International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.12 (2015), pp.153-164
http://dx.doi.org/10.14257/ijmue.2015.10.12.16

An Improved Ant Colony Optimization Algorithm for Solving TSP

Yimeng Yue[1] and Xin Wang[2]

[1]*College of Mathematics and Computer Application, Shangluo University, Shangluo 726000 China*

[2]*College of Economics and Management, Shangluo University, Shangluo 726000 China*

Abstract

The basic ant colony optimization (ACO) algorithm takes on a longer computing time in the search process and is prone to fall into local optimal solutions, an improved ACO (CEULACO) algorithm is proposed in this paper. In the CEULAC algorithm, the direction guidance and real variable function are used to initialize pheromone concentration according to the path information of undirected graph. The pheromone dynamic evaporation rate strategy is proposed to control the pheromone evaporation in order to increase the global search capability and accelerate the convergence speed. An adaptive dynamic factor is introduced into pheromone updating rule to control the updating proportion of pheromone concentration with optimal solution in single iteration. And the local search strategy is used to improve the quality of the solution and select the current optimal path for global updating the pheromone in order to save some computing time and not reduce the searching efficiency. Some traveling salesman problems are selected to test the performance of the CEULACO algorithm. The simulation experiments show that the improved ACO algorithm can effectively improve the accuracy and the quality of solutions, and distinctly speed up the convergence speed and computing time.

Keywords: *Ant colony optimization; pheromone; evaporation rate; direction guidance; global search; TSP*

1. Introduction

In nature, the individual behaviors of the most gregarious animals are very simple and random, such as birds, fish and ant colony and so on. However, they are similar to foraging and resisting the enemies. These populations show the intelligence, called swarm intelligence [1]. Many scholars proposed a series of new algorithms to solve the difficult optimization problems based on simulating the complex behavior of these social animals, such as genetic algorithm(GA),immune algorithm(IA),neural network(NN), ant colony optimization(ACO) algorithm, particle swarm optimization(ACO) algorithm and so on [2-5]. These algorithms are collectively called as swarm intelligence algorithm.

The ACO algorithm is a simulated evolutionary algorithm based on ant colony foraging. It is proposed by Dorigo in 1991 to achieve good results [6]. Many scholars have also got a lot of research results in this algorithm. In fourteen years, the research shows that the ACO algorithm can not only search global optimal solution, but also take on characteristics of robust, positive feedback, distributed computing, easy fusing with other algorithms. The positive feedback can speed up the evolution process, the distributed computing can make the algorithm to easily implement in parallel. The ACO algorithm is easy to combine with other algorithms in order to improve the performance of algorithm. Because of the robustness, the basic ACO algorithm is slightly modified, they can be applied to solving other optimization problems. So the ACO algorithm is a powerful tool to solve complex

ISSN: 1975-0080 IJMUE

optimization problems in traveling salesman problem, robot path planning, quadratic assignment, continuous function optimization and so on. But the ACO algorithm is easy to fall into local optimum, a long search time and so on.

