International Journal of Database Theory and Application
Vol.8, No.5 (2015), pp.195-206
http://dx.doi.org/10.14257/ijdta.2015.8.5.16

# A Novel Hybrid Evolution Optimization Algorithm and its Application

Bin Gao, Jing-Hua Zhu and Wen-Chang Lang

*Wenzhou Key Laboratory of Material Processing and Die & Mould Technology, Wenzhou Vocational & Technical College, Wenzhou 325035, PR China*

## *Abstract*

*For the premature convergence and initial pheromone distribution problem of ant colony optimization algorithm, an improved particle swarm optimization (MPSO) algorithm is introduced into ant colony optimization algorithm in order to propose a novel hybrid evolution optimization (HEACO) algorithm in this paper. In the proposed HEACO algorithm, the ergodicity of the chaos is used to initialize the swarm in order to enhance the diversity of the particle swarm, and adjust the mutation probability and inertia weighting factor in order to improve the capability of local and global search. Then the MPSO algorithm is used to control the parameters of the heuristic factor, pheromone evaporation coefficient, and the stochastic selection threshold in order to effectively overcome the parameter influences of ACO, reduce the numbers of useless experiments and balance the developing optimal solution and enlarging search space. A series of typical traveling salesman problems are selected to validity the effectiveness of the proposed HEACO algorithm. The simulation results show that the performance of the proposed HEACO algorithm is better than the traditional ACO algorithm and PSO algorithm. So the proposed HEACO algorithm can effectively improve the solving efficiency and quality, and speed up the convergence and computation.*

***Keywords:*** *ant colony optimization algorithm; particle swarm optimization algorithm; travelling salesman problem; chaos; control parameter*

## 1. Introduction

The traveling salesman problem (TSP) is a well-known combinatorial optimization problem and NP-complete problem, which find a route covering each city once and only once with a minimum route distance by the salesman [1]. Because the TSP and its variants are a good ground for testing optimization techniques, many researchers proposed a lot of solving methods for solving TSP in various fields of artificial intelligence, biology, mathematics, physics, and operations research and so on [2-5]. These solving methods include genetic algorithms [6], ant colony optimization algorithm [7], simulated annealing [8], particle swarm optimization algorithm [9], neural networks [10], evolutionary algorithms [11], immune algorithm…, *etc*.

Ant colony optimization (ACO) algorithm was introduced by Marco Dorigo in the early 1991 [12]. It is a branch of newly developed form of artificial intelligence called swarm intelligence, which studies ‘‘the emergent collective intelligence of groups of simple agents”. When ants move, ants will leave a chemical pheromone trail on the ground. The indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources. The ACO algorithm has received increased interests to solve optimization problems of actual engineering in recent years. But in the actual engineering application, the ACO algorithm has some deficiencies, such as stagnation phenomenon, slow solving speed and poor solution quality with improper parameter values, and so on. Therefore, many researchers

ISSN: 2005-4270 IJDTA

proposed a lot of improved ACO algorithm to solve the TSP. Wu *et al.* [13] proposed population declining ant colony optimization (PDACO) algorithm to the traveling salesman problem (TSP) and multiuser detection. The PDACO can enlarge searching range through increasing the initial population of the ant colony, and the population declines in successive iterations. Borkar and Das [14] proposed a MAF-ACO algorithm, which emulates the foraging behavior of ants found in nature. Uğur and Aydin [15] developed a web-based simulation and analysis software (TSPAntSim) for solving TSP using ACO algorithms with local search heuristics. Zhao *et al.* [16] proposed mutated ant colony optimization (MACO) algorithm by introducing the mutation mechanism to the ACO algorithm, and the MACO algorithm is applied to the traveling salesman problem and multiuser detection. Shuang *et al.* [17] proposed a hybrid PS-ACO algorithm, ACO algorithm modified by particle swarm optimization (PSO) algorithm. Dong *et al.* [18] proposed a new hybrid algorithm, cooperative genetic ant system (CGAS) to deal with the travelling salesman problem. Saenphon *et al.* [19] proposed a new evolutionary optimization algorithm based on the actual manifold of objective function and fast opposite gradient search to improve the accuracy and speed of solution finding. Elloumi *et al.* [20] presented a novel hybrid method (PSO–ACO) approach by introducing a PSO, which is modified by the ACO algorithm to improve the performance. Shima and Hossein [21] proposed a novel feature selection (ABACO) algorithm based on ant colony optimization (ACO) for feature subset selection.

