TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.2, February 2014, pp. 984 ~ 990
DOI: http://dx.doi.org/10.11591/telkomnika.v12i2.4211

Optimized Ant Colony Algorithm by Local Pheromone Update

Hui Yu
School of Computer Science and Technology/ HuBei University of Education
Wuhan, China. 027-87943677
e-mail: 29982668@qq.com

Abstract

Ant colony algorithm, a heuristic simulated algorithm, provides better solutions for non-convex, non-linear and discontinuous optimization problems. For ant colony algorithm, it is frequently to be trapped into local optimum, which might lead to stagnation. This article presents the city-select strategy, local pheromone update strategy, optimum solution prediction strategy and local optimization strategy to optimize ant colony algorithm, provides ant colony algorithm based on local pheromone update, also inspects and verifies it by TSP problems. The results of the numeric experiments suggest that on some TSP problems, the optimized ant colony algorithm acquired more satisfied solutions than all that we have already known.

Keywords: *ant colony algorithm, local pheromone update strategy, optimum solution prediction strategy, local optimization strategy*

1. Introduction

Ant Colony Optimization (ACO) is an evolutionary algorithm, which simulates the ant behavior of feeding. It was first proposed by Italian scientist M.Dorlgo, enlightened by biological evolution mechanism in 1990s' [1-2]. In the nature world, the ants search the shortest path between their colony and a source of food by information communication and cooperation among individuals. In the movement of the ants, a substance called "pheromone" is laid down on the trail, of which the intensity can be detected and estimated by every ant, and lead them to move towards the direction of high pheromone intensity. So in the process of ant colony seeking food, the more ants have passed on one path, the higher the probability other ants would choose this path, the more pheromone would be kept consequently. Time after time, this path would be selected by all the ants as the shortest. The Ant Colony Algorithm simulates the ants foraging mechanism, constructs a number of artificial ants, searching the path according to the pheromone intensity on it. When all ants finish one searching, update the intensity of the pheromone. Iterated continuously, most ants will follow the same path(the optimal path) to end the searching.

The research suggests that the ant colony optimization has many advantages such as distributed computing, and it can be easily composited with other methods, and has good robustness [2]. It achieved great successes in solving Combinatorial Optimization Problems like TSP [3] (Traveling Salesman Problem), QAP (Quadratic Assignment Problem), and JSP(Jobshop Scheduling Problem). Meanwhile, it is defective. The biggest problem of it is that when solving problems it easily gets involved in stagnation and might be trapped in local optima. For the last twenty years, scientists have been studying the ant colony optimization and came up with plenty of algorithms to improve the standard ant colony algorithm such as MAX-MIN Ant System [4], the composite of Immunity Algorithm and Ant Algorithm[5-6], the hybrid of Genetic Algorithm and Ant Colony Algorithm [7-10]. This article briefly presents the Ant Colony Algorithm and TSP problems then particularly discusses the introduction of city-select strategy, local pheromone update strategy, optimum solution prediction strategy and local optimization strategy and tests some TSP problems with experiment. The first part of this article presents the principle of the Ant Colony Algorithm; the second part is about city-select strategy, in which the Prior Probability has been brought; the third part is about local pheromone update strategy, in which the update of local pheromone by Nearest Neighbor Algorithm has been introduced; the fourth

Received July 25, 2013; Revised September 3, 2013; Accepted September 28, 2013

part is about optimum solution prediction strategy, in which how to predict the optimum solution and the incentives and disincentives have been introduced; the fifth part talks about the local optimization strategy, using the Simulated Annealing Algorithm to optimize the optimal solution and finally the sixth part presents the results and analysis of our experiments which is also the conclusion of this paper.

