Improved Ant Colony Algorithm for Traveling Salesman Problems

Pei-dong Wang，Gong-You Tang，Yang Li，Xi-Xin Yang

College of Information Science and Engineering, Ocean University of China, Qingdao, 266100

E-mail:{qdrnocking; gtang; mtlyab; yangxixin}@ouc.edu.cn

Abstract: An improved ant colony algorithm is proposed in this paper for Traveling Salesman Problems **(**TSPs). In the process of searching, the ants are more sensitive to the optimal path because the inverse of distance among cities is chosen as the heuristic information, while a candidate list is used to limit the number of candidate city. The method of local and global dynamic phenomenon update is used in order to adjust the distribution of phenomenon according to the routes. The method of 2-opt is only used for the current optimal tour, enhancing the convergence speed. The simulation results demonstrate the proposed algorithm works well and efficient.

Key Words: Ant colony algorithm, Path planning, Dynamic pheromone updating, TSPs

1 INTRODUCTION

The Traveling Salesman Problems (TSPs) is perhaps the most popularly studied discrete optimization problem. Its popularity is due to the facts that TSP is easy to formulate, difficult to solve, and has a large number of applications. TSPs become a benchmark for many new approaches in combinatorial optimization. A large number of publications have been dedicated to study the TSPs, together with some of its variations.

At present there are several methods to solve TSPs, such as Branch-and-cut Algorithm [1], Genetic Algorithm [2, 3], Ant Colony Algorithm [4-12] and so on. Gambardella [5] introduced Ant Colony System (ACS) with local pheromone updating and global pheromone updating rules, improving the global convergence capacity. Stutzle [6] introduced Max-Min Ant System (MMAS) imposing restrictions on the value of pheromone, avoiding the algorithm at the stagnation. Huang [7] introduced Pheromone Diffusion Ant colony Optimization (PDACO) using pheromone diffusion model to modify original pheromone updating rule. Ding [8] introduced an algorithm (GAAA) with combination of Genetic algorithm and Ant Algorithm, enhancing the convergence speed. Zhu [9] introduced an algorithm (NDMACO) with mutation scheme and dynamic pheromone updating rule. Yang [10] introduced a method with a mutation process and a new local searching technique. Ji [11] introduced an algorithm (PIPDMACO) with a pheromone increment model, new pheromone diffusion model and mutation strategy with lower computational complexity.

In this paper an improved ant colony algorithm is proposed for TSPs. Firstly, in the process of searching ants are more sensitive to the optimal path because the inverse of distance among cities is chosen as the heuristic information while a candidate list is used to limit the number of candidate city. Meanwhile, the method of local and global dynamic phenomenon update is used so as to adjust the distribution of phenomenon according to vehicle routes. Finally, the method of 2-opt is also used to expand the scope of search for current shortest path, enhancing the convergence speed of optimal solution. The simulation results demonstrate this proposed algorithm has high efficiency..

This work is supported in part by the National Nature Science Foundation of China under Grant 61074092, the Natural Science Foundation of Shandong Province under Grant ZR2010FM019, and the Scientific and Technological Project of Shandong Province under Grant 2008GGB01192.

2 PROBLEM DESCRIPTION

TSPs can be stated simply that a salesman spends his time visiting n cities (or nodes) cyclically. In one tour he visits each city just once, and finishes up where he started. The question is that in what order he should visit the cities to minimize the distance traveled.

Define an undirected graph which is used to model TSPs. The undirected graph $G(V,A)$ consists of a vertex set

$$V(G)=\{v_k : k\in l_n\} \quad l_n=\{1,2,...,n\}$$

and an arc set

$$A(G)\subset\left\{\left\langle v_p, v_q\right\rangle : v_p, v_q\in V(G)\right\}$$

where n is the number of cities. Each arc is assigned a value d_{ij}, which is the length of $arc(i,j)\in A$, that is, the distance between city i and j.