Although the proposed improved ACO algorithm are widely used for solving the traveling salesman problem, they exist some shortcomings for solving complex large-scale optimization problems, such as slow premature convergence, improper parameter values and stagnation phenomenon and so on. So a novel hybrid evolution optimization algorithm, named HEACO algorithm is proposed to solve the traveling salesman problem. Thee chaos is used to introduced into the PSO algorithm to initialize the swarm and adjust the mutation probability and inertia weighting factor. The improved PSO algorithm used to optimize the parameters of the ACO algorithm in order to obtain the HEACO algorithm with higher efficiency and quality, better convergence speed and computation time.

The rest of the paper is organized as follows. Section 2 briefly describes travelling salesman problem. Section 3 briefly introduces the used basic method, including the chaos, particle swarm optimization and ant colony optimization algorithms. Section 4 describes the idea and flow of the hybrid evolution optimization algorithm, named HEACO algorithm in details. Section 5 tests the HEACO algorithm on TSP, and the obtained results are presented and analyzed. Finally, conclusions are given in Section 5.

## 2. Travelling Salesman Problem

The TSP is one which has commanded much attention of mathematicians and computer scientists specifically because it is so easy to describe and so difficult to solve. The problem can simply be stated as: a search for the shortest closed tour that visits each city once and only once. The TSP can be described by using a complete directed graph $G = (N, A)$, where $N$ is a set with $n$ cities, $A$ is a set with the arcs. $D = d_{ij}$ is the distance matrix associated with each $arc\ (i, j) \in A$. The distance matrix $D$ is symmetric or asymmetric. The essence of the TSP is to find out a shortest closed tour visiting each of the $n = |N|$ cities of $G$. The distances between two cities are independent of the direction of traversing the arcs. That is $d_{ij} = d_{ji}$ for each two cities in symmetric TSP and $d_{ij} \neq d_{ji}$ for each two cities in asymmetric TSP. Define the variables:

$$x_{ij} = \begin{cases} 1 & \text{if } \textit{the arc } (i, j) \textit{ is in the tour} \\ 0 & \textit{otherwise} \end{cases} \tag{1}$$

This follows from the fact that the TSP can be formulated by the following generalization of a well known integer program formulation for the classical TSP.

Objective function [39]:

$$z = \min \sum_{i} \sum_{j} d_{ij} x_{ij} \tag{2}$$

The constraints are written as follows:

$$\sum_{i=1}^{n} x_{ij} = 1, \ j = 1,2,3,\cdots,n \tag{3}$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1,2,3,\cdots,n \tag{4}$$

$$x_{ij} \in \{0,1\}, \ i, \ j = 1,2,3,\cdots,n \tag{5}$$

$$\sum_{i,j \in S}^{n} x_{ij} \leq | \ S \ | -1, 2 \leq | \ S \ | \leq \ N - 2 \tag{6}$$

In these formulations, the objective function (1) minimizes the completion time of the one among *m* salesmen consuming the maximum traveling time. Equation (2) computes the traveling time of each salesman. Constraints (3) ensure that each customer is visited exactly once. Constraints (4) ensure that exactly one city is connected with the terminal city of any arc. Constraints (5) ensure that exactly one terminal city shall be connected with it for the departure city of any arc. Constraints (6) indicate that any solutions of branch tracks unable to form a complete route shall be rejected.