2. TSP Problem by Ant Colony Algorithm

2.1. Mathmatic Model

We use n cities (1,2,…,i,…j,…, n is the city number) TSP problems to explain the basic ant colony algorithm. The so-called n city TSP problem is to find the shortest path when ants visiting n cities without repetition. First we introduce conventional symbols in TSP problem:

m is the quantity of ant in the colony, D_{ij} is the distance between city i and city j, τ_{ij} is the amount of residual pheromone on side (i, j), η_{ij} is the trigger factor on side (i, j), $\eta_{ij} = 1 / D_{ij}$, $Tabu_k$ is the tabu list of the ant k (recording the path that the ant k has visited), $\Delta\tau_{ij}^{k}$ is the amount of pheromone laid down by ant k on the side (i, j), L_k is the distance of ant k passing around cities.

Initially, the amount of pheromone is same on each path: $\tau_{ij} = C$ (C is a non-zero constant), ant k ($k = 1,2,\cdots,m$) is put randomly in one city, and it selects next city according to the amount of pheromone on each path, the probability P_{ij}^{k} from city i to city j, at time t, can be calculated as below:

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{s\in allowed_k}\tau_{is}^{\alpha}(t)\eta_{is}^{\beta}}, & j \in allowed_k \\ 0 & other \end{cases} \quad (1)$$

In the formula: $allowed_k(1,2,\cdots,n)$ are the cities allowed to select, α is the importance of the pheromone , β is the importance of the trigger factor.

To avoid ant visiting one city repeatedly, add the visited city into tabu list $Tabu_{ij}^{k} = to\_visit$.

With the time goes by, the pheromone is declining. After n moments, the ant finishes one circulation, adjust the pheromone on each path by formulas as follows:

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \Delta\tau_{ij} \quad (2)$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \quad (3)$$

In the formula: $\Delta\tau_{ij}$ is the total residual amount of pheromone on path ij in this circulation, and can be calculated by Ant-cycle System [5]:

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{ant k passes through path ij in the circulation} \\ 0 & Others \end{cases} \quad (4)$$

In the formula: Q is a constant, indicates the intensity of pheromone, which can affect the convergence rate of the algorithm to some extent, L_k represents the total Length of the path ant k has visited in the circulation.

After all ants have finished one circulation, one iteration process is completed. Record the optimal path. The final optimal path can be searched by repeating the above processes until the pre-concerted iteration.

2.2. The Defect of Ant Colony Algorithm

The standard ant colony algorithm uses the global update pheromone method, which has taken into account the searching result of each ant, which indicates the feedback of pheromone, however, global pheromone update partly results in the postpone of pheromone update. So in that case, the ant colony can not be guided to the optimal solution. In the process of searching, the ant colony always tends towards the path of the highest pheromone intensity. As the existence of positive feedback, the currently best path will accumulate much more pheromone than any other paths after a period of time, even if there is a better solution, it can not be found by the ant colony as the pheromone intensity on it is far less than the currently best solution. The ant colony is trapped in local optimum, the better solution can not be found [12-13].

3. Improved ant Colony Algorithm

3.1. City Select Strategy

The city select strategy which is introduced here is defined as below, in the process of ant visiting cities, set a prior probability q_0 , and a random number q is generated, when the ant selects next city; compare q with q_0 , if it is bigger than q_0 ,select next city node by probability P_{ij}^k according to formula(5), otherwise by probability P_{ij}^k according to formula (7).

$$P_{ij}^k = MAX\{P_{ij1}^k\} \tag{5}$$

$$P_{ij1}^k = \begin{cases} \tau_{ij}^{\alpha}\eta_{ij}^{\beta} & j \in allowed_k \\ 0 & i = j \end{cases} \tag{6}$$

$$P_{ij}^k = MAX\{P_{ij2}^k\} \tag{7}$$

$$P_{ij2}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum \tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}}, & j \in allowed_k \\ 0 & i = j \end{cases} \tag{8}$$

In the formula: τ_{ij} indicates the pheromone from city i to city j , α is the importance of pheromone, η_{ij} is the trigger information from city i to city j , β is the importance of the trigger information, $allowed_k(1,2,\cdots,n)$ are cities that can be selected by ant k in next step.