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

The goal of the TSPs is to find an optimal length Hamiltonian circuit of graph $G(V,A)$ to minimize transportation cost J

$$J=\min\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}x_{ij} \tag{2}$$

where

$$\sum_{\substack{i=1\\i\neq j}}^{n} x_{ij}=1,\quad j\in(1,2,...,n) \tag{3}$$

$$\sum_{\substack{j=1\\j\neq i}}^{n} x_{ij}=1,\quad i\in(1,2,...,n) \tag{4}$$

$$\sum_{\substack{i\in S\\i\neq j}}\sum_{\substack{j\in S\\i\neq j}} x_{ij}\le |S|-1,\quad |S|<n 且 S\subset V(G) \tag{5}$$

Eq. (3) ensures that a tour must come into vertex j exactly once and Eq. (4) indicates that a tour must leave every vertex i exactly once. These two sets of constraints ensure that there are two arcs adjacent to each vertex, one in and one out. Eq. (5), sub-tour elimination constraints, requires that there're no other circuits in graph. S is a subset of vertices that has $|S|$ arcs totally.

3 ALGORITHM DESIGN

3.1 Applying Ant Colony System to the TSPs

The Improved ACS (IACS) algorithm is based on the ACS algorithm that was proposed by Dorigo and Gambardella [5]. In ACS, ants are initially positioned on cities chosen randomly. Each ant builds a tour by repeatedly applying a probabilistic nearest neighbour heuristic. While choosing a city, an ant modifies the pheromone level on the visited edges by applying a local updating rule. When all ants have completed their tours, the pheromone level on each edge is modified again by applying the global updating rule which favors the edges associated with the best tour found from the start iteration. Compared with other intelligent algorithms, ACS is more efficient. However, some limitations still exist in the course of solving TSPs as follows:

(i) Limitation of constructing route： in ACS, while a city is chosen by an ant, transition probability of cities not visited is needed to be calculated according to the ant's tabu table. Usually a large amount of time is consumed.

(ii) Limitation of pheromone update： in the iterative process of algorithm, while the optimal path isn't obtained, pheromone level on the current shortest path is reinforced increasingly under the effect of pheromone updating rule. It can cause two negative results: Firstly, the algorithm can be at the stagnation because of pheromone level on the current shortest path reinforced excessively. Secondly, while the optimal path is just obtained, pheromone level on the optimal path can far lower than on the original shortest path. Sometimes the case still cannot be changed after several iterations. In summary, ACS's pheromone update rules make the alterations of the current shortest path not rapidly presenting pheromone level on the path, therefore the algorithm search efficiency is reduced.

(iii) Limitation of local search：2-opt is a method often used to solve TSPs. The 2-opt process can be shown by Fig.1. [6, 7, 8, 10] apply it to the tour each ant constructs in every iteration. However, if the number of city and ant is large, a mount of time will be consumed.

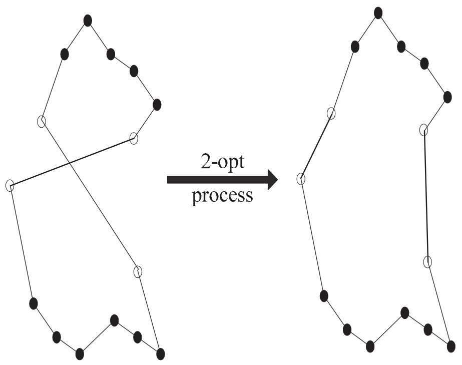


Fig 1. Schematic diagram of the 2-opt process

3.2 Algorithm improvement

In this part three strategies are proposed to improve original ACS to solve TSPs on three limitations.

3.2.1 Route Construction

In an optimal tour for TSPs of a complicated map where there're several cities, the city with which city i is connected cannot be the one of those cities that are far away form city i . According to this reason, those cities, with which city i are connected and remain to be visited by ant k positioned on city i , are sorted ascending in distance between city i and them. $n/5$ of those cities are selected as candidates. This method can improve search efficiency well.

In the route construction, an ant k moves from the city i to the next city j according to the state transition rule given by Eq. (6).

$$\begin{cases} j=\arg\max\limits_{y\in allow(i)}\{\tau_{iy}(t)\eta_{iy}^{\beta}\}, & q\le q_0 \\ p_{ij}{}^{k}=\dfrac{\tau_{ij}(t)\eta_{ij}^{\beta}}{\sum\limits_{y\in allow(i)}\tau_{iy}(t)\eta_{iy}^{\beta}}, & \text{else} \end{cases} \tag{6}$$

where $allow(i)$ is the set of candidate cities that meet the conditions, $\tau_{ij}(t)$ is the pheromone level on edge (i,j) at the time of t, η_{ij} is the heuristic information, which is the inverse of the length of edge (i,j) . β is the parameter that determines the relative influence of heuristic information $(\beta>0)$. Moreover, q is a random number uniformly distributed in $[0,1]$, and q_0 is a pre-defined parameter in $[0,1]$. This state transition rule favors transitions toward cities connected by short edges and with a large amount of pheromone. If $q\le q_0$ the best one next city j , according to Eq. (6) (exploitation), is chosen. Otherwise, a city is chosen according to Eq. (6) (exploration). This selection process continues until all cities are visited and the tour is complete.