## 3. Basic Method

### 3.1. Chaos

Chaos is an interesting nonlinear dynamics, which take on many outstanding properties, such as randomicity, ergodicity, sensibility and so on. So there is a significant interest in constructing optimization methods based on chaotic search routines in the past 20 years. The chaos represents the extremely superior performance in the local search field. The basic idea of chaos is to linearly map chaos variable into the value range of the optimization variable. Then the chaos variable is used to search. The randomicity and ergodicity of the chaos can avoid falling into the local minimum and overcoming the shortcomings of traditional optimization algorithm. The mathematical expression of the one-dimensional Logistic map method is:

$$x(t+1) = \mu x(t)(1 - x(t)) \tag{7}$$

Where the control variable $\mu \in [0,4]$ is the parameter of the Logistic. When control variable $\mu = 4$, Logistic mapping is full mapping on the value of [0,1], and at the same time, the Logistic mapping is full in the chaotic state. That is, the generated sequences under Logistic mapping function (the initial condition $x_0$) are not periodic. While outside this range, the generated sequences must converge to one specific value.

### 3.2. Particle Swarm Optimization Algorithm

The PSO algorithm is a population-based search algorithm based on the simulation of the social behavior of birds within a flock. In PSO algorithm, individuals, referred to as particles, are “flown” through hyper dimensional search space. The particles’ positions within the search space are changed based on the social-psychological tendency of individuals in order to delete the success of other individuals. The changing of one particle within the swarm is influenced by the experience, or knowledge. The consequence of modeling for this social behavior is that the search is processed in order to return toward previously successful regions in the search space. Namely, the velocity($v$ ) and position($x$ ) of each particle will be changed by the particle best value ( $pB$ ) and global best value ( $gB$ ). The velocity and position updating of the particle is shown by the followed expression:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 \left(pB_{ij}(t) - x_{ij}(t)\right) + c_2 r_2 \left(gB_{ij}(t) - x_{ij}(t)\right) \tag{8}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{9}$$

Where $v_{ij}(t+1)$, velocities of particle *i* at iterations *j*, $x_{ij}(t+1)$, positions of particle $i^{th}$ at iterations $j^{th}$. $w$ is inertia weight to be employed to control the impact of the previous history of velocities. Accordingly, the parameter $w$ regulates the trade-off between the global and local exploration abilities of the swarm. The $t$ denotes the iteration number, $c_1$ is the cognition learning factor, $c_2$ is the social learning factor, $r_1$ and $r_2$ are random numbers uniformly distributed in [0-1] which denote remembrance ability for study.

### 3.3. Ant Colony Optimization Algorithm

Ant colony algorithm (ACO) was introduced by Marco Dorigo [12]. The ACO is a metaheuristic inspired by the behaviour of real ants in their search for the shortest path to food sources. Ants tend to choose the paths marked by the strongest pheromone concentration. The ACO algorithm is an essential system based on agents that simulates the natural behaviour of ants, including the mechanisms of cooperation and adaptation. The ACO algorithm simulates the techniques employed by real ants to rapidly establish the shortest route from a food source to their nest and vice versa without the use of visual information. The ACO algorithm consists of a number of cycles (iterations) of solution construction. In each iteration, a number of ants construct complete solutions by using heuristic information and the collected experiences of previous groups of ants. These collected experiences are represented by the pheromone trail which is deposited on the constituent elements of a solution. Pheromone can be deposited on the components and/or the connections used in a solution depending on the problem. The procedure of the ACO algorithm is illustrated in Figure 1.

In the ACO algorithm, we define a list of nodes which the $k^{th}$ ant cannot choose as the next node. This list is called Tabu*k*, which includes all the customer nodes that have been visited by the $k^{th}$ ant until the current state in addition to all the depots except the one which the current tour has been started. Assume that there are $n$ cities and $m$ ants, the initial intensity of pheromone on each edge is set to a very small non-zero positive constant $\tau_0$. In each cycle, each ant starts at a stochastic chosen city, then visits the other cities once and only once according to the transition rule based on the initial intensity of pheromone. The intensity of pheromone will be updated by using the pheromone update rule. The procedure of pheromone update rule is shown as follows:

