Journal of Systems Engineering and Electronics Vol. 21, No. 2, April 2010, pp.329–333
Available online at www.jseepub.com

Improved ant colony optimization algorithm for the traveling salesman problems

Rongwei Gan[1], Qingshun Guo[2], Huiyou Chang[1], and Yang Yi[1,*]

1. School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275, P. R. China;
2. Information and Network Center, Sun Yat-sen University, Guangzhou 510275, P. R. China

Abstract: Ant colony optimization (ACO) is a new heuristic algorithm which has been proven a successful technique and applied to a number of combinatorial optimization problems. The traveling salesman problem (TSP) is among the most important combinatorial problems. An ACO algorithm based on scout characteristic is proposed for solving the stagnation behavior and premature convergence problem of the basic ACO algorithm on TSP. The main idea is to partition artificial ants into two groups: scout ants and common ants. The common ants work according to the search manner of basic ant colony algorithm, but scout ants have some differences from common ants, they calculate each route's mutation probability of the current optimal solution using path evaluation model and search around the optimal solution according to the mutation probability. Simulation on TSP shows that the improved algorithm has high efficiency and robustness.

Keywords: ant colony optimization, heuristic algorithm, scout ants, path evaluation model, traveling salesman problem.

DOI: 10.3969/j.issn.1004-4132.2010.02.025

1. Introduction

Swarm intelligence (SI) is artificial intelligence based on the collective behavior of decentralized and self-organized systems. SI systems have the characteristic that an individual follows simple rules, but interactions between individuals leads to the emergence of complex global behavior. Classical SI system is ant colony. A colony can solve problems unthinkable for individual ant, such as finding the shortest path from nest to food source, and allocating workers to different tasks. A new heuristic algorithm named ant colony optimization (ACO) has been proposed through studying foraging behavior of real ants and applied to solving combinatorial optimization problems such as traveling salesman problem (TSP) [1], assignment problem [2], and job-shop scheduling problem [3].

ACO has received increased interests from researchers in recent years and a relatively large amount of successful applications are now available [4–15]. The MAX-MIN ant system (MMAS) and local search for TSP are proposed [4]. Reference [5] developed a computationally efficient vector optimizer using ACO algorithm for multiobjective designs. Reference [6] designed an ant algorithm for balanced job scheduling in grids. Reference [7] studied a formulation of ant algorithms for the water distribution system optimization. Reference [8] proposed an immunity-based ACO algorithm for solving weapon–target assignment (WTA) problems.

However, as other heuristic algorithms, ACO suffers from some problems. In this paper, an ACO algorithm with scout characteristic (SCACO) is proposed for solving the stagnation behavior and premature convergence problem of the basic ACO algorithm on TSP. The new algorithm partitions artificial ants into two groups: scout ants and common ants. The common ants work according to the searching model of basic ant colony algorithm: ants randomly placed at the cities move to the next city using the city state transition rule, and finish a construction after all the cities are selected. The scout ants have some difference from common ants, they donot randomly choose the start city but calculate each route's mutation probability of the current optimal solution using path evaluation model, search around the optimal solution according to the mutation probability, shorten the time ever cost in city conversion during each construction of basic ACO, exert maximally the leading function of the current optimal solution, accelerate the convergence and simultaneously avoid from local solution. Simulation on TSP shows that the improved algorithm has high efficiency and robustness.

2. Ant colony optimization for TSPs

TSP can be represented by a completely weighted directed graph $G = (V, A, d)$, where $V = \{1, 2, \ldots, n\}$ is the city set, $A = \{(i, j)|(i, j) \in V * V\}$ is the arc set, d is a weighted function associating a positive integer with every

Manuscript received December 9, 2008.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (60573159).

arc (i, j). The aim of TSP is to find a route of minimal length with every city visited just once. Now, TSP becomes a standard test for heuristic algorithms which find the optimal solutions [4]. So the ACO can be extended to the TSP.

ACO algorithm is a heuristic algorithm to model the behavior of real ant colonies in establishing the shortest path between food sources and nests. As an ant moves, it releases a trail pheromone that can be detected by other ants. As more ants pass by the path, more pheromones are deposited. Because ants move according to the amount 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. In the following, we will provide a formal definition of the basic ACO.