For these shortcomings, many scholars have studied the improved methods. Zou *et al*., [7] proposed an improved ant colony algorithm (ACA) based on evolutionary strategy with searching in variable neighborhood parameter. Jalali *et al*., [8] proposed an improved ant colony optimization algorithm for reservoir operation. Shelokar *et al*., [9] proposed PSACO (particle swarm ant colony optimization) algorithm for highly non-convex optimization problems. Both particle swarm optimization (PSO) and ant colony optimization (ACO) are co-operative, population-based global search swarm intelligence metaheuristics. Gao *et al*., [10] proposed a hybrid clonal selection and ant colony optimization (CSA-ACO) reasonably by utilizing the superiorities of both algorithms and also overcoming their inherent disadvantages. Simulation results based on the traveling salesman problems have demonstrated the merit of the proposed algorithm over some traditional techniques. Foong *et al*., [11] proposed an improved ant colony optimization-power plant maintenance scheduling optimization (ACO-PPMSO) formulation based on considering such options in the optimization process. As a result, both the optimum commencement time and the optimum outage duration are determined for each of the maintenance tasks that need to be scheduled. Baskan *et al*., [12] proposed an improved solution algorithm using ant colony optimization (ACO) for finding global optimum for any given test functions. β is proposed for improving ACO's solution performance to reach global optimum fairly quickly. Yu *et al*., [13] proposed an improved ant colony optimization (IACO), which possesses a new strategy to update the increased pheromone, called ant-weight strategy, and a mutation operation, to solve vehicle routing problem (VRP). Kaveh and Talatahari [14] proposed an improved ant colony optimization (IACO) for constrained engineering design problems. IACO has the capacity to handle continuous and discrete problems by using sub-optimization mechanism (SOM). Santos *et al*., [15] proposed an improved ant colony optimization based algorithm for the capacitated arc routing problem. The new metaheuristic was tested on seven standard test networks for the capacitated arc routing problem. Yang and Zhuang [16] proposed an improved ant colony optimization algorithm (IACO) for solving mobile agent routing problem. The ants cooperate using an indirect form of communication mediated by pheromone trails of scent and find the best solution to their tasks guided by both information (exploitation) which has been acquired and search (exploration) of the new route. Gan *et al*., [17] proposed an ACO algorithm based on scout characteristic 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. Duan and Liao [18] proposed an ACO technique based on AoN networks. All six algorithms were tested with several benchmark problems. The test results strongly indicate that AoN-based ACO algorithms are more effective and efficient in finding critical paths than AoA-based algorithms. Yu *et al*., [19] proposed an improved ant colony optimization with coarse-grain parallel strategy, ant-weight strategy and mutation operation for solving the multi-depot vehicle routing problem with the virtual central depot (V-MDVRP). Geng *et al*., [20] proposed a directional ant colony optimization (DACO) algorithm for solving nonlinear resource-leveling problems. Jovanovic and Tuba [21] proposed a pheromone correction heuristic strategy that uses information about the best-found solution to exclude suspicious elements from it. This hybridization improves pure ant colony optimization algorithm by avoiding early trapping in local convergence. Ding *et al*., [22] proposed a hybrid ant colony optimization (HACO). This paper gives a guideline on how to adjust the parameters to achieve the good performance of the algorithm. Deng *et al*., [23] proposed a novel two-stage hybrid swarm intelligence optimization algorithm called GA-PSO-ACO algorithm that combines the evolution ideas of the genetic algorithms, particle swarm optimization and ant colony optimization based on the compensation for solving the traveling salesman problem. Cheng *et al*., [24] proposed an improved ant colony optimization for scheduling identical parallel batching machines with arbitrary job sizes. The computational results show the effectiveness of the algorithm, especially for large-scale instances. Wang *et al*., [25] proposed a modified multi-objective ant colony algorithm, in which a reachability matrix is set up to constrain the feasible search nodes of the ants and a new pseudo-random-proportional rule and a pheromone adjustment mechinism are constructed to balance conflicts between the optimization objectives. Zheng *et al*., [26] proposed an improved ant colony optimization. A novel pheromone between two adjacent tasks in the same side station is defined to describe the order relation between them. A new bound strategy is proposed to reduce the search space of ants. Guo and Diao [27] proposed a kind of improved ant colony algorithm with crossover operator which makes crossover operator among better results at the end of each iteration. Luo *et al*., [28] proposed an improved intelligent ant colony algorithm to solve the torsion bar optimization problem. Zhang *et al*., [29] proposed two improvements based on ant colony optimization meta-heuristic constraint solving algorithm. Tang *et al*., [30] proposed an improved ant colony optimization (IACO) for solving the mathematical model for multi-depot heterogeneous vehicle routing problem with soft time windows (MDHVRPSTW). Pang *et al*., [31] proposed an improved ant colony optimization algorithm. The new heuristic information and the improved pheromone update rules are defined. Deng *et al*., [32] proposed an improved CACO (CACOAMS) algorithm based on multi-population strategy, the neighborhood comprehensive learning strategy, the fine search strategy, the chaotic optimization strategy, the super excellent ant strategy, the punishment strategy and min–max ant strategy. Jiang *et al*., [33] proposed a co-evolutionary improved multi-ant colony optimization (CIMACO) algorithm for ship multi and branch pipe route design. The purpose of CIMACO algorithm is to design appropriate pipe routes to connect the starting points and ending points in the layout space under various kinds of constraints.

In this paper, for the limitations of the classical ACO algorithm, a new improved ACO (CEULACO) algorithm is proposed. The direction guidance and real variable function, the pheromone dynamic evaporation rate strategy, the local search strategy are used to improve the ACO algorithm in order to enhance the global search capability and further improve the performance of the solution. The TSP is used to prove the effectiveness and practicality of the proposed CEULACO algorithm.