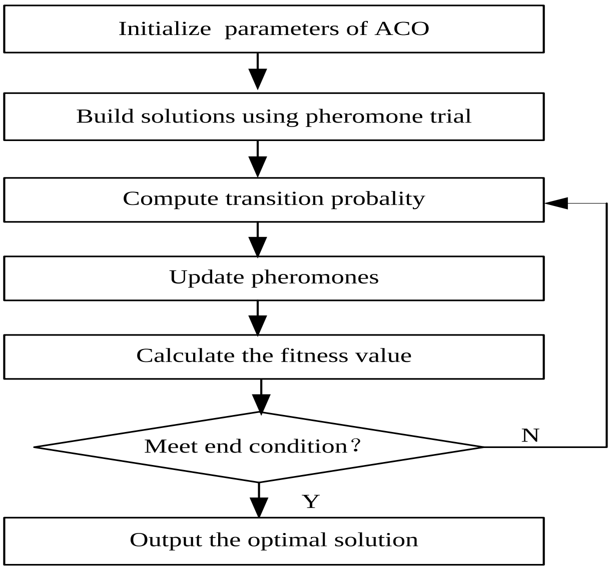

Initialize parameters of ACO
Build solutions using pheromone trial
Compute transition probality
Update pheromones
Calculate the fitness value
Meet end condition?
N
Y
Output the optimal solution

**Figure 1. Searching Procedure of the ACO**

(1) The transition rule

In the route, the $k^{th}$ ant starts from city r, the next city s is selected among the unvisited cities memorized in $J_r^k$ according to the following formula:

$$s = \arg\max_{u \in J_r^k}[\tau_i(r,u)^{\alpha} \cdot \eta(r,u)^{\beta}] \quad if \ q \leq q_0 \,(Exploitation) \tag{4}$$

To visit the next city s with the probability $p_k(r,s)$ ,

$$p_k(r,s) = \begin{cases} \dfrac{\tau(r,s)^{\alpha} \cdot \eta(r,s)^{\beta}}{\sum_{u \in J_r^k} \tau(r,u)^{\alpha} \cdot \eta(r,u)^{\beta}} & if \quad s \in J_r^k \\ 0 & otherwise \end{cases} \quad if \ q > q_0 \,(Bias \ \ Exploitation) \tag{5}$$

(2) The pheromone update rule

In order to improve the solution, the pheromone trails must be updated. Trail updating includes local updating and global updating. The local trail updating formula is given by:

$$\tau(r,u) = (1-\rho)\tau(r,s) + \sum_{k=1}^{m} \Delta\tau_k(r,s) \tag{6}$$

In the formula (6), $\rho$ ($0 < \rho < 1$) is the pheromone trial evaporating rate. $\Delta\tau_k(r,s)$ is the amount of pheromone trail added to the edge(r,s ) by ant *k* between time *t* and t+Δ t in the tour. It is given by:

$$\Delta\tau_k(r,s) = \begin{cases} \dfrac{Q}{L_k} & (\mathrm{r},s) \in \pi_k \\ 0 & otherwise \end{cases} \tag{7}$$

Where $Q$ is a constant parameter, $L_k$ is the distance of the sequence $\pi_k$ toured by ant in Δ *t*.

## 4. A Novel Hybrid Evolution Optimization (HEACO) Algorithm

### 4.1. The Idea of the HEACO Algorithm

The ACO algorithm and PSO algorithm are based on swarm intelligence. The ACO algorithm is good at solving discrete optimization problems, and the PSO algorithm is good at solving continuous optimization problems. In solving TSP, the ACO algorithm is a discrete optimization problem, but the control parameters of the algorithm is continuously changing. The traditional ACO algorithm relies on the ant from the source to the end path. If the parameters of the ACO algorithm are set improperly, the solving speed will be very slow and very poor solution quality. And the parameters of the PSO algorithm will affect the overall optimization performance of this algorithm. The chaos is introduced into the PSO algorithm to initialize the swarm in order to enhance the diversity of the PSO algorithm, and adjust the mutation probability and inertia weighting factor in order to improve the capability of local and global search. Then improved PSO algorithm makes use of the its experience and swarm to optimize the parameters of the ACO algorithm, such as the heuristic factor ( $\beta$ ), pheromone evaporation coefficient ( $\rho$ ), the stochastic selection threshold ( $q$ ). And a novel hybrid evolution optimization (HEACO) algorithm is propose to effectively overcome the parameter influences of ACO, reduce the numbers of useless experiments and balance the developing optimal solution and enlarging search space.