Initially, each ant is randomly put on a city. During the construction of a feasible solution, ants select the following city to be visited through a probabilistic decision rule. When ant k states in city i and constructs the partial solution, the probability moving to the next city j neighboring on city i is given by

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{k \notin tabu} \tau_{ik}^{\alpha}(t)\eta_{ik}^{\beta}(t)}, & j \notin tabu_k \\ 0, & j \in tabu_k \end{cases} \tag{1}$$

where $\tau_{ij}(t)$ is the amount of pheromone trail on arc (i, j) at time t, $\eta_{ij}(t) = 1/d_{ij}$ is the heuristic value of moving from city i to city j, $tabu_k$ is the visited city set of ant k, α and β are two parameters that control the relative weight of pheromone trail and heuristic value.

The amount of pheromone trail on a path evaporates step by step. After time m, the trail intensity is updated according to the following formula

$$\tau_{ij}(t+m) = \rho\tau_{ij}(t) + \Delta\tau_{ij}, \quad \rho \in (0,1) \tag{2}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{n} \Delta\tau_{ij}^k \tag{3}$$

where ρ is a coefficient which represents the evaporation of trail between time t and $t+m$, $\Delta\tau_{ij}^k$ is the quantity per unit of length of trail substance (pheromone in real ants) laid on edge (i, j) by the ant k between time t and $t+m$, n is the number of ants.

An ant-cycle system information update model is

$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{L_k}, & \text{arc } (i,j) \text{ belongs to best tour} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Reference [4] proposed the MAX-MIN ant system [4], which is the same as the ant system, but trails are initialized to their maximum value $\tau_{\max}$, pheromone trail values are restricted to an interval $[\tau_{\min}, \tau_{\max}]$, and only the best ant updates the trails.

3. ACO with scout characteristic

3.1 Path evaluation model

Given a random sample $(X_1, X_2, \ldots, X_n)$ from an n-dimensional random variable, the sample mean is

$$\overline{X} = \frac{\sum\limits_{i=1}^{n} X_i}{n} \tag{5}$$

The sample variance s^2 is the second sample central moment and is defined by

$$s = \sqrt{\frac{1}{n}\left[\sum_{i=1}^{n}(X_i - \overline{X})^2\right]} \tag{6}$$

The goal of the path evaluation model is to find an improvement space of each path in optimal solution. The improvement space is calculated according to the following formula

$$S_{ij} = \sqrt{\frac{1}{m+n}\left(\sum_{s=1}^{n} w_{is}|d_{is} - d_{ij}|^r + \sum_{r=1}^{m} w_{jr}|d_{jr} - d_{ij}|^r\right)}$$
$$\forall d_{is} \leqslant d_{ij} \text{ and } \forall d_{jr} \leqslant d_{ij} \tag{7}$$

where d_{is} is the length of arc (i, s), n is the corresponding city number, d_{jr} is the length of arc (j, r), m is the corresponding city number, w_{is} is the weight of arc (i, s), w_{jr} is the weight of arc (j, r). w_{is} and w_{jr} are decided by rank sum, and r is the adjustment parameter. We can control the differences of path evaluation through setting a different value to r.

For example, given a weighted directed graph and an optimal solution: 1->2->3->4->1, $r = 2$, the rank of each path is as follows.

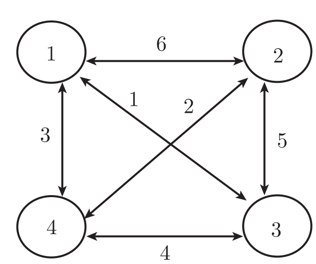


Fig. 1 An example of counting weight

1->2: 3{arc (1,4)} 5{arc (1,3)} rank: 3 5

2->3: 3{arc (2,4)} rank: 3

3->4: 3{arc (1,3)} rank: 3

4->1: 1{arc (2,4)} rank: 1

The evaluation value of each path is

1->2: $\sqrt{0.5*(0.6*9+1*25)} \approx 3.9$

2->3: $\sqrt{0.6*9} \approx 2.3$

3->4: $\sqrt{0.6*9} \approx 2.3$

4->1: $\sqrt{0.2*1} \approx 0.45$

After getting the evaluation value of each path in optimal solution, the scout ants calculate the mutation probability as follows

$$M_{ij} = \frac{S_{ij}}{\sum S_{lr}} \tag{8}$$

where n is the number of city, S_{ij} and S_{lr} are the paths in optimal solution.