The rest of this paper is organized as follows. Section 2 briefly introduces the ACO algorithm. Section 3 proposed an improved ACO algorithm based on the direction guidance, real variable function, the pheromone dynamic evaporation rate strategy and the local search strategy. Section 4 gives the steps of CEULACO algorithm. Section 5 briefly introduces the TSP. Section 6 applies the CEULACO algorithm to solve TSP. Finally, the conclusions are discussed in Section 7.

2. ACO Algorithm

Ant colony algorithm (ACO) was introduced by Dorigo [6]. The ACO is a metaheuristic inspired by the behavior of real ants in their search for the shortest path to food. 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. Small quantities are deposited during the construction phase while larger amounts are deposited at the end of each iteration in proportion to solution quality. Pheromone can be deposited on the components and/or the connections used in a solution depending on the problem.

In the ACO algorithm, the ACO algorithm simulates the optimization of ant foraging behavior. The procedure of the ACO algorithm is illustrated in Figure1.

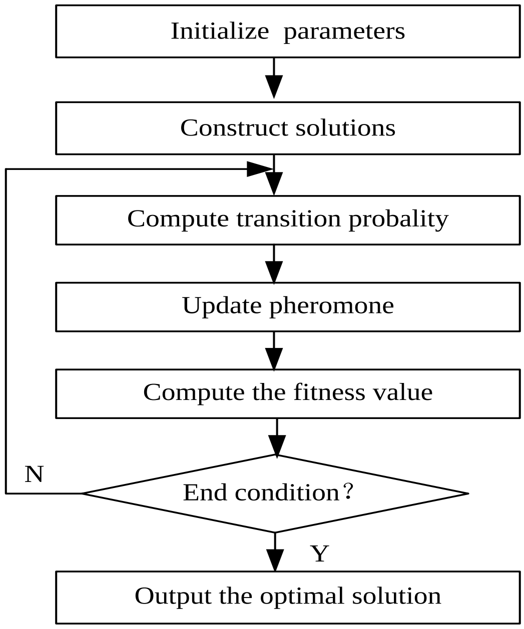


Figure 1. The Flow of the ACO Algorithm

The procedure of pheromone update rule is shown as follows:

(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 expression:

$$s = \arg\max_{u \in J_r^k} [\tau_i(r,u)^{\alpha} \cdot \eta(r,u)^{\beta}] \quad if \; q \le q_0 \, (Exploitation) \tag{1}$$

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

$$p_k(r,s) = \begin{cases} \dfrac{\tau(r,s)^{\alpha} \cdot \eta(r,s)^{\beta}}{\sum_{u \in J_r^k} \tau(r,u)^{\alpha} \cdot \eta(r,u)^{\beta}} & if \;\; s \in J_r^k \\ & if \; q > q_0 \, (Bias \;\; Exploitation) \\ 0 & otherwise \end{cases} \tag{2}$$

In two formula, $p_k(r,s)$ is the transition probability, $\tau(r,u)$ is the intensity of pheromone between city r and city u in the i[th] group, $\eta(r,u)$ is the length of the path from city r to city u, J_r^k is the set of unvisited cities of the k[th] ant in the i[th] group, the parameter α and β are the control parameters, q is a uniform probability [0-1].

(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:

$$\tau(r,u) = (1-\rho)\tau(r,s) + \sum_{k=1}^{m} \Delta\tau_k(r,s) \tag{3}$$

In the formula (3), ρ $(0<\rho<1)$ 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:

$$\Delta\tau_k(r,s) = \begin{cases} \dfrac{Q}{L_k} & (r,s)\in\pi_k \\ 0 & otherwise \end{cases} \tag{4}$$

Where Q is a constant parameter, L_k is the distance of the sequence π_k toured by ant in Δt.

3. Improved ACO (CEULACO) Algorithm

The ACO algorithm is easy to fall into local optimal solution and appear stagnation in the search by analyzing the algorithm. The update of pheromone concentration cannot accurately react path information to mainly cause these problems. So the update and set of pheromone concentration are improved.