The core idea of the proposed HEACO algorithm is: In a three-dimensional space, the number of particles are initialized, the heuristic factor ( $\beta$ ), pheromone evaporation coefficient ( $\rho$ ), the stochastic selection threshold ( $q$ ) respectively becomes three axis coordinates of the particles. The position of each particle on three-dimensional space is a set of solutions for the parameter optimization problem. The initial particle position feedbacks to AS, then the fitness value of each particle is calculated according to the objective function. Judge whether the position of each particle is optimal. The velocity and position are calculated according to the given formula in one iteration in order to obtain the set of solutions, which is feedbacked to the objective function of the AS in order to reevaluate each particle. Loop to execute the above operation until meeting end condition. The optimal values of the parameters of the ACO algorithm is obtained.

### 4.2. The Flow of the HEACO Algorithm

The flow of the HEACO algorithm is shown in Figure 2.

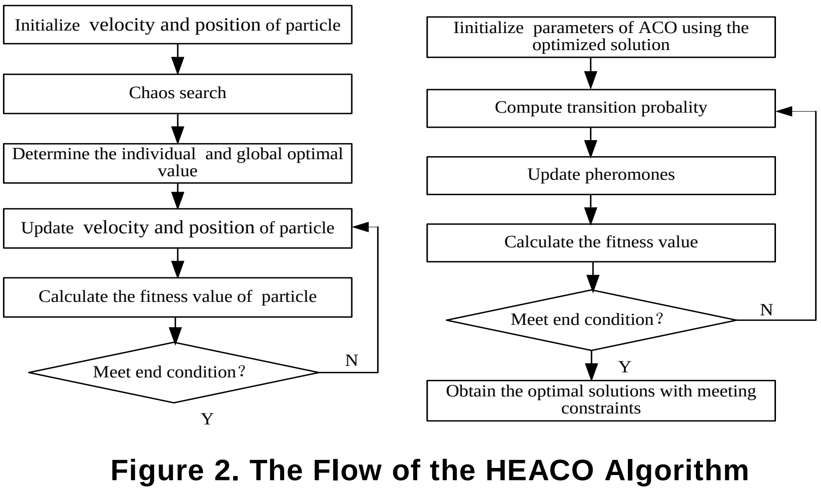

Initialize velocity and position of particle
Chaos search
Determine the individual and global optimal value
Update velocity and position of particle
Calculate the fitness value of particle
Meet end condition？
N
Y
Iinitialize parameters of ACO using the optimized solution
Compute transition probality
Update pheromones
Calculate the fitness value
Meet end condition？
N
Y
Obtain the optimal solutions with meeting constraints

Figure 2. The Flow of the HEACO Algorithm

## 5. Experimental Simulation and Analysis

In order to demonstrate the effectiveness and performance of the proposed HEACO algorithm, we have implemented the proposed HEACO algorithm by using MATLAB 7.8 on a 2.0G Pentium(R) CPU E5200 PC with obtained 12 datasets from TSPLIB (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/). According to TSPLIB, the distance between any two cities is calculated by the Euclidian distance and then rounded off after the decimal point in this paper. The parameters of these algorithms are selected after testing. A number of different alternative values were tested for all instances. The selected ones are those that gave the best computational results concerning both the quality of the solution and the computational time needed to achieve this solution. Thus, the selected parameters for PSO, ACO, P-ACO and HEACO algorithms are shown in Table 1.

**Table 1. The Parameters for PSO, ACO, P-ACO and HEACO Algorithm**

| Parameter | PSO | ACO | P-ACO | HEACO |
|---|---|---|---|---|
| Population size | 80 | 80 | 80 | 80 |
| iteration times | 1000 | 1000 | 1000 | 1000 |
| Maximum velocity | 0.50 | N/A | 0.50 | 0.50 |
| Inertia Weight( $w$ ) | 0.85 | N/A | 0.85 | 0.85 |
| Learn factor | $c_1 = c_2$ =2.0 | N/A | $c_1 = c_2$ =2.0 | $c_1 = c_2$ =2.0 |
| Pheromone Factor( $\alpha$ ) | N/A | 1.0 | 1.0 | 1.0 |
| heuristic factor( $\beta$ ) | N/A | 2.0 | 2.0 | 2.0 |
| Evaporation coefficient( $\rho$ ) | N/A | 0.05 | 0.05 | 0.05 |
| Stochastic selection threshold ( $q_0$ ) | N/A | 0.95 | 0.95 | 0.95 |

