch150 6,528
ACOMAC (2004) [14]	Avg.	430.68		555.70		21,457.00			
	SD								
	Error (%)	1.10		3.29		0.82			
ACOMAC+NN (2004) [14]	Avg.	430.68		555.90		21,433.30			
	SD								
	Error (%)	1.10		3.33		0.71			
RABNETTSP (2006) [15]	Avg.	438.70	8,073.97	556.10		21,868.47	654.83	14,702.17	6753.20
	SD	3.52	270.14	8.03		245.76	6.57	328.37	83.01
	Error (%)	2.98	7.05	3.36		2.76	4.11	2.25	3.45
Modified RABNETTSP (2009) [16]	Avg.	437.47	7932.50	556.33		21,522.73	648.63	14,400.7	6738.37
	SD	4.20	277.25	5.30		93.34	3.85	44.03	76.14
	Error (%)	2.69	5.18	3.41		1.13	3.12	0.15	3.22
SA ACO PSO (2011) [17]	Avg.	427.27	7542.00	540.20		21,370.30	635.23	14,406.37	6563.70
	SD	0.45	0.00	2.94		123.36	3.59	37.28	22.45
	Error (%)	0.30	0.00	0.41		0.41	0.99	0.19	0.55
ACO+2 opt (2012) [18]	Avg.	439.25	7556.58			23,441.80	672.37		
	SD								
	Error (%)	3.11	0.19			10.15	6.90		
WFA with 3 opt (2013) [19]	Avg.	426.60	7542.00	539.44		21,282.80	633.50	14,459.40	6700.10
	SD	0.52	0.00	1.51		0.00	3.47	1.38	60.82
	Error (%)	0.14	0.00	0.27		0.00	0.72	0.56	2.64
ACO with Taguchi Method (2013) [20]	Avg.	435.40	7635.40	565.50		21,567.10	655.00	14,475.20	
	SD								
	Error (%)	2.21	1.24	5.11		1.34	4.13	0.67	
ACO with ABC (2014) [21]	Avg.	433.39	7544.37	557.98	700.58	22,435.31	683.39		6677.12
	SD	5.25	0.00	4.10	7.51	231.34	6.56		19.30
	Error (%)	4.08	0.03	3.71	3.79	5.42	8.65		2.28
Propose Method FSMAS	Avg.	426.76	7542.00	538.74	676.98	21,336.05	633.20	14,385.00	6555.97
	SD	0.43	0.00	1.65	1.88	5.07	5.07	23.75	7.94
	Error (%)	0.18	0.00	0.14	0.29	0.67	0.67	0.04	0.43

Table 3. The comparison between FS-MMAS and MMAS.

Method	Problem OPT	Eil51 426	Berlin52 7542	Eil76 538	St70 675	Kroa100 21,282	Eil101 629	Lin105 14,379	Ch150 6,528
MMAS	Avg.	427.23	7542.17	539.03	677.34	21,303.90	634.90	14382.10	6554.29
	SD	0.87	5.47	1.40	3.04	39.82	2.13	15.75	4.03
	Error (%)	0.29	0.002	0.19	0.35	0.10	0.94	0.02	0.40
FS-MMAS	Avg.	426.76	7542.00	538.74	676.98	21,336.05	633.20	14,385.00	6555.97
	SD	0.43	0.00	1.65	1.88	43.40	5.07	23.75	7.94
	Error (%)	0.18	0.00	0.14	0.29	0.25	0.67	0.04	0.43

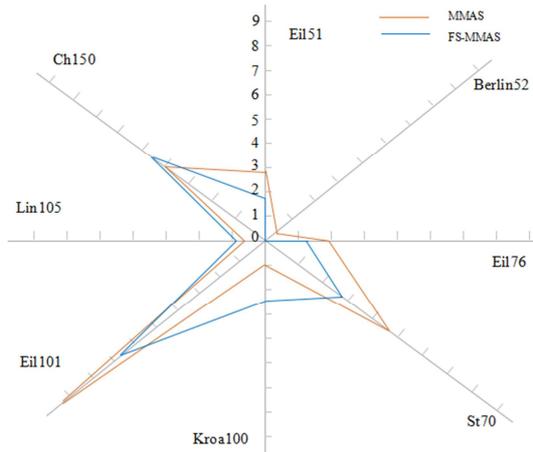


Fig. 3. Comparison of Error (%) between the FS-MMAS and MMAS.

In Table 2, eight experimental results of FS-MMAS are compared with the results of studies in the literature. It can be seen that FS-MMAS has better performance in terms of Berlin 52, Eil 76, St 70, Eil 101, Lin 105 and Ch 150. More precisely, the results obtained by FS-MMAS for Eil 51, Berlin 52, Eil 76, St 70, KroA 100 and Lin 105 are close to the global optimization whose percentage relative errors are less than 0.3%. As we see from Table 3 and Fig. 3, the results of FS-MMAS for Eil 51, Berlin 52, Eil 76, St 70 and Eil 101 are better than those by MMAS. However, the results obtained by FS-MMAS for KroA 100, Lin 105 and Ch 150 are similar to those by MMAS.