3.1. Improve Initial Pheromone Concentration

In the classical ACO algorithm, because the initial pheromone concentration is equal, this will lead to blindly search at the beginning, which generate a large number of irrelevant paths and mislead the update of pheromone concentration. This problem will not only lead to the long time in initial search, but also hinder the search process of the optimal path because of enhancing pheromone concentration on the irrelevant path. The ACO algorithm is introduced into the local optimum. In this paper, the directional guidance is introduced into the ACO algorithm in initializing pheromone concentration according to the path information of undirected graph. So the initial pheromone formula is modified as follows:

$$\tau_{ij}(0) = Q(t)/(d_{Sj} + d_{jE}) \tag{5}$$

Where d_{Sj} and d_{jE} indicate respectively the line distances of the node j to start node S and end node E . $Q(t)$ is real variable function to balance the exploration and exploitation between the random search of ant and the evocation function of information. The real variable function $Q(t)$ is given:

$$Q(t) = \begin{cases} Q_1 & t \le T_1 \\ Q_2 & T_1 < t \le T_2 \\ Q_3 & T_2 < t \le T_3 \end{cases} \tag{6}$$

The shortest distance between start node S and end node E is their connected distance. When the node j is more close to the connected line SE , $(d_{Sj} + d_{jE})$ is smaller, then $\tau_{ij}(0)$ is bigger. The ant is inclined to choose the node as moving direction. The improved strategy will generate a more reasonable direction to search by introducing into different weights in the initial search. This will restrain the search of irrelevant path and speed up finding the global optimal solution.

3.2. Improve Pheromone Evaporation Rate

The pheromone evaporation rate (ρ) is a constant in the classical ACO algorithm. The value of ρ will directly affect the global search ability and convergence speed. If the pheromone concentration is too large, the selection probability of unvisited path will be large, but it will directly affect the global search ability. So it is the key to give the pheromone concentration. In this paper, pheromone dynamic evaporation rate strategy is used to control the pheromone evaporation. At the beginning of ACO algorithm, the larger value of evaporation rate (ρ) is given to enhance the global search ability. With the running of ACO algorithm, the pheromone evaporation rate (ρ) is decaying, it will quickly converge to the optimal solution. So the pheromone dynamic evaporation rate ($\rho(t)$) can not only increase the global search capability, but also accelerate the convergence speed. In here, the curve decay model of pheromone evaporation rate is used to control the pheromone concentration. The curve decay model is described as follow:

$$\rho(t) = \frac{T \times (\tau_{\max} - \tau_{\min}) \times t}{T - 1} + \frac{T \times \tau_{\min} - \tau_{\max}}{T - 1} \tag{7}$$

$$\tau(t) = (1 - \rho(t)) \times \tau(t) + \sum_{k=1}^{m} \Delta \tau_{ij}^{k}(t) \tag{8}$$

Where $\tau_{\min}$ and $\tau_{\max}$ respectively are the lower and upper pheromone. t and T respectively are the current iteration and the maximum iteration.

3.3. Improve Pheromone Updating Rule

In the classical ACO algorithm, the global update rule only strengthens the pheromone concentration on the global optimal path(the shortest path in each iteration). On the one hand, the pheromone concentration of shortest path was excessively strengthened to easily lead to falling into local optimal solution. On the other hand, the optimal solution path of single iteration has a greater relation with global optimal path. The optimal path result in each iteration should make full use in the ACO algorithm. So adaptive dynamic factor $\sigma \in (0,1)$ is introduced into pheromone updating rule to control the updating proportion of pheromone concentration with optimal solution in single iteration. This will greater strengthen the pheromone concentration of better path in order to better use the pheromone concentration to react the path information. The pheromone concentration of the optimal path in this iteration is globally updated by using expression:

$$\tau_{ij}(t+1) = \tau_{ij}(t+1) + \mu\sigma\Delta\tau_{ij} \tag{9}$$

where $\sigma = \frac{1}{\pi}\arctan\left(\frac{\gamma(L_{local\ \min} - L_{\min})}{\overline{L} - L_{\min}}\right) + \frac{1}{2}$

$\overline{L}$ is the average length of the current finding local optimal solution, $L_{\min}$ is the length of current global optimal solution, $L_{local\ \min}$ is the length of the finding optimal path in this iteration, γ is a parameter for controlling the arc tangent function shape. In here, when $L_{local\ \min}$ is bigger and the search path is longer, the adaptive dynamic factor σ is close to 0. And when $L_{local\ \min}$ is smaller and the search path is shorter, the adaptive dynamic factor σ is close to 1.