In here, the ACO and P-ACO and HEACO algorithm are performed on 12 TSP benchmark instances from TSPLIB with cities scale from 51 to 14051. The number of cities in each instance is the number following the letters that name the instances, for example, in eil51 the number of cities is 51. for example, in eil51 the number of cities is 51. In each time, we evolved 1000 generations. Each algorithm is coded with Matlab language. The comparison to be performed here will take into account the best value, percentage value and error in relation to the best known solutions. The best value denotes the best solution found by each algorithm. The average value denotes the average value of the total run solutions. The error value denotes the percent difference of the solution. All algorithms were run 30 times for each instance. The simulated experiment results are shown in Table 2.

### Table 2. The Results of the Simulated Experiments

| Instances | Optimal Value | ACO | | | P-ACO | | | HEACO | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Best | Avg. | Err.(%) | Best | Avg. | Err.(%) | Best | Avg. | Err.(%) |
| eil51 | 426 | 436.9 | 446.5 | 2.56% | 428.4 | 440.1 | 0.56% | 426 | 432.2 | 0.00% |
| st70 | 675 | 696.8 | 709.2 | 3.23% | 692.2 | 705.8 | 2.55% | 679.3 | 695.5 | 0.64% |
| pr76 | 108159 | 110520 | 114970 | 2.18% | 109851 | 111174 | 1.56% | 109326 | 120412 | 1.08% |
| kroD100 | 21294 | 21841 | 22213 | 2.57% | 21728 | 21996 | 2.04% | 21498 | 21893 | 0.96% |
| pr152 | 73682 | 75365 | 75431 | 2.28% | 74914 | 75317 | 1.67% | 74713 | 74938 | 1.40% |
| rat195 | 2323 | 2398 | 2489 | 3.23% | 2376 | 2423 | 2.28% | 2364 | 2471 | 1.76% |
| gil262 | 2378 | 2457 | 2532 | 3.32% | 2433 | 2498 | 2.31% | 2421 | 2539 | 1.81% |
| pcb442 | 50778 | 52741 | 53967 | 3.87% | 51991 | 52232 | 2.39% | 51717 | 51903 | 1.85% |
| rat783 | 8806 | 9165 | 9245 | 4.08% | 9075 | 9176 | 3.05% | 8978 | 9145 | 1.95% |
| d1291 | 50801 | 52965 | 53789 | 4.26% | 52373 | 53789 | 3.09% | 51873 | 52189 | 2.11% |
| d1655 | 62128 | 65925 | 67012 | 6.11% | 64761 | 65863 | 4.24% | 64151 | 65788 | 3.26% |
| brd14051 | 469385 | 518640 | 519325 | 10.49% | 505737 | 511647 | 7.74% | 498479 | 503603 | 6.20% |

As can be seen from the Table 2, for the 17 TSP instances with the HEACO algorithm, the best value and average value are the best than the ACO algorithm and the P-ACO algorithm for 12 datasets of the traveling salesman problem from TSPLIB. In addition, for TSP instances eil51, the HEACO algorithm can find the best known solutions 426. Particularly, for TSP instances st70, eil76, kroD100, pr152, rat195 and gil262, the new best known solutions 679.3,109326,21498,74713,2364,2421 and 51717 are approaching to the best known solutions 675,108159, 21294, 73682, 2323 and 2378. For larger scale instances, the Table 2. shows that the average results of the proposed HEACO algorithm is better than the best value and average value.

The best routes and their costs (their route lengths) TSP instances att51 and kroD100 are found by using the proposed HEACO algorithm in Figure 3. Note that the way the network grows, like an expanding ring, reduces the possibility of crossings in the routes, which are characteristic of locally optimal routes.