3.2. Local Pheromone Update

The global update pheromone strategy used in standard ant colony algorithm has delay effect, so local pheromone update strategy is introduced to guide the movement of ants without delay. When ant selects next city according to city-select strategy, update the pheromone matrix according to formula(9). The local pheromone update strategy formula (9) is as below:

$$\tau_{ij} = (1-\delta)\tau_{ij} + \frac{\delta}{N \cdot Lnn} \tag{9}$$

In the formula, $\delta(0 < \delta < 1)$ is the evaporation rate of local pheromone, N is the number of city nodes, Lnn is calculated by Nearest Neighbor Algorithm. Start from one random city, according to Greedy Algorithm, a city with shortest distance is selected as next city node, the total distance after visiting all the city nodes is defined as Lnn .

3.3. Optimal Solution Prediction Strategy

A good solution can be searched instantly by running the Ant Colony Algorithm, but as it's easily trapped into local optimum, the solution might be taken for optimum and preserved for continuous iterations. With the update to pheromone matrix by global pheromone strategy, the amount of pheromone on this path is far more than that on other paths. The superior solution can't be searched even if it exists due to the less amount of pheromone on the path. Accordingly, this article proposes a method to predict the current optimal solution found by ant colony. When the current better solution is preserved for certain iterations, better solutions can be found by rewarding the pheromone on current better path to accelerate convergence of ant colony; when the current better solution is preserved over certain iterations, it will be predicted to have already been trapped into local optimum. Pheromone on the path of current solution will be penalized to avoid its accumulation so as to prevent local optimum.

The process of optimal solution prediction strategy is as follows: after visiting all city nodes the ant colony will find a currently better solution and count the preserved iterations. When the preserved iterations number of the better solution found by ant colony is smaller than $0.05nMax$ of the total number of iterations, update the information on the optimal path according to formula (10) to accelerate convergence; when the preserved iteration number of the better solution is bigger or equal to $0.05nMax$, and smaller than $0.10nMax$, penalize the current optimal path to avoid the local optimum, penalty coefficient is 0.05, and update the pheromone according to formula (11). The rest may be deduced by analogy: when the preserved iteration number of the better solution is bigger than or equal to $0.10nMax$,and smaller than, penalty coefficient is 0.10; when the preserved $0.20nMax$ iteration number of the better solution is bigger than or equal to $0.20nMax$, and smaller than $0.30nMax$, penalty coefficient is 0.20; when the preserved iteration number of the better solution is bigger than or equal to $0.30nMax$, and smaller than $0.40nMax$, penalty coefficient is 0.30; when the preserved iteration number of the better solution is bigger than or equal to $0.40nMax$, the current optimal path is predicted to have already been trapped into local optimum, set the pheromone on current optimal path as initial value according to formula (12).

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \frac{\rho}{N\cdot L_{best}} \tag{10}$$

In the formula: $\rho(0 < \rho < 1)$ is the evaporation rate of global pheromone, N is the number of city nodes, L_{best} is the currently shortest path.

$$\tau_{ij} = \tau_{ij} + \frac{\gamma}{N\cdot L_{best}} \tag{11}$$

In the formula: $\gamma(0 < \gamma < 1)$ is the pheromone penalty factor.

$$\tau_{ij} = \frac{1}{N\cdot Lnn} \tag{12}$$

In the formula: Lnn is calculated from Nearest-Neighborhood distance.

3.4. Local Optimization Strategy

Local optimization strategy optimizes the current better solution by Simulated Annealing Algorithm. A better solution and corresponding path will be found after ant colony visiting all city nodes. Take the path as initial solution of simulated annealing algorithm, and search by simulated annealing algorithm. Optimize the better solutions of each iteration by simulated annealing algorithm, and it helps Ant Colony Algorithm jump out the local optimal solution trap, and get superior solution in solution domain.

3.5. Algorithm Flow

The general Ant Colony Algorism flow based on local pheromone update is given as below:

Step 1: initialize the quantity of ant M, the maximum iteration number $nMax$, the importance of pheromone α, the importance of heuristic information β, the evaporation rate of local pheromone δ, the evaporation rate of global pheromone ρ, pheromone matrix τ_{ij}, heuristic information matrix η_{ij}, pheromone penalty factor γ, iteration number $nc = 1$.