3.4. Local Search Strategy

2-opt method is often used to solve the TSP problem. This method is used to improve the quality of the solution and select the current optimal path for global updating the pheromone. Because the ACO algorithm completes each iteration, the results of most salesman are not to contribute to the current optimal solution in using process of 2-opt. The 2-opt method is not effective for selecting the current optimal solution on all paths. The results of the current optimal solutions always come from the shorter paths in the current iteration. This will result in consuming a lot of time by using 2-opt method for useless paths in the solving process of ACO algorithm. In the improved ACO algorithm, only part of the better paths use the 2-opt method. This strategy can save some computing time and not reduce the searching efficiency.

4. The Steps of CEULACO Algorithm

According to the improved strategies, the steps of improved ACO(CEULACO) algorithm are described as follows:

Step 1. Read the data of cities, calculate the distance (d_{ij}) of path (i, j) and the adjacency matrix ($allowed_i$).

Step 2. Initialize parameters

The parameters in the CEULACO algorithm are initialized. These parameters include ants(m), the pheromone factor(α), heuristic factor(β),initial pheromone evaporation rate (ρ_0), initial pheromone amount ($Q(0)$),pheromone concentration ($\tau_{ij}(i, j = 1,2,3,\cdots,n)$),the maximum iteration($T_{\max}$), the current iteration($t = 0$),the distance between cities($d_{ij}(i, j = 1,2,3,\cdots,n)$).

Step 3. Initialize position of ant

Initialize Tabu list $t_k(k = 1,2,3,\cdots,m)$, the passed path length $l_k(k = 1,2,3,\cdots,m)$. All the ants randomly choose the initial city, then the selected cities are added into Tabu list t_k, and update l_k.

Step 4. Construct the path

Select the path according to the expression (1). The selected cities are added into Tabu list t_k, and update l_k.

Step 5. After the ants have completed a choice, the path length is calculated. Then the respective Tabu list is modified. The passed path (i, j) is locally updated the pheromone.

Step 6. Execute 2-opt method

When all ants completed the search of all cities, the path construction is finished. The path lengths of all ants are sorted, then the path lengths of the smallest $\frac{m}{2}$ ants are selected to execute 2-opt local search method. The Tabu list t_k and passed path length l_k are updated.

Step 7. Globally update the pheromone

Compare the passed path length l_k with S. If $l_k < S$, the S is replaced by l_k. The pheromone on the current optimal path length is globally updated according to the expression (9) in the improved pheromone updating rule.

Step 8. Iteration control

Set the iterative counter $t = t + 1$. If $t < T_{\max}$, return to Step 3. Otherwise, the proposed CEULACO algorithm is terminated, and the optimal path is output.

5. Traveling Salesmen Problem (TSP)

TSP is one which has commanded much attention of mathematicians and computer scientists specifically, because it is easy to describe and difficult to solve. The TSP can simply be stated as: a search for the shortest closed tour that visits each city once and only once. The distance between the cities is independent of the direction of traversing the arcs, that is, $d_{ij} = d_{ji}$ for every pair of nodes in symmetric TSP.

Define the variables:

$$x_{ij} = \begin{cases} 1 & \text{if } \textit{the arc } (i, j) \textit{ is in the tour} \\ 0 & \textit{otherwise} \end{cases} \tag{10}$$

Objective function:

$$z = \min \sum_{i} \sum_{j} d_{ij} x_{ij} \tag{11}$$

The constraints are written as follows:

$$\sum_{i=1}^{n} x_{ij} = 1, \; j = 1,2,3,\cdots,n \tag{12}$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1,2,3,\cdots,n \tag{13}$$