3.2 Scout ants searching schematic

In basic ant colony optimization, ants randomly choose the start city at each step of the solution construction and select the following city to be visited according to the probabilistic decision rule. However the scout ants have some difference from common ants, they search around the current optimal solution. At some time, they will directly visit the following city in optimal solution but using a stochastic mechanism. When ant k states in city i, and arc (i, j) is included by the current optimal solution, the probability of moving to the neighbor $r(r \neq j)$ of city i is given by

$$p_{ir}^{k} = \begin{cases} \dfrac{\tau_{ir}^{\alpha}(t)\eta_{ir}^{\beta}(t)}{\sum\limits_{l \notin tabu} \tau_{il}^{\alpha}(t)\eta_{il}^{\beta}(t)}, \begin{pmatrix} r \notin tabu \\ \cap p < M_{ir} \cap d_{ir} \leqslant d_{ij} \end{pmatrix} \\ 0, (r \in tabu) \cup (r \notin tabu \cap (p \geqslant M_{ir})) \end{cases} \tag{9}$$

where M_{ir} is the mutation probability of arc (i, j), p is the random number and $p \in (0, 1]$.

As shown in Fig. 2, suppose the arc (A, B) is one of the paths in optimal solution. When a scout ant states in city A, according to (9), given a random number p, if $p \geqslant M_{ab}$, then the scout ant chooses the city B as the following city; if $p < M_{ab}$, then the scout ant calculates the moving choice probability of unvisited cities and selects the following city whose distance is shorter than d_{ab}.

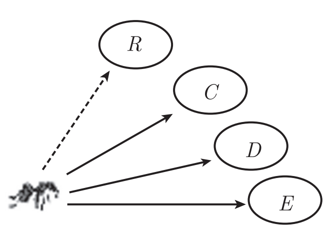


Fig. 2 Ants searching schematic

3.3 SCACO for TSP

The methodology of ant colony algorithm with scout characteristic can be described as follows

Procedure: SCACO

Input: ACO parameters, TSP

Output: Optimal solution

INIT. $\alpha, \beta, Q, \rho, \tau_{\max}, \tau_{\min}, tabu$, iteration $N_{\max}$, the ant number m, scout ants rate $ratio$, adjustment index r, randomly produce an optimal solution as the global optimum G;

While (NOT satisfy stop condition){
  Calculate the mutation probabilities according to (8) and (9);
 For search ants {
  Randomly choose the start city;
  While (NOT all city visited) {
   Calculate the following city according to (2);
  }
 }
 For scout ants {
  Select a path from optimal solution;
  While (NOT all city visited) {
   Calculate the following city according to (9);
  }
 }
 Locally optimize using 2-opt or 3-opt;
 Update the trails of local optimal solution L;
  IF $(G > L)$ {
   Set L to G;
  }
 }
 Output the optimal solution G.

4. Computational results

Two TSPs from the TSPLIB (http://www.iwr.uni-heidelberg.de/groups/como-pt/ software/TSPLIB95/) are simulated.

The maximum iteration is set to 100, pheromone trail to 4, heuristic value to 5, evaporation coefficient to 0.95, scout rate is 0.25, and the adjustment index is 0.8. In order to make the results more readable and easier to do comparison, all the data in Table 1 are processed. The optimal solution value 15 780 is treated as the cardinal number 0, data 8 represents 15 788, and so on.

In order to make the difference more obvious and direct, the results distribution (sort the data ascending) of both algorithms is shown in Fig. 3.

Table 1 Results of d198

MMAS +2-opt	0	8	14	19	16	1	7	9	2	1
	31	16	15	14	6	0	20	3	8	7
SCACO +2-opt	1	0	2	1	1	1	1	8	1	0
	8	8	1	1	15	21	0	2	1	3

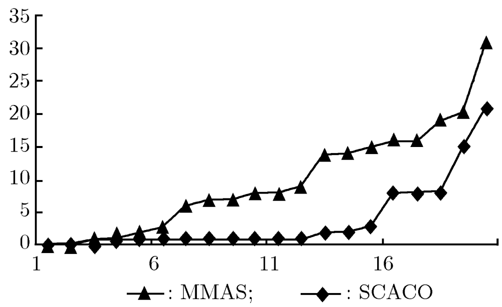


Fig. 3 Results distribution of MMAS and SCACO

Comparing the optimal solutions, the worst solutions, the average solutions, and the standard deviations (see Table 2) of the two algorithms in Table 1, it is known that although both algorithms have attained the optimal solutions of the problem, the worst solution of scout algorithm is only 15 801, the average solution is 15 783.80, the standard deviation is 5.6, whereas the worst solution of MMAS algorithm is 15 810, the average solution is 15 789.80, the standard deviation is 8.17.