3.2.2 Phenomenon Update

Phenomenon local and globe dynamic updating rules are proposed in this section.

(a). Local update

In the process of route construction, the local updating rule in Eq. (7) is applied to change pheromone level of the edge after an ant passes one edge (i, j).

$$\tau_{ij}(t+1) = (1-\varepsilon)\tau_{ij}(t) + \varepsilon\tau_0 \tag{7}$$

Where ε is a user-defined parameter which is called local evaporation coefficient, and τ_0 is the initial pheromone level of all edges. The effect of pheromone local update is to cut down the pheromone level of the edge that ants pass to reduce probability of other ants select the edge and increase opportunities of ants exploring edges not visited.

(b). Globe dynamic update

After every iteration, the globe dynamic updating rule in Eq. (8) is applied to change pheromone level of the current shortest path.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij} \tag{8}$$

and
$$\Delta\tau_{ij} = \frac{L_1 - L_g}{L_g} \tag{9}$$

where ρ is a user-defined parameter called globe evaporation coefficient, $\Delta\tau_{ij}$ is the increment of pheromone update, L_1 is the length of the iteration-best path, L_g is the length of the current shortest path. The effect of pheromone globe dynamic update is to continue to explore the current shortest path, transmit the positive feedback of pheromone level on the current shortest path to next iteration.

Eq. (8) and Eq. (9) can adjust the distribution of phenomenon according to the current shortest path. In the process of iteration, after a current shorter path is obtained, the difference between L_1 and L_g can be very large. It illustrates that only few ants obtain the current shortest path. At this time, Eq. (8) can reinforce the phenomenon level of the current shortest path to attract more ants gathering on the path. With the incensement of iteration, the phenomenon level of the current shortest path continues to get be reinforced and the difference between L_1 and L_g can be gradually smaller. At this monent, $\Delta\tau_{ij}$ can also be very small until 0. When $\Delta\tau_{ij}$ becomes zero, the pheromone on the current shortest only evaporates. The method of local and global dynamic phenomenon update can adjust the distribution of phenomenon according to route construction, and make the alterations of the current shortest path rapidly present pheromone level on the path and improve algorithm search efficiency.

3.2.3 Local Search

In the original ACS, after the ants have constructed their solutions and before the pheromone is globe updated, each ant's solution is improved by applying a local search. However, local search is a time-consuming procedure in ACS. In order to save the computation time, we will only apply local search to the current shortest path. The idea is that better solution may have better chance to find a local optimum via local search.

3.3 Overall Procedure

According to the description of above improvement strategies, the procedures of our IACS are described as follows:

Step1. Set parameters. Set iteration counter NC=0, the length of the optimal path $shortest\_len$, the largest iteration number NC_CONST, the distance between cities $length[i][j](i,j=0,1,...,n)$, the table of the current shortest path $tabu\_\min$, the initial value of phenomenon τ_0 and the phenomenon level of the path $\tau_{ij}(i, j = 1,2,...,n)$.

Step2. Initialize algorithm. m Ants are randomly positioned on the n cities. Initialize every ant's tabu table $ant[k].tabu(k=1,2,...,m)$, the total length of every ant passing the path $ant[k].length(k=1,2,...,m)$.

Step3. Construct Routes. Each ant constructs its route according to Eq. (6), adding the selected city to $ant[k].tabu$ and updating the value of $ant[k].length$.

Step4. Phenomenon local Update. When an ant passes one edge (i, j), the local updating rule in Eq. (7) is applied to change pheromone level of the edge.