$$x_{ij} \in \{0,1\}, \; i, j = 1,2,3,\cdots,n \tag{14}$$

$$\sum_{i,j\in S}^{n} x_{ij} \le |S| - 1, 2 \le |S| \le N - 2 \tag{15}$$

6. Experiment Results and Analysis

In order to demonstrate the performance of proposed CEULACO algorithm, 10 datasets of the TSP from TSPLIB standard library (http://www.iwr.uni-heidelberg.de/ groups/comopt/software/TSPLIB95/) with cities scale from 48 to 14051 are selected in this paper. According to TSPLIB, the distance between any two cities is computed by the Euclidian distance and then rounded off after the decimal point. In order to prove the effectiveness of the proposed CEULACO algorithm, the standard ACO algorithm and IMACO algorithm are used to compare the optimized performances. The experiment environments are: Matlab2010b, the Pentium CPU 2.40GHz, 2.0GB RAM with Windows XP. The values of parameters in these algorithms could be a complicated problem itself, the change of parameters could affect the optimum value. So the most reasonable initial values of these parameters are obtained by testing and modifying. The obtained initial values of these parameters are: ants m =30, pheromone factor α =1.0, heuristic factor β =2.0, initial concentration $\tau_{ij}(0)$ =1.5, initial pheromone amount $Q(t)$ =100, maximum iteration times $T_{\max}$ =300.

For each TSP, the standard ACO algorithm, IMACO algorithm and CEULACO algorithm are run independently 30 times, and the best optimal value and average optimal value are found. The results are listed in Table 1.

Table 1. The Results for 10 TSP

No.	Instances	Optimal Value	ACO		IMACO		CEULACO	
			Best	Average	Best	Average	Best	Average
1	att48	33522	34067	343531	33872	34331	**33523**	33734
2	eil51	426	443	462	431	446	**426**	439
3	eil76	538	575	589	552	567	**545**	558
4	rat 99	1211	1264	1295	1226	1253	**1215**	1237
5	lin105	14379	14460	14523	14405	14493	14383	14421
6	ch130	6110	6154	6186	6130	6158	6119	6139
7	kroA200	29368	32142	32217	31034	31317	29526	30853
8	rd400	15281	15441	15516	15369	15487	15358	15441
9	d1655	62128	62618	63014	62553	62705	62495	62647
10	brd14051	469385	477312	479037	476604	476817	476338	476735

As can be seen from Table 1, for the 10 TSP instances, the best value and average value of the proposed CEULACO algorithm are best than the standard ACO algorithm and IMACO algorithm in the experiment. There have 4 TSP instances, which are close to the optimal value. For TSP instances eil51, the proposed CEULACO algorithm can find the best known solution 426. Particularly, for TSP instances att48, eil76 and rat 99, the found best values 33523, 545 and 1215 are approaching to the best known values 33522, 538 and 1211. At the same time, for larger scale instances, the experiment results show that the proposed CEULACO algorithm in the best value is better than the ACO algorithm and IMACO algorithm.

In order to further prove the performance of the proposed CEULACO algorithm, the best route found is shown in Figure 2 and Figure 3.

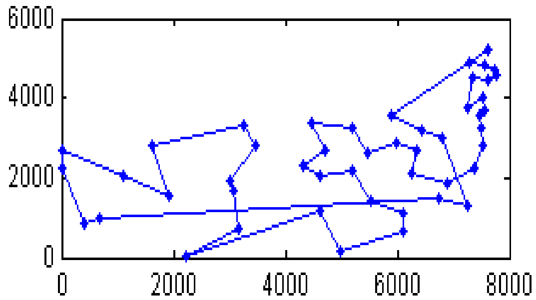


Figure 2. The Best Route Found for att48 (33523)

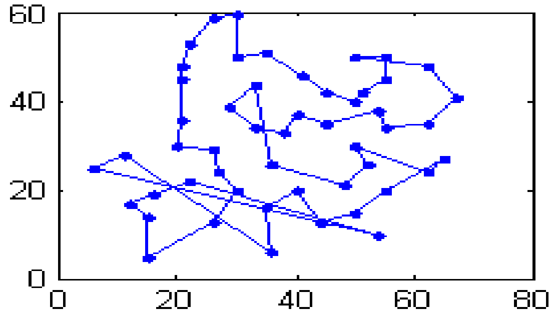


Figure 3. The Best Route Found for eil51 (426)

7. Conclusion

The ACO is a metaheuristic inspired by the behavior of real ants in their search for the shortest path to food. It consists of a number of iterations of solution construction. But it exits a longer computing time in the search process is prone to fall into local optimal solutions. So an improved ACO (CEULACO) algorithm is proposed in this paper. The direction guidance and real variable function, the pheromone dynamic evaporation rate strategy, the local search strategy are used to improve the basic ACO algorithm in order to enhance the global search capability and further improve the performance of the solution. Finally, 10 TSP instances are selected to prove the performance of the CEULACO algorithm. The simulation experiments show that the CEULACO algorithm can effectively improve the accuracy and the quality of solutions, and distinctly speed up the convergence speed and computing time.

