AN IMPROVED ANT COLONY ALGORITHM BASED ON 3-OPT AND CHAOS FOR TRAVELLING SALESMAN PROBLEM

Qingping Yu[1] ,Xiaoming You[1] and Sheng Liu[2]

[1] College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
[2] School of Management, Shanghai University of Engineering Science
Shanghai, 201620, China

ABSTRACT

This paper presents an improved chaotic ant colony system algorithm (ICACS) for solving combinatorial optimization problems. The existing algorithms still have some imperfections, we use a combination of two different operators to improve the performance of algorithm in this work. First, 3-opt local search is used as a framework for the implementation of the ACS to improve the solution quality; Furthermore, chaos is proposed in the work to modify the method of pheromone update to avoid the algorithm from dropping into local optimum, thereby finding the favorable solutions. From the experimental results, we can conclude that ICACS has much higher quality solutions than the original ACS, and can jump over the region of the local optimum, and escape from the trap of a local optimum successfully and achieve the best solutions. Therefore, it's better and more effective algorithm for TSP.

KEYWORDS

Ant colony system algorithm, 3-opt, Chaos, Travelling Salesman Problem

1. INTRODUCTION

Swarm intelligence is an emerging evolutionary computation method that takes inspiration from the social behaviors of insects and of other animals, getting more and more attention from researchers. In particular, [1] ants have inspired a number of methods and techniques among which the most studied and the most successful one is the general purpose optimization technique known as ant colony optimization (ACO). ACO algorithm is a heuristic in which a colony of artificial ants cooperates in finding good solutions [2]. Ants release the pheromone on the path to mark some favorable paths and communicate through pheromones with other ants of colony. As time goes on, the higher the concentration of the pheromone is on the shorter path from their nest to the food source. Thus, most ants tend to the path with highest pheromone intensity i.e., the shortest path.

The ant colony optimization (ACO) algorithm was proposed by Dorigo and his colleagues as a method for solving optimization problems [3], and developed by Dorigo, referred to as ant system(AS). Ant colony system (ACS) is one of the most successful ACO algorithms which achieve a much better performance than ant system (AS) which was introduced by Colorni,

DOI: 10.5121/ijci.2014.3501

Dorgio and Maniezzo. In this work, ACS has been chosen as an original algorithm for solving travelling salesman problem (TSP), and will be improved by two strategies.

The traveling salesman problem (TSP) is a classical combinatorial optimization problem which is to find a cost-minimal Hamiltonian cycle, i.e. a cycle in a graph that contains each node exactly once [4].Moreover, TSP is the famous NP-hard problem even though to descript the problem is very simple, it is very difficult to solve, and to obtain the optimal solution of such problems requires extremely long running time and a great storage space, so existing computers could not be completed. ACS algorithm as an approximation algorithm to solve TSP reflects the combination of global and local search, showing good optimization results.

Now, TSP has become a standard test for heuristic algorithms whether to find the optimal solutions. So the ACS can be applied to the TSP . In this work, we proposed an ICACS algorithm for TSP, based on 3-opt and chaos, and applied successfully to eil51、KroA100、KroA200、pr264 and Lin318 in TSP problems. This paper is organized as follows: In Section 2, we further to study the original ACS, and give a brief description of TSP. Section 3 presents our approach and describe in detail its main components. In the next Section, we perform simulation on TSP which contains a different number of nodes and relate this analysis to the performance of original ACS. Moreover, we explore the importance of the operators proposed in this work to the overall algorithm performance. Conclusions are provided in the last Section.

2. Ant Colony System Algorithm (ACS) for TSP.

It is well known that ants find a path from nest to the food source very quickly by using indirect communication via pheromones. As ants pass the path it release pheromone trail that can be discovered by other ants. The path more ants pass, more pheromones are deposited. Because ants move according to the intensity of pheromones, the richer the pheromone trail on a path is, the more likely it would be followed by other ants. Hence, ants can construct the shortest way from their nest to the food sources and back [5]. This ants' behavior inspired researchers to build an algorithmic framework that uses artificial ants to solve combinational optimization problems [6]. In the following, we will provide a formal definition of the basic ACS.