Step 2: put M ants in N cities randomly, and store these cities in the ant's tabu list.

Step 3: if the iteration number nc is smaller than the maximum iteration number $nMax$, continue the next step, otherwise jump to Step 11.

Step 4: ant k chooses next city j according to the city-select strategy, meanwhile put city j in $Tabu$ list.

Step 5: update the pheromone of τ_{ij} according to formula (9).

Step 6: once M ants have visited N cities, a better solution $LocalbestValue$ comes out.

Step 7: compare the better solution $LocalbestValue$ with the global optimal solution $GlobalbestValue$, if it is smaller, then $GlobalbestValue = LocalbestValue$, counting variable $num = 1$; otherwise counting variable $num = num + 1$.

Step 8: the pheromone τ_{ij} is processed in the optimal solution prediction strategy, according to the counting variable num.

Step 9: optimize the optimal solution by Local Optimization Strategy.

Step 10: $nc = nc + 1$, jump to Step3.

Step 11: output the optimal solution (the shortest path and path length).

4. Experiment Results and Analysis

Pick some TSP problems from TSPLIB. (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/), and run them several times by C++ language. The results are listed as below in Table 1.

Table 1. The Comparison of Results of some TSP Problems

problem	TSPLIB Optimal solution	Refs. [14~17]	Optimal solution	The worst solution	deviation
Oliver30	423.741	423.74	423.741	424.692	0.00
C-TSP	15404	15402	15402	15471	0.0000
Elion51	426	426	426	428.468	0.0000
Elion76	542.31	538	544.882	549.492	0.0047
KroA100	21294	21292	20583	21644	-0.0345
Ch130	6110	6110	6168.66	6180.54	0.0094
Ch150	6532	6528	6402	6574	-0.02
Gr202	549.9981	-	496.3180	499.9715	-0.1082
Tsp225	3919	3919	3859	3982	-0.0155
Pcb442	50784	50778	50783	51601	0.0000
Pa561	19331	-	15776	15905	-0.2253
Pa561	19331	-	15776	15905	-0.2253

The deviation in Table 1 presents the discrepancy percentage compared with the optimal solutions we have known. It shows in Table 1 that the optimal solutions are very close, comparing the optimized Ant Colony Algorithm with the best experiment results we have known on Oliver30、Elion51、Pcb42 and Travelling salesman problems in 31 cities of China. On Elin75 and Ch130 problems, the deviations of the optimal solutions found by optimized Ant Colony Algorithm and the currently-known optimal solutions are 0.0047 and 0.0094, which are less than the acceptable probability range of 0.1. On Ch150, Gr202, Tsp225, and Pa561 problems, the optimized Ant Colony Algorithm can find better solutions than optimal experiment results we have known.

Table 2 is the parameter configuration acquired from Table 1 on solving different TSP problems. Table 1 shows, ant number of the colony is identical to the number of city node $M = N$,parameter α and β accrue with the growth of the city size, and inspected by experiment, we can get better solutions when α is smaller than β , set both parameter δ and ρ to 0.1, identical to Reference.[18], set parameter q_0 to 0.01 when the number of city node is less than 50, otherwise set it to 0.1, tested by experiment.

Table 2. Parameter Settings of the TSP Problems

problem	M	$nMax$	α	β	δ	ρ	q_0
Oliver30	30	500	2	3	0.1	0.1	0.01
31cities of China	31	500	2	3	0.1	0.1	0.01
Elion51	51	5000	2	4	0.1	0.1	0.01
Elion76	76	1000	2	5	0.1	0.1	0.1
KroA100	100	1000	3	5	0.1	0.1	0.1
Ch130	130	1000	3	7	0.1	0.1	0.1
Ch150	150	1000	3	8	0.1	0.1	0.1
Gr202	202	1000	4	6	0.1	0.1	0.1
Tsp225	225	1000	4	7	0.1	0.1	0.1
Pcb442	442	2000	5	9	0.1	0.1	0.1
Pa561	561	2000	5	10	0.1	0.1	0.1