Acknowledgment

The program for the initialization, study, training, and simulation of the proposed algorithm in this article was written with the tool-box of MATLAB 2010b produced by the Math-Works.

References

[1] X. S. Yan, “Efficiency analysis of swarm intelligence and randomization techniques”, Journal of Computational and Theoretical Nanoscience, vol. 9, no. 2, **(2012)**, pp. 189-198.

[2] W. Deng, R. Chen, J. Gao, Y. J. Song and J. J. Xu, “A novel parallel hybrid intelligence optimization algorithm for a function approximation problem”, Computers and Mathematics with Applications, vol. 63, no. 1, **(2012)**, pp. 325-336.

[3] S. Z. Zhang, C. K. M. Lee, H. K. Chan, K. L. Choy and Z. Wu, “Swarm intelligence applied in green logistics: A literature review”, Engineering Applications of Artificial Intelligence, vol. 37, **(2015)**, pp. 154-169.

[4] M. Kumar, A. Sharma and R. Kumar, “Optimization of test cases using soft computing techniques: A critical review”, WSEAS Transactions on Information Science and Applications, vol. 8, no. 11, **(2011)**, pp. 440-452.

[5] L. N. de Castro and J. I. Timmis, “Artificial immune systems as a novel soft computing paradigm”, Soft Computing, vol. 7, no. 8, pp. 526-544.

[6] M. Dorigo and M. Gambardellal, “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.

[7] C. W. Zou, J. Q. Xiong and Z. Y. Li, “Improved ant colony algorithm and its application to parameters optimization in storm intensity formula”, Journal of Sichuan University (Engineering Science Edition), vol. 37, no. 5, **(2005)**, pp. 9-13.

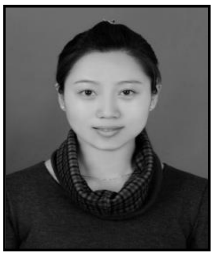
[8] M. R. Jalali, A. Afshar and M. A. Mariño, “Improved ant colony optimization algorithm for reservoir operation”, Scientia Iranica, vol. 13, no. 3, **(2006)**, pp. 295-302.