2.1. Ant Colony System Algorithm

In this work, ant colony system has been chosen as an original algorithm for solving TSP problem. Here we give a brief description of ACS algorithm:

In ACS, the most interesting thing is the transition rules to ants' searching solutions. When an ant chooses the next nodes, it no longer completely obeys the previous experience, but it has a certain probability to choose the shorter routers with higher pheromone intensity. It is called pseudorandom proportional rule, defined by Dorigo and Stülzle [3] as:

$$j = \begin{cases} \arg max\left\{\left[\tau_{il}\right]\left[\eta_{jl}\right]^{\beta}\right\}, & q \le q_0 \\ J\left(p_{ij}^{k}\right), & q > q_0 \end{cases} \tag{2.1-1}$$

Where q is a random variable uniformly distributed in $[0,1]$ $(\mathrm{q} \in [0,1])$, q_0 is a parameter according to the previous empirical values used in this case for the best possible move $(q_0 \in [0,1])$, k is an ant, i, j are the initial and the next node, l is a candidate solution that ant k have not travel the node in a loop, J is a random function gained from the probability distribution p_{ij}^k (see 2.1-2), and it will be chose by the ant k if $q > q_0$.

$$p_{ij}^k = \frac{\left[\tau_{ij}\right]^\alpha \left[\eta_{ij}\right]^\beta}{\sum_{l\in N_i^k}\left[\tau_{ij}\right]^\alpha [\eta_{ij}]^\beta} \tag{2.1-2}$$

Parameter β determines the relative influence of the heuristic information, η_{ij} $(\eta_{ij} = \frac{1}{d_{ij}})$ is a heuristics function that is defined to the inverse of the distance between node i and j . N_i^k is the set of candidate nodes connected to node i , with respect to ant k . τ_{ij} represents the pheromone from node i to node j , After constructing its tour, only the ant to find the shortest path to the current loop was allowed to update the pheromone by applying the pheromone updating rule (2.1-3), which accelerates the convergence of the algorithm to some extent.

$$\tau_{ij}(\mathrm{t}) = \left(1-\varphi\right)\tau_{ij}(\mathrm{t}) + \varphi\Delta\tau_{ij}(\mathrm{t}) \tag{2.1-3}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{n}\Delta\tau_{ij}{}^k \tag{2.1-4}$$

With φ in (0, 1), and $\Delta\tau_{ij}(\mathrm{t})$ is the quantity of pheromone deposited by the ant k between time t and $\mathrm{t+1}$ on the edge $(\mathrm{i,j})$. In addition to the global update rule, ACS uses the ant-cycle updating rule. An ant-cycle system information update model is

$$\Delta\tau_{ij}{}^k = \begin{cases} \frac{Q}{L_k}, & arc(\mathrm{i,j})\ \text{belong to best loop} \\ 0, & \text{otherwise} \end{cases} \tag{2.1-5}$$

Q represents the intensity of the pheromone, it affects the convergence speed of algorithm to some extent, L_k denotes the total length that the ant k passed in this cycle .

2.2. Description of the Traveling Salesman Problem

Given a digraph $G = (\mathrm{V,E})$, where $\mathrm{V} = \{1,2,\cdots,n\}$ presents city set and E presents edge set. Variables are defined as follows [7]:

$$x_{ij} = \begin{cases} 1 & (\text{if the edge } (\mathrm{i,j}) \text{ in the loop}) \\ 0 & \textit{otherwise} \end{cases} \tag{2.2-1}$$

So, TSP problem can be descripted by linear programming:

$$\min\sum_i\sum_j d_{ij}x_{ij} \tag{2.2-2}$$

$$\text{s.t.}\begin{cases} \sum_{i} x_{ij} = 1 \quad (\mathrm{i}=1,\dots,\mathrm{n}), & (a)\\ \sum_{j} x_{ij} = 1 \quad (\mathrm{j}=1,\dots,\mathrm{n}), & (\mathrm{b})\\ x_{ij} \in \{0,1\} \quad (\mathrm{i},\mathrm{j}=1,\dots\mathrm{n};\mathrm{i}\neq\mathrm{j}), & (\mathrm{c})\\ \sum_{i,j\in s} x_{ij} \le |S|-1 \quad (\mathrm{S}\subset\mathrm{V};2\le|S|\le\mathrm{n}-2), & (\mathrm{d}) \end{cases} \qquad (2.2\text{-}3)$$

Where n presents the number of cities, d_{ij} is the Euclidean distance between city i and city j, and it was calculated by using $d_{ij}=\sqrt{(x_i-x_j)^2+\quad(y_i-y_j)^2}$. If the edge (i, j) in the loop, $x_{ij}=1$, else, $x_{ij}=0$. V presents the city set and the set S is a subset of V, $|S|$ is the number of elements of set S. In the formula expressions (2.2-3), the constraint items (a), (b), (c) ensure there is only one path to the city as a starting point and one path to the city as a ending point, Similarly, constraint items (d) indicate that sub-loop set of constraints can limit the feasible solutions constitute a loop.