Table 2 Analysis results of d198

Algorithm	Best	Worst	Means	Deviation
MMAS+2-opt	15 780	15 810	15 789.80	8.17
SCACO+2-opt	15 780	15 801	15 783.80	5.6

Meanwhile, we use the Z test to find the difference between two groups of data. The Z test is a statistical test using Z values to compare differences in proportions among sets of data or among individual members of different sets of data.

Table 3 gives the rank of data, the rank sum of MMAS is 502, and the one of SCACO is 318.

Table 3 Ranks of d198

MMAS rank	3	27	31.5	37	35.5	11	23.5
	30	18	11	40	35.5	33.5	31.5
	22	3	38	20.5	27	23.5	
SCACO rank	11	3	18	11	11	11	11
	27	11	3	27	27	11	11
	33.5	39	3	18	11	20.5	

H_0: There is no difference. Both groups are from the same population.

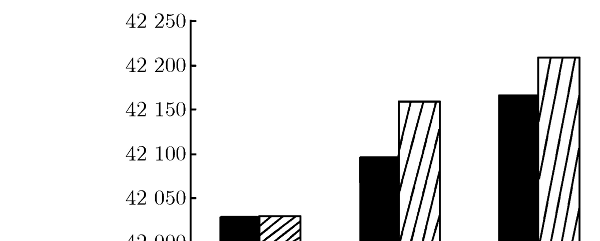
H_1: There is a significant difference between the two groups. They belong to two different populations.

$$n_1 = n_2 = 20, \quad T = 318$$

$$Z = \frac{T - \dfrac{n_1(n_1+n_2+1)}{2}}{\sqrt{\dfrac{n_1 n_2 (n_1+n_2+1)}{12}}} = \frac{318-410}{36.968\,5} \approx -2.5$$

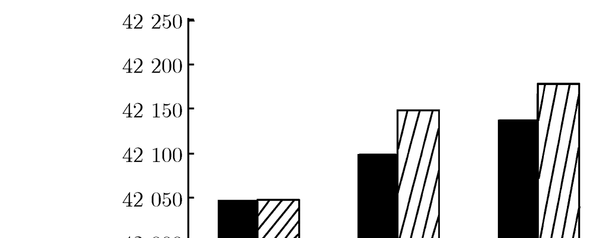
In this case, the difference between the means is a z-score of 2.5. We can conclude with 95 percent certainty that the two studied samples come from different populations. So we should reject the H_0 hypothesis, conclude that these two samples come from two different populations, and that the difference between them is significant. In other words, SCACO has more efficiency and robustness than MMAS.

Figs. 4 and 5 give the test results for lin318. 1 is the best solution, 2 is the average solution and 3 is the worst solution. The maximum iteration is set to 200, pheromone trail to 4, heuristic value to 5, evaporation coefficient to 0.6, scout rate is 0.25, and the adjustment index is 1.2.



	1	2	3
SCACO+2-opt	42 029	42 096.4	42 166
MMAS+2-opt	42 029	42 157.8	42 208

Fig. 4 Comparison of MMAS+2-opt and SCACO+2-opt



	1	2	3
SCACO+3-opt	42 029	42 059.7	42 083
MMAS+3-opt	42 029	42 089.6	42 107

Fig. 5 Comparison of MMAS+3-opt and SCACO+3-opt

Both algorithms have attained the optimal solutions of the problem, whereas the worst solution and average solution of MMAS are higher than that of SCACO.

The experiment shows that the ant colony algorithm with scout characteristic proposed in this paper attained the results for TSPs, its stability and efficiency of solution are higher than that of MMAS algorithm, which illuminates the improvement to the basic ant colony algorithm is effective.