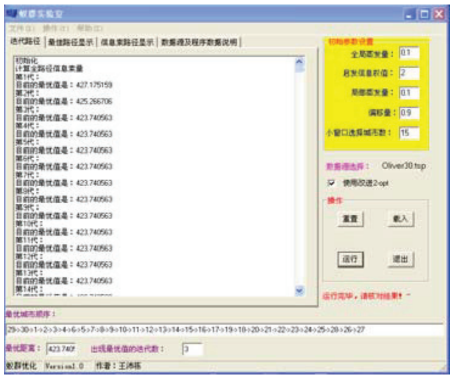
Step5. 2-opt local search. When all ants finish their route constructions, local search 2-opt is applied to the current shortest path in the iteration. Update the value of $ant[k].length$ and $ant[k].tabu$. Otherwise go to **Step2.**

Step6. Phenomenon globe dynamic Update. Compare $ant[k].length$ with $shortest\_len$. The globe dynamic updating rule in Eq. (8) and (9) is applied to change pheromone level on the shortest path.

Step7. Cyclic iteration. If $NC \le NC\_CONST$, return **Step2** and start a new iteration, otherwise the algorithm terminates, output $shortest\_len$ and $tabu\_\min$.

4 SIMULATIONS EXPERIMENTS

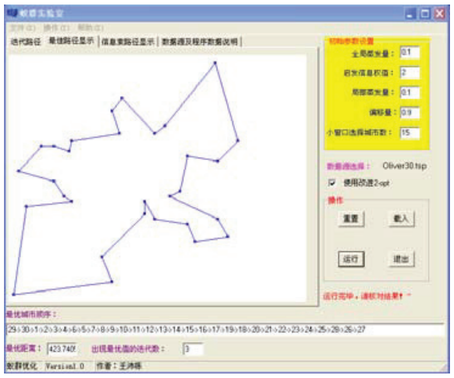
In this section we compare performance of the proposed algorithm with ACS and other algorithms for some TSPs instances. All the TSPs instances we use are taken form the TSPLIB Benchmark library [13] which contains a large collection of instances. Our algorithm is coded in Borland C++ Builder 6.0 and executed on a PC equipped with 512 MB of RAM and an Intel Celeron 1.7GHz. Parameters setting: ρ=0.1~0.2, ε = 0.1 ~ 0.4, β=4~5, q_0=0.7~0.95.



```
初始化
计算全路径信息素量
第1代：
目前的最优值是：427.175159
第2代：
目前的最优值是：425.266706
第3代：
目前的最优值是：423.740563
第4代：
目前的最优值是：423.740563
第5代：
目前的最优值是：423.740563
第6代：
目前的最优值是：423.740563
第7代：
目前的最优值是：423.740563
第8代：
目前的最优值是：423.740563
第9代：
目前的最优值是：423.740563
第10代：
目前的最优值是：423.740563
第11代：
目前的最优值是：423.740563
第12代：
目前的最优值是：423.740563
第13代：
目前的最优值是：423.740563
第14代：
```

初始参数设置
全局挥发量：0.1
启发信息权值：2
局部挥发量：0.1
偏移量：0.9
小窗口选择城市数：15
数据源选择：Oliver30.tsp
☑ 使用改进2-opt
运行完毕，请核对结果！
最优城市顺序：29->30->1->2->3->4->6->5->7->8->9->10->11->12->13->14->15->16->17->19->18->20->21->22->23->24->25->28->26->27
最优距离：423.740 出现最优值的迭代数：3

(a) The result for Oliver30



初始参数设置
全局挥发量：0.1
启发信息权值：2
局部挥发量：0.1
偏移量：0.9
小窗口选择城市数：15
数据源选择：Oliver30.tsp
☑ 使用改进2-opt
运行完毕，请核对结果！
最优城市顺序：29->30->1->2->3->4->6->5->7->8->9->10->11->12->13->14->15->16->17->19->18->20->21->22->23->24->25->28->26->27
最优距离：423.740 出现最优值的迭代数：3

(b) The graph of Oliver30

Fig 2. Simulation result of Oliver30

Simulation result for instance Oliver30 is shown in fig. 2. Fig. 2(a) indicates the length of the current shortest path every an iteration. Fig. 2 (b) indicates the graph of the optimal path.