3. THE IMPROVED CHAOTIC ANT COLONY SYSTEM (ICACS) FOR TSP

In this section we will introduce the two strategies used in ACS in this work. One is the introduction of the chaotic disturbance to change the update way of the pheromone, to avoid the algorithm from dropping into local optimal. The other is the 3-opt that is used to improve the solution quality.

3.1. Improved Ant Colony System Algorithm based on 3-OPT

A 3-OPT move changes a tour by replacing 3 edges from the tour by 3 edges in such a way that a shorter tour is achieved. Given a feasible TSP tour, the algorithm repeatedly performs exchanges that reduce the length of the current tour, until a tour is reached for which no exchanges yields an improvement [8]. This process may be repeated many times from initial tours generated in some randomized way to an optimal rout.

Let T is the current tour without a 3-OPT operation. Set $\mathrm{X}=\{x_1,x_2,\cdots,x_i\}$ and $\mathrm{Y}=\{y_1,y_2,\cdots,y_i\}$, if the edges of X are deleted from T and replaced by the edges of Y, thereby gaining a better tour. The two sets X and Y are empty initially, with the number of iterations increase, they are established element by element. x_i and y_i must share an endpoint, and so must y_i and x_{i+1}. If q_1 denotes one of the two points of x_1, we have in general: $x_i=(q_{2i-1},q_{2i})$, $y_i=(q_{2i},q_{2i-1})$ and $x_{i+1}=(q_{2i+1},q_{2i+2})$ for $i\ge 1$ [8]. Note that the exchange of edges X with edges Y which results in a tour is that the chain is closed (see figure 3.1.1), and the result of an close tour by 3-opt move shows follow(see figure 3.1.2).

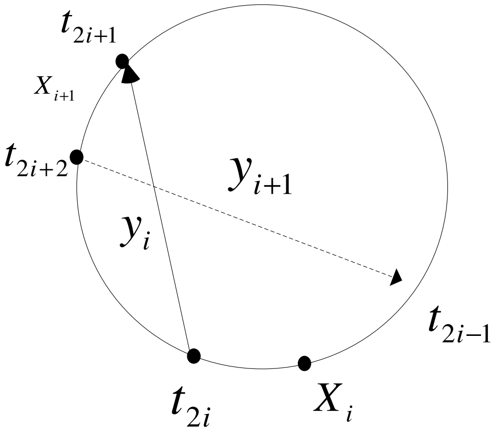


Fig. 3.1.1 Restricting the choice of x_i, y_i, x_{i+1} and y_{i+1}

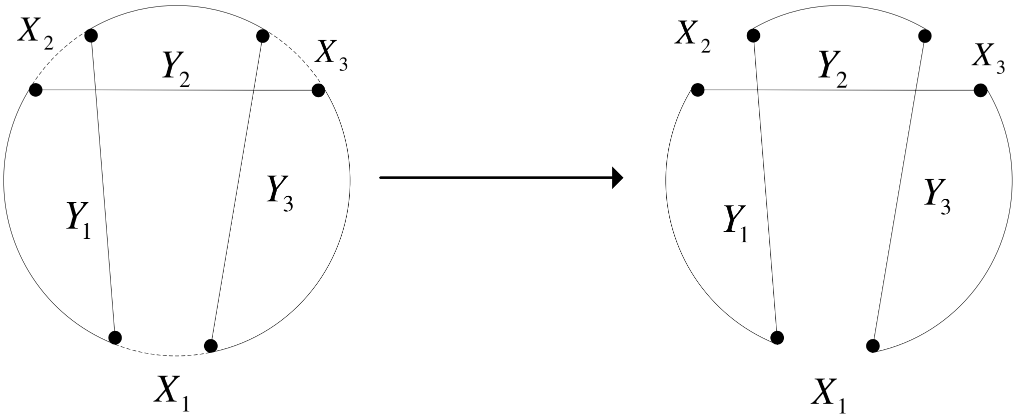


Fig.3.1.2 A 3-opt move. x_1, x_2, x_3 are replaced by y_1, y_2, y_3

The improved ant colony system algorithm based on 3-opt has greatly improved the accuracy of the solutions, but did not find the optimal solution, still need to be further improved. The simulation data refer to the table(4.2).