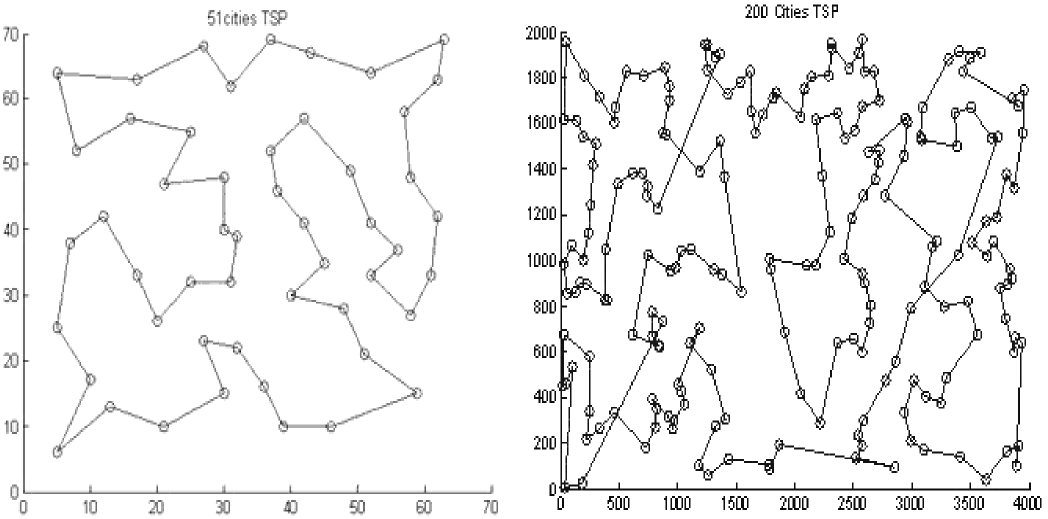


**Figure 3. The Best Routes Found by HEACO Algorithm for att51 and kroD100**

## 6. Conclusion

In this paper, a novel hybrid evolution optimization algorithm based on combining the chaos, particle swarm optimization algorithm and ant colony optimization algorithm is presented, named HEACO algorithm to solve the traveling salesman problem. Because the PSO is provided with the global best experience, the ACO is provided with the procedure of converging to the global optimum, the best solution and average solution quality is improved by using the cooperative evolution scheme of the chaos, PSO algorithm and ACO algorithm. As a result of the experiments we can see that the best solution quality and average solution quality of the HEACO algorithm is really better than those of the ACO and P-ACO algorithm respectively by testing the 12 data sets obtained from the TSPLIB with cities scale from 51 to 14051, and have compared the experimental results of the proposed HEACO method with the ACO and P-ACO algorithm. As a whole, it has also been shown that the proposed HEACO algorithm can solve larger scale optimization problems than those solved using the existing algorithms. In the future, I hope that the proposed HEACO algorithm can be applied to other practical problems, such as the vehicle routing problem, the job-shop scheduling problem, data transmission and design an optimization in the mechanical Engineering.

## Acknowledgments

This research was supported by Zhejiang Provincial Natural Science Foundation of China (LQ12E01003) and Zhejiang Provincial Technical Plan Project (2012C21099).

## References

[1] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem”, IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, **(1997)**, pp. 53-66.

[2] Y. Wang, “The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem”, Computers & Industrial Engineering, vol. 70, **(2014)**, pp. 1240-133.

[3] Z. Ardalan, S. Karimi, O. Poursabzi and B. Naderi, “A novel imperialist competitive algorithm for generalized traveling salesman problems”, Applied Soft Computing, vol. 26, no. 1, **(2015)**, pp. 546-555.

[4] S. C. Sarin, H. D. Sherali, J. D. Judd and P. F. Tsai, "Multiple asymmetric traveling salesmen problem with and without precedence constraints: Performance comparison of alternative formulations", Computers & Operations Research, vol. 51, no. 11, **(2014)**, pp. 64-89.

[5] K. Florios and G. Mavrotas, "Generation of the exact Pareto set in Multi-Objective Traveling Salesman and Set Covering Problems", Applied Mathematics and Computation, vol. 237, **(2014)**, pp. 1-19.