5. Conclusions

An ant colony algorithm with scout characteristic is proposed in this paper. The concept of scout ant colony is introduced, and path evaluation model is presented. Scout ants shorten the search time and effectively retain the characteristics of the optimal solution, accelerate the convergence pace of the algorithm, and ensure the algorithm to go on towards the direction of optimal solution. The final experiment results indicate that the new algorithm has good capability to find optimal solution. Our future work will focus on evaluating the effectiveness of the path evaluation model theoretically.

The contributions of this paper are concluded as follows:

Analyze the shortcomings of basic ACO.

Partition the ants into two groups and introduce the concept of scout ant colony.

Give a path evaluation model for scout ants searching.

An improved ant colony algorithm is discussed and provided.

Experiments testify the efficiency of the presented algorithm.

References

[1] M. Dorigo, L. M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evolutionary Computation*, 1997, 1(1): 53–66.

[2] L. M. Gambardella, E. D. Taillard, M. Dorigo. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, 1999, 50(2): 167–176.

[3] A. Colorni, M. Dorigo, V. Maniezzo, et al. Ant system for job-shop scheduling. *Belgian Journal of Operations Research and Statistic Computing Science*, 1994, 34(1): 39–53.

[4] T. Stützle, H. Hoos. The MAX-MIN ant system and local search for the traveling salesman problem. *Proc. of IEEE 4th International Conference on Evolutionary Computation*, 1997: 308–313.

[5] S. L. Ho, S. Y. Yang. A computationally efficient vector optimizer using ant colony optimizations algorithm for multobjective designs. *IEEE Trans. on Magnetics*, 2008, 44(6): 1034–1037.

[6] R. S. Chang, J. S. Chang, P. S. Lin. An ant algorithm for balanced job scheduling in grids. *Future Generation Computer System*, 2009, 25(1): 20–27.

[7] A. C. Zecchin, A. R. Simpson, H. R. Maier, et al. Parametric study for an ant algorithm applied to water distribution system optimization. *IEEE Trans. on Evolutionary Computation*, 2005, 9(2): 175–191.

[8] Z. J. Lee, C. Y. Lee, S. F. Su. An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem. *Applied Soft Computing*, 2002, 2(1): 39–47.

[9] F. Chicano, E. Alba. Ant colony optimization with partial order reduction for discovering safety property violations in concurrent models. *Information Processing Letters*, 2008, 106(7): 221–231.

[10] B. J. Zhao, S. Y. Li. Ant colony optimization algorithm and its application to neuro-fuzzy controller design. *Journal of Systems Engineering and Electronics*, 2007, 18(3): 603–610.

[11] S. U. Seckiner, M. Kurt. Ant colony optimization for the job rotation scheduling problem. *Applied Mathematics and Computation*, 2008, 201(1/2): 149–160.

[12] S. C. Chu, J. F. Roddick, J. S. Pan. Ant colony system with communication strategies. *Information Science*, 2004, 167: 63–76.

[13] G. M. Kurnar, A. N. Haq. Hybrid genetic-ant colony algorithms for solving aggregate production plan. *Journal of Advanced Manufacturing Systems*, 2005, 4(1): 103–111.

[14] B. R. Hong, F. H. Jin, Q. Guo. Hopfield neural network based on ant system. *Journal of Harbin Institute of Technology*: *New Series*, 2004, 11(3): 267–269.

[15] A. Badr, A. Fahmy. A proof of convergence for ant algorithms. *Information Sciences of Elsevier*, 2004, 160(1/2/ 3/4): 267–279.

Rongwei Gan was born in 1982. He is now pursing the Ph.D. in the School of Information Science and Technology of Sun Yat-sen University. His research interests include evolutionary computation algorithm and combination auction.
E-mail: ganrongwei@tom.com

Qingshun Guo was born in 1959. He is a doctor and associate professor at the Department of Computer Science of Sun Yat-sen University. His research interests are enterprise information and combination acution.
E-mail: gqs@mail.sysu.edu.cn

Huiyou Chang was born in 1962. He is a doctor and professor at the Department of Computer Science of Sun Yat-sen University. His research interests are artificial optimization & intelligent algorithm, data mining and workflow.
E-mail: isschy@mail.sysu.edu.cn

Yang Yi was born in 1967. She is a doctor and associate professor at the Department of Computer Science of Sun Yat-sen University. Her research interests include complex system modeling and optimization, and evolutionary computation algorithm.
E-mail: issyy@mail.sysu.edu.cn