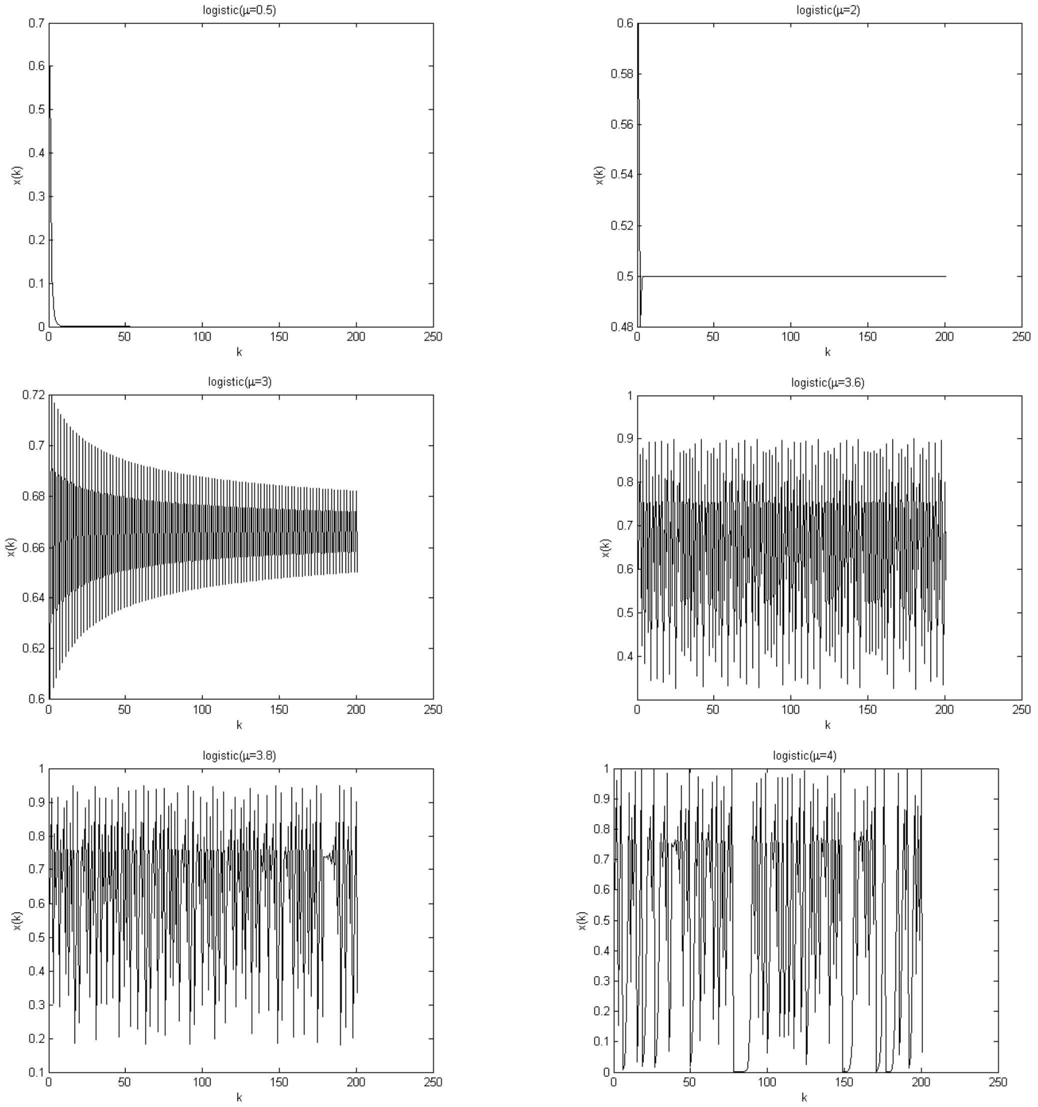
3.2. Improvement based on chaos

Chaos is different from random, which does not mean it is completely disordered, but includes in the orderly disorder, i.e., it is implicated that local random and global stability in the chaos. The main characteristics of chaos are that pseudo-randomness, ergodicity and sensitivity to initial conditions.