[9] P. S. Shelokar, P. Siarry, V. K. Jayaraman and B. D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization", Applied Mathematics and Computation, vol. 188, no. 1, **(2007)**, pp. 129-142.
[10] S. C. Gao, W. Wang, H. W. Dai, F. J. Li and Z. Tang, "Improved clonal selection algorithm combined with ant colony optimization", IEICE Transactions on Information and Systems, E91-D, no. 6, **(2008)**, pp. 1813-1823.
[11] W. K. Foong, H. R. Maier and A. R.Simpson, "Power plant maintenance scheduling using ant colony optimization: An improved formulation", Engineering Optimization, vol. 40, no. 4, pp. 309-329.
[12] O. Baskan, S. Haldenbilen, H. Ceylan and H. Ceylan, "A new solution algorithm for improving performance of ant colony optimization", Applied Mathematics and Computation, vol. 211, no. 1, **(2009)**, pp. 75-84.
[13] B. Yu, Z. Z. Yang and B. Z. Yao, "An improved ant colony optimization for vehicle routing problem", European Journal of Operational Research, vol. 196, no. 1, **(2009)**, pp. 171-176.
[14] A. Kaveh and S. Talatahari, "An improved ant colony optimization for constrained engineering design problems", Engineering Computations (Swansea, Wales), vol. 27, no. 1, **(2010)**, pp. 155-182.
[15] L. Santos, J. Coutinho-Rodrigues and J. R. Current, "An improved ant colony optimization based algorithm for the capacitated arc routing problem", Transportation Research Part B: Methodological, vol. 44, no. 2, **(2010)**, pp. 246-266.
[16] J. A. Yang and Y. B. Zhuang, "An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem", Applied Soft Computing Journal, vol. 10, no. 2, **(2010)**, pp. 653-660.
[17] R. W. Gan, Q. S. Guo, H. Y. Chang and Y. Yi, "Improved ant colony optimization algorithm for the traveling salesman problems", Journal of Systems Engineering and Electronics, vol. 21, no. 2, **(2010)**, pp. 329-333.
[18] Q. Duan and T. W. Liao, "Improved ant colony optimization algorithms for determining project critical paths", Automation in Construction, vol. 19, no. 6, **(2010)**, pp. 676-693.
[19] B. Yu, Z. Z. Yang and J. X. Xie, "A parallel improved ant colony optimization for multi-depot vehicle routing problem", Journal of the Operational Research Society, vol. 62, no. 1, **(2011)**, pp. 183-188.
[20] J. Q. Geng, L. P. Weng and S. H. Liu, "An improved ant colony optimization algorithm for nonlinear resource-leveling problems", Computers and Mathematics with Applications, vol. 61, no. 8, **(2011)**, pp. 2300-2305.
[21] R. Jovanovic and M. Tuba, "An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem", Applied Soft Computing Journal, vol. 11, no. 8, **(2011)**, pp. 5360-5366.
[22] Q. L. Ding, X. P. Hu, L. J. Sun and Y. Z. Wang, "An improved ant colony optimization and its application to vehicle routing problem with time windows", Neurocomputing, vol. 98, **(2012)**, pp. 101-107.
[23] W. Deng, R. Chen, B. He, Y. Q. Liu, L. F. Yin and J. H. Guo, "A novel two-stage hybrid swarm intelligence optimization algorithm and application", Soft Computing, vol. 16, no. 10, **(2012)**, pp. 1707-1722.
[24] B. Y. Cheng, Q. Wang, S. L. Yang and X. X. Hu, "An improved ant colony optimization for scheduling identical parallel batching machines with arbitrary job sizes", Applied Soft Computing Journal, vol. 13, no. 2, **(2013)**, pp. 765-772.
[25] X. Q. Wang, Y. Zhao, D. Wang, H. J. Zhu and Q. Zhang, "Improved multi-objective ant colony optimization algorithm and its application in complex reasoning", Chinese Journal of Mechanical Engineering (English Edition), vol. 26, no. 5, **(2013)**, pp. 1031-1040.
[26] Q. X. Zheng, M. Li, Y. X. Li and Q. H. Tang, "An improved ant colony optimization for two-sided assembly line balancing problem", Acta Electronica Sinica, vol. 42, no. 5, **(2014)**, pp. 841-845.
[27] J. E. Guo and W. G. Diao, "An improved ant colony optimization algorithm with crossover operator", Open Mechanical Engineering Journal, vol. 8, no. 1, **(2014)**, pp. 96-100.
[28] S. L. Luo, L. L. Cheng, B. Ren and Q. M. Zhu, "An improved intelligent ant colony algorithm for the reliability optimization problem in cyber-physical systems", Journal of Software, vol. 9, no. 1, **(2014)**, pp. 20-25.
[29] Y. G. Zhang, S. B. Zhang and Q. S. Xue, "Improved ant colony optimization algorithm for solving constraint satisfaction problem", Journal on Communications, vol. 36, no. 5, **(2015)**, pp. 1-6.
[30] Y. L. Tang, Y. G. Cai and Q. J. Yang, "Improved ant colony optimization for multi-depot heterogeneous vehicle routing problem with soft time windows", Journal of Southeast University (English Edition), vol. 31, no. 1, **(2015)**, 94-99.
[31] S. C. Pang, T. M. Ma and T. Liu, "An improved ant colony optimization with optimal search library for solving the traveling salesman problem", Journal of Computational and Theoretical Nanoscience, vol. 12, no. 7, **(2015)**, pp. 1440-1444.
[32] W. Deng, H. M. Zhao, J. J. Liu, X. L. Yan, Y. Y. Li, L. F. Yin and C. H. Ding, "An improved CACO algorithm based on adaptive method and multi-variant strategies", Soft Computing, vol. 19, no. 3, **(2015)**, pp. 701-713.

[33] W. Y. Jiang, Y. Lin, M. Chen and Y. Y. Yu, “A co-evolutionary improved multi-ant colony optimization for ship multiple and branch pipe route design”, Ocean Engineering, vol. 102, **(2015)**, pp. 63-70.

Authors

Yimeng Yue, Lecturer, received the Master degree in computational mathematics from Xi'an University of Architecture And Technology in 2010, Xi'an, China. The main research directions: Algorithm, Mathematical Modeling.



Xin Wang, Lecturer, received the Master degree in human resource management from Xi'an University of Technology in 2011, Xi'an, China. The main research directions: Managerial Mathematics.