[6] Z. Hua and F. Huang, "A variable-grouping based genetic algorithm for large-scale integer programming, Information Sciences, vol. 176, no. 19, **(2005)**, pp. 2869-2885.

[7] I. Ellabib, P. Calamai and O. Basir, "Exchange strategies for multiple ant colony system", Information Sciences, vol. 177, no. 5, **(2007)**, pp. 1248-1264.

[8] C. C. Lo and C. C. Hus, "Annealing framework with learning memory", IEEE Transactions on System, Man, Cybernetics, Part A, vol. 28, no. 5, **(1998)**, pp. 1-13.

[9] Y. Marinakis, M. Marinaki and G. Dounias, "A hybrid particle swarm optimization algorithm for the vehicle routing problem", Engineering Applications of Artificial Intelligence, vol. 23, no. 4, **(2010)**, pp. 463-472.

[10] T. A. Masutti and L. N. de Castro, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem", Information Sciences, vol. 179, no. 10, **(2009)**, pp. 1454-1468.

[11] G. Shen and Y. Q. Zhang, "A new evolutionary algorithm using shadow price guided operators", Applied Soft Computing, vol. 11, no. 2, **(2011)**, pp. 1454-1468.

[12] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem", Biosystems, vol. 43, no. 2, **(1997)**, pp. 73-81.

[13] Z. L. Wu, N. Zhao, G. H. Ren and T. F. Quan, "Population declining ant colony optimization algorithm and its applications", Expert Systems with Applications, vol. 36, no. 3, **(2009)**, pp. 6276-6281.

[14] V. S. Borkar and D. Das, "A novel ACO algorithm for optimization via reinforcement and initial bias", Swarm Intelligence, vol. 3, no. 1, **(2009)**, pp. 3-34.

[15] A. Uğur and D. Aydin, "An interactive simulation and analysis software for solving TSP using ant colony optimization algorithms", Advances in Engineering Software, vol. 40, no. 5, **(2009)**, pp. 341-349.

[16] N. Zhao, Z. L. Wu, Y. Q. Zhao and T. F. Quan, "Ant colony optimization algorithm with mutation mechanism and its applications", Expert Systems with Applications, vol. 37, no. 7, **(2010)**, pp. 4805-4810.

[17] B. Shuang, J. P. Chen and Z. B. Li, "Study on hybrid PS-ACO algorithm", Applied Intelligence, vol. 34, no. 1, **(2011)**, pp. 64-73.

[18] G. F. Dong, W. W. Guo and K. Tickle, "Solving the traveling salesman problem using cooperative genetic ant systems", Expert Systems with Applications, vol. 39, no. 5, **(2012)**, pp. 5006-5011.

[19] Saenphon, S. Phimoltares and C. Lursinsap, "Combining new fast opposite gradient search with ant colony optimization for solving travelling salesman problem", Engineering Applications of Artificial Intelligence, vol. 35, no. 1, **(2014)**, pp. 324-334.

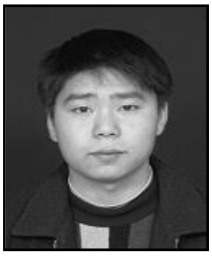
[20] W. Elloumi, H. E. Abed, A. Abraham and A. M. Alimi, "A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP", Applied Soft Computing, vol. 25, no. 12, **(2014)**, pp. 234-241.

[21] K. Shima and N. Hossein, "An advanced ACO algorithm for feature subset selection", Neurocomputing, vol. 147, no. 5, **(2015)**, pp. 271-279.

## Authors

**Bin Gao**, Associate Researcher, received the Doctor degree in mechanical engineering from Zhejiang University in 2006, Hangzhou, China. The main research directions: optimization design, CAE, physical vapor deposition.

**Jing-Hua Zhu**, Lecturer, received the Engineering Master degree in computer software from University of Electronic Science and Technology of China in 2011, Chengdu, China. The main research directions: CAD&CG.



**Wen-chang Lang**, Assistant Researcher, received the Doctor degree in materials processing engineering from Chinese Academy of Sciences in 2010, Shenyang, China. The main research directions: physical vapor deposition.