When ant colony algorithm is initialized, pheromone on every path are equally so that ants choose the path with same probability, and it is difficult to make ants to find a better path quickly from a large number of the chaotic path, leading a slow convergence. So to use chaos operator is necessary that it can increase the searching efficiency by the random and ergodic of chaos [9]. At present, there is no strict definition to chaos. Generally, the state of random motion will be called chaos obtained from deterministic equations. Logistics mapping as a typical chaotic system is used in the work which has already been the foundation of the chaos optimization algorithm, iterative formula is as follows:

$$x_{k+1}=\mu(1-x_k),\mathrm{k}=0,1,2,...;x_0\in\left[0,1\right]\mu\in\left(3,4\right] \qquad (3.2\text{-}1)$$

For each identified, we can obtain corresponding series $x_1,x_2,x_3,...x_k$. By discussing the value of μ, we can easily see that: when $0<\mu\leq 1$, system dynamics shape is very simple, only one periodic point $x_0=0$; When $1<\mu<3$, the system dynamics shape is also simple, two-cycle point $0,1-\frac{1}{m}$; when $3\leq\mu\leq 4$, forming a very complex system dynamics, the system by the period-doubling to chaos. Here is the Logistic map with Matlab simulation. Let the initial system m is 0.5, the number of iteration is 200, and the $\mu=0.5,2,3,3.6,3.8,4$, draw the x_k corresponding to different μ.



Where μ is the control parameter, when $\mu = 4$, $0 \le x_0 \le 1$, logistic completely drops into chaotic state.

Ant colony is based on the positive feedback principle, to a certain extent, speeding up the evolutionary process, but there are some shortcomings, such as phenomenon of stagnation or fall into local optimal solution. Improved measures of chaotic disturbance added to adjust the amount of information, by adding the amount of the chaotic scramble to make the solution out of the local extreme range. Equation (2.1-3) is amended as Equation (3.2-2):

$$\tau_{ij}\left(t+1\right) = \left(1-\varphi\right)\tau_{ij}\left(t\right) + \varphi\Delta\tau_{ij}\left(t\right) + qx_{ij} \qquad (3.2\text{-}2)$$

where x_{ij} is a chaotic variable obtained from the formula (3.2-1) iteration; q is a coefficient.

3.3 The Procedure of Improved Chaotic Ant Colony System (ICACS)

We design new algorithms ICACS based on the two improving strategies presented in this paper. In this section, we describe the specific steps of the ICACS algorithm. The steps of the algorithm described in detail as follows:

Step 1.Chaotic initialization, initialize the pheromone of all path section, set the iteration number $i = 0$.

Step 2. Put the starting city into $tabu_k$, choose one subsequent node j to move according to (2.1-1), finally put node j into $tabu_k$.

Step 3.Loop (2) until each ant find a closed tour, calculate the tour length L_k of ant k , and record the current optimal solution, adjust the current optimal solution by 3-opt operate.

Step 5.Update the pheromone intensity of paths which is shorter than L_i (it has already set in this work) by formula (3.2-2).

Step 6.The number of iterations plus 1.

Step 7.If the number of iteration reach the maximum number of iterations and find the same solutions, output the best solution.

Step 8.To clear the $tabu_k$ of each and move to (2)

4. EXPERIMENTS AND SIMULATION

In this section, the simulation is implemented in Matlab, we describe the performance of the improved algorithm in the work and analyze the experiment results as a function of various parameters.

To examine the effectiveness of the two sorts of strategies, a series of standard TSP problems are employed. The settings of the parameters of the improved chaotic ant colony system (ICACS) algorithms refer to the follow table (Table4.1)

Table 4.1 ICACS parameters (present work)

Parameter	α	β	φ	q_0	μ
Value	1	2	0.1	0.9	4

Table 4.2 report the experimental results for the two operators used in this work on TSP problems. As can be seen, tour quality is acceptable. The convergence comparison is as Figure 4.1.

Table 4.2 Comparison of the results in TSP instances

Data set	**Algorithm**	**Obtained solution**	**Optimual solution**	**IR**
eil51	ICACS ACS+3-opt ACS	426.21 428.87 430.67	426	0 0.67% 1.10%
kroA100	ICACS ACS+3-opt ACS	21282.39 21294.78 21571.56	21282	0 0.06% 1.36%
KroA200	ICACS ACS+3-opt ACS	29369.57 29385.93 29775.43	29368	0 0.06% 1.39%
pr264	ICACS ACS+3-opt ACS	49139.72 49165.60 49978.72	49135	0.01% 0.06% 1.72%
Lin318	ICACS ACS+3-opt ACS	42185.91 43238.95 44067.76	42029	0.37% 2.88% 4.85%

(Note: $IR = \frac{onatained \quad solution - optimual \quad solution}{optimual \quad solution} \%$)