5. Conclusion

In this paper, a fractional-order weighted and self-adaptive method based on the Max-Min Ant System with 3-Opt algorithms is proposed in order to make full use of spatial information and find the global optimization. In this method, ants could select the next city multi-scalely and self-adaptively, which contributes to improving the probability of finding the global optimization. And 3-Opt algorithm is used to avoid falling into local minimums. Eight different benchmark problems were tested with FS-MMAS, whose performance is determined by standard deviation and average tour length. In most cases, FS-MMAS has better performance than other methods. It can be concluded that FS-MMAS is effective and efficient to solve the optimization problems. There is no doubt to see further developments in application based on FS-MMAS in the future. So the next research will focus on the application of the FS-MMAS algorithm, and we expect that it has better performance at the design of Large Scale Integrated circuit, Vehicle Scheduling Problem, Routing Problem etc.

References

- [1] G. Laporte, The Traveling Salesman Problem – an overview of exact and approximate algorithms, *Eur. J. Oper. Res.* 59, 1992, pp. 231–247.
- [2] Wikipedia, Traveling Salesman Problem. Available: http://en.wikipedia.org/wiki/Travelling_salesman_problem
- [3] X. T. Geng, Z. H. Chen, W. Yang, D. Q. Shi, K. Zhao, "Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search", *Appl. Soft Comput.* 11, 2011, pp. 3680–3689.
- [4] J. Grefenstette, R. Gopal, B. Rosmaita, D. Van Gucht, "Genetic algorithms for the traveling salesman problem", in *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, Lawrence Erlbaum, NJ, 1985, pp. 160–168.
- [5] Dorigo. M, Maniezzo. V, Colomi. A, "Ant system: optimization by a colony of cooperating agents", *IEEE Trans. Syst. Man Cybern. B* 26, 1996, pp. 29–41.
- [6] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, Q. X. Wang, "Particle swarm optimization based algorithms for TSP and generalized TSP", *Inf. Process. Lett.* 103, 2007, pp. 169–176.
- [7] Dorigo, M., & Gambardella, L. M, "Ant Colony System: A cooperating learning approach to the traveling salesman problem", *IEEE Transactions on Evolutionary Computation*, 1 January 1997, pp. 53–66.
- [8] Bullnheimer, B., Hartl, R. F., & Strauss, C., "A new rank-based version of the Ant System: A computational study", *Central European Journal for Operations Research and Economics*, 7 January 1996, pp. 25–38.
- [9] Stutzle, T., & Hoos, H. H., "Max–min ant system", *Future Generation Computer Systems*, 16 August 2000, pp. 889–914.
- [10] M. Dorigo, V. Maniezzo, A. Colomi, V. Maniezzo, *Positive Feedback as a Search Strategy*, 1991.
- [11] Oldham K B, Spanier J., *The Fractional Calculus*, New York and London: Academic Press, 1974.
- [12] Wikipedia, 3-Opt Algorithm. Available: <http://en.wikipedia.org/wiki/3-Opt>
- [13] G. Reinelt, "TSPLIB—a traveling salesman problem library", *ORSA J. Comput.* 3, 1991, pp. 376–384.
- [14] C. F. Tsai, C. W. Tsai, C. C. Tseng, "A new hybrid heuristic approach for solving large traveling salesman problem", *Inf. Sci.* 166, 2004, pp. 67–81.
- [15] R. Pasti, L. N. De Castro, "A neuro-immune network for solving the traveling sales-man problem", in *International Joint Conference on Neural Networks*, 2006. IJCNN'06, IEEE, 2006, pp. 3760–3766.
- [16] T. A. S. Masutti, L. N. de Castro, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem", *Inf. Sci.* 179, 2009, pp. 1454–1468.
- [17] S. M. Chen, C. Y. Chien, "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques", *Expert Syst. Appl.* 38, 2011, pp. 14439–14450.
- [18] K. Jun-man, Z. Yi, "Application of an improved Ant Colony Optimization on generalized Traveling Salesman Problem", *Energy Proc.* 17, 2012, pp. 319–325.

- [19] Z. A. Othman, A. I. Srour, A. R. Hamdan, P. Y. Ling, "Performance water flow-like algorithm for TSP by improving its local search", *Int. J. Adv. Comput. Technol.* 5, 2013.
- [20] M. Peker, B. Sen, P. Y. Kumru, "An efficient solving of the traveling salesman problem: the ant colony system having parameters optimized by the Taguchi method", *Turk. J. Electr. Eng. Comput.* 21, 2013, pp. 2015–2036.
- [21] M. Gunduz, M. S. Kiran, E. Ozceylan, "A hierarchic approach based on swarm intelligence to solve traveling salesman problem", *Turk. J. Electr. Eng. Comput. Sci.*, 2014.