5. Conclusion

This article introduced city-select strategy, local pheromone update strategy, optimum solution prediction strategy and local optimization strategy to optimize ant colony algorithm. Compared with some optimal solutions on TSP problems that we have known, the optimized algorithm acquired satisfied solutions. Because of the introduction of the optimal prediction strategy, the pheromone accumulation of the optimal solution was blown up, more iterations were needed when searching the optimal solution in the solution domain, and the convergence rate was slowed down. These problems need to be improved and further research is needed in next phase of our work.

Acknowledgement

This work is supported by the Research Project of Hubei Provincial Department of Education in 2013, No. B2013018 and Hubei Natural Science Foundation in 2012, No. 2012EFB00402.

References

[1] Marco Dorigoa, Christian Blum.Ant colony optimization theory: A survey. *Theoretical Computer Science*. 2005; 344: 243-278.

[2] Marco Dorigo, Vittorio Maniezzo, Alberto Colorni. The ant system: optimization by a colony of cooperation agents.1996; 26(1): 29-41.

[3] Marco Dorigo, Luca Maria Gambardella. Ant colonies for the traveling salesman problem. *BioSystems*. 1997; 43(2): 73-81.

[4] Thomas Stützle, Holger H Hoosb. MAX-MIN Ant System. *Future Generation Computer Systems*. 2000; 16(8): 889-914.

[5] Wei Gao. *Study on immunized ant colony optimization.* Proceedings Third International Conference on Natural Computation, ICNC. 2007; 4: 792-796.

[6] Licheng Jiao, Lei Wang. A Novel Genetic Algorithm Based on Immunity. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans.* 2000; 30(5): 552-561.

[7] Daoxiong Gong, Xiaogang Ruan. *A hybrid approach of GA and ACO for TSP*. Proceedings of the World Congress on Intelligent Control and Automation (WCICA). 2004; 3: 2068-2072.

[8] Zne-Jung Lee, Shun-Feng Su, Chen-Chia Chuang, Kuan-Hung Liu. Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. Applied Soft Computing. 2008; 8: 55–78.

[9] Sarayut Nonsiri, Siriporn Supratid. *Modifying Ant Colony Optimization.* IEEE Conference on Soft Computing in Industrial Applications (SMCia/08), Muroran, JAPAN. 2008.

[10] Hui Wang. Comparison of several intelligent algorithms for solving TSP problem in industrial engineering. Systems Engineering Procedia. 2012; 4: 226-235.

[11] Chengming Qi. Vehicle Routing Optimization in Logistics Distribution Using Hybrid Ant Colony Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(9): 5308-5315.

[12] Guoqiang Zheng, Zhao Liang, Jishun Li. Optimization of an Intelligent Controller for Parallel Autonomous Parking. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(2): 1069-1075.

[13] Wang Xuan, Li Yuanxiang. An Evolutionary Algorithm Combined with Local Searching Strategy for Solving TSP. *Computer Engineering*. 2006: 1000-3428

[14] Ji Junzhong, Huang Zhen, Liu Chunnian. A Fast Ant Colony Optimization Algorithm for Traveling Salesman Problems. *Journal of Computer Research and Development.* 2009; 1000-1239.

[15] CHEN Xing-yu, XIAO WeiQUAN, Hui-yun. A Hybrid Ant Colony Algorithm Using LK Algorithm for Traveling Salesman Problem. *Computer Engineering*. 2008:1000-3428.

[16] Cao Langcai, Luo Jian, Li Tiancheng. Intelligent Ant Algorithm: The Improvement of Ant Colony Algorithm. *Application Research of Computers.* 2003; 1001-3695.

[17] Lingling Wang, Qingbao Zhu. *An Efficient Approach for Solving TSP: the Rapidly Convergent Ant Colony Algorithm.* Proceedings International Conference on Natural Computation, ICNC. 2008; 4: 448-452.