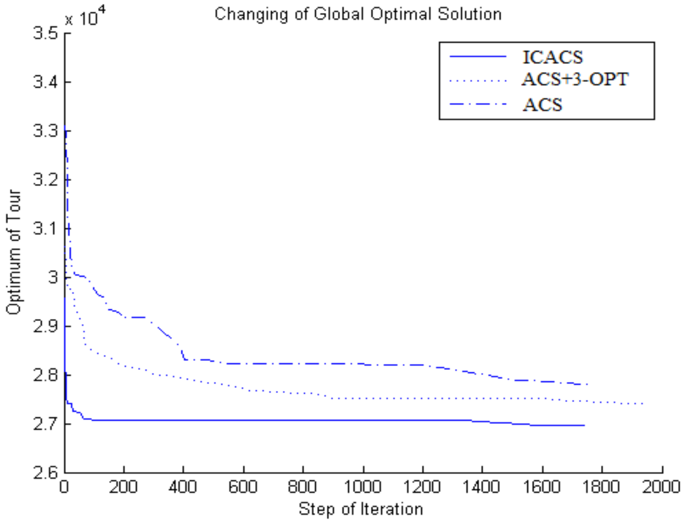


Figure 4.1 The convergence comparison of ICACS with other algorithm using KroA 150.

5. Conclusions

This work focused on how to improve the performance of ACS algorithm. And we adopt two improving strategies employed by the original ACS algorithm: improvement based on 3-opt to improve the local search capability and improvement based on chaos to modify the update of pheromone trails. Through a number of experiments, the effectiveness and advantages of the strategies are verified. Specifically, firstly, improvement based on 3-opt can improve the quality of solutions. Secondly, improvement based on chaos can avoid the algorithm from dropping into local optimal and improve the probability of algorithms to find optimal solution. These two strategies possess their own features and offer a clear guideline for designing the new ACO algorithm. In the future, more experiments and further research about the improved strategies are required.

Further study should be suggested to explore the better management for the optimal setting of the parameters of ACO algorithms, which will be very helpful in the application.

Acknowledgements

The authors gratefully acknowledge the support of Innovation Program of Shanghai Municipal Education Commission (Grant No.12ZZ185), Natural Science Foundation of China (Grant No.61075115), Foundation of No. XKCZ1212. Xiao-Ming You is corresponding author.

References

[1] Nan Zhao , Zhilu Wu, Yaqin Zhao, Taifan Quan: Ant colony optimization algorithm with mutation mechanism and its applications. Expert Systems with Applications 37

[2] Rubem Matimoto Koide,Gustavo von Zeska de França andMarco Antônio Luersen: An ant colony algorithm applied to lay-up optimization of laminated composite plates. Latin American journal of solids and structures, 2013: 491-504

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

[4] TORE GRU¨ NERT, Stefan Irnich, A Note on Single Alternating Cycle Neighborhoods for the TSP. Journal of Heuristics, 11: 135–146, 2005

[5] Rongwei Gan, Qingshun Guo, Huiyou Chang, and Yang Yi: Improved ant colony optimization algorithm for the traveling salesman problems. Journal of Systems Engineering and Electronics Vol. 21, No. 2, April 2010, pp.329–333

[6] Timo Kötzing • Frank Neumann • Heiko Röglin •Carsten Witt, Theoretical analysis of two ACO approaches for the traveling salesman problem. Swarm Intelligence (2012) 6:1–21.

[7] Lixiang Li, Haipeng Peng, Yixian Yang: Chaos Ant Colony Algorithm and Applications [M].Beijing: China Science and Technology Press, 2013:186-208

[8] Keld Helsgaun, General k-opt submoves for the Lin–Kernighan TSP heuristic. Math. Prog. Comp. (2009) 1:119–163

[9] Yunwu, W. (2009). Application of chaos ant colony algorithm in web service composition based on QoS. International Forum on Information Technology and Applications, 172(2), 225–227.

Authors

Qing-ping Yu is currently studying in Mechanical and Electronic Engineering from Shanghai University of Engineering Science, China, where she is working toward the Master degree. Her current research interests include ant colony algorithm, their design and develop in Embedded system.

Xiao-Ming You received her M.S. degree in computer science from Wuhan University in 1993, and her Ph.D. degree in computer science from East China University of Science and Technology in 2007. Her research interests include swarm intelligent systems, distributed parallel processing and evolutionary computing. She now works in Shanghai University of Engineering Science as a professor.