Table1 Best results for our algorithm on various instances

Instances of TSPs	BK	Best Result	Num of Cycles
Ulysses16	**74**	74	9
Ylysses22	**75.31**	75.31	2
Oliver30	**423.74**	423.74	3
Dantzig42	**699**	699	7
Att48	**33523.7**	33523.7	26
Eil51	**426**	426	12
Berlin52	**7542**	7542	10
St70	**675**	675	21
KroA100	**21285.4**	21285.4	72
Pr107	**44303**	44303	53

The best results for our algorithm on various instances I n 10 runs are listed in Table 1. BK indicates the known optimal solution value of each instance and Num of Cycles indicates the iteration number of the algorithm when Best Result is obtained. Each Best Result of instance is optimal in Table 1. Table 2 shows comparison of performances among the algorithms, including ACS [5], ACS+2-opt [11], PDACO [7], PIPDMACO [11] and our algorithm (IACS). PIPDMACO and our algorithm can obtain the optimal solutions of all instances and have better performances than other three algorithms. Num of Cycles of only 2 instances (Eil51 and Pr107) with PIPDMACO is lower than our algorithm. The main reason is using a new route search method and phenomenon updating rules in our algorithm, enhancing the convergence speed. Simulation experiments results on the different benchmark data sets show that the proposed algorithm can not only get better optimal solution but also enhance the convergence speed of the optimal solution.   .

Table2 Comparison among the algorithm performances

Instance of TSPs		Oliver30	Dantzig42	Eil51	Berlin52	St70	Pr107
BK		**423.74**	**699**	**426**	**7542**	**675**	**44303**
ACS	Best Result	423.74	707	441	7598	690	44401
	Num of Cycles	830	1300	2100	2500	3900	4700
ACS+2-opt	Best Result	423.74	700	431	7573	681	44352
	Num of Cycles	210	450	900	1200	2200	2600
PDACO	Best Result	423.74	712	445	7664	696	44619
	Num of Cycles	76	180	290	320	550	850
PIPDMACO	Best Result	423.74	699	426	7542	675	44303
	Num of Cycles	9	8	10	13	28	47
IACS	Best Result	423.74	699	426	7542	675	44303
	Num of Cycles	3	6	12	10	20	52

5 CONCLUSIONS

In this paper, we propose an improved ant colony algorithm for TSPs. According to solving the standard instances and making performance comparison with other algorithms, the proposed algorithm has high efficiency and can also be used for other related optimization problems.

REFERENCES

[1] M. Fischetti, J. J. Salazar, P. Toth, A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Operations Research, Vol.45, 378–94, 1997.

[2] L. V. Snyder, M. S. Daskin, A random-key genetic algorithm for the generalized traveling salesman problem. Operations Research, Vol.173, 38-53, 2006.

[3] W. Darrell, H. Doug and H. Adele, A Hybrid Genetic Algorithm for the Traveling Salesman Problem Using Generalized Partition Crossover. Lecture Notes in Computer Science, Vol.6238, 566-575, 2011.

[4] M. Dorigo, V. Maniezzo, A. Colorni.., Ant System: Optimization by a colony of cooperating agents. IEEE Transaction on Systems, Man and Cybernetics Part B, Vol.26, No.1, 29-41, 1996.

[5] M. Dorigo, L. M. Gambardella, Ant colony System: A cooperative learning approach for the traveling salesman problem. IEEE Transaction on Evolutionary Computation, Vol.1, No.1, 53-66, 1997

[6] T. Stutzle, H. H. Hoos, MAX-MIN ant system and local search for the traveling salesman problem, Proceedings of the IEEE International Conference on Evolutionary Computation, 309-314, 1997.

[7] G. R. Huang, X. B. Cao, X. F. Wang, An ant colony optimization algorithm based on pheromone diffusion, Acta Electronica Sinica, Vol.32, No.4, 865-868, 2004.

[8] J. L. Ding, ZH. Q. Chen, ZH. ZH. Yuan, On the combination of genetic algorithm and ant algorithm. Journal of Computer Research and Development, Vol.40, No.9, 1351-1356, 2003.

[9] Q. B. Zhu, ZH. J. Yang, An ant colony optimization algorithm based on mutation and dynamic pheromone updating. Journal of Software, Vol.15, No.2, 185-192, 2004.

[10] J. H. Yang, X. H. Shi, Maurizio M, Y. CH. Liang, An ant colony method for generalized TSP problem. Progress in Natural Science, Vol.18, 1417-1422, 2008.

[11] J. ZH. Ji, ZH. Huang, CH. N. Liu, A fast ant colony algorithm for traveling salesman problem. Journal of Computer Research and Development, Vol.46, No.6, 968-978, 2009.

[12] C. Blum, M. Dorigo, The hyper- cube framework for ant colony optimization. IEEE Transaction on Systems, Man and Cybernetics, Vol.34, No.2, 1161-1172, 2004.

[13] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/