Proceedings of the 2006 IEEE
International Conference on Information Acquisition
August 20 - 23, 2006, Weihai, Shandong, China

Time Ant Colony Algorithm with Genetic Algorithms

Hong-hao Zuo
Institute of Intelligent Machines,
Chinese Academy of Sciences;
Department of Automation,
University of Science and Technology of China
P.o. Box 1130, He Fei, An Hui, China
zuohefei@mail.hf.ah.cn, zuohh@ustc.edu

Fan-lun Xiong
Institute of Intelligent Machines,
Chinese Academy of Sciences
P.o. Box 1130, He Fei, An Hui, China
flxiong@tom.com

Abstract – Time ant colony algorithm has good effect on combinatorial optimization problems as well as that of the ant colony algorithm while it has the shortcoming of long convergence time. A new method combined with genetic algorithms is proposed. Firstly a genetic algorithms procedure is used to solve the problem in specifying time. Secondly the solution having gotten is used to distribute the original pheromone. At the last the time ant colony algorithm is used to search the optimal solution, which supposed that each ant's velocity is the same and all ants are crawling in full time. The new method accelerates the convergence speed. It is testified by the experiment that the novel algorithm is better than before.

Index Terms – Time ant colony algorithm, genetic algorithms, traveling salesman problem.

I. INTRODUTCTION

Ant Colony Algorithm (ACA)[1,2] is a kind of evolution computation algorithm based on bionics, which has the character of positive feedback and parallel processing. It was first put forward to solve the famous traveling salesman problem(TSP)[3,4]. Now it has been successfully used to solve all kinds of combination optimization problems such as quadratic assignment, job-shop scheduling and so on. Although ACA has been successfully used in many fields, it is inconvenient for rookies because many parameters need to be adjusted in its application. Time ant colony algorithm(TACA) [5] is put forward and it is a improved version of ACA. It is supposed that each ant's velocity is the same and all ants are crawling in full time. Ants communicate with others by the pheromone that is left on the road. It is testified by the experiment that the novel algorithm is as well as other ant colony algorithm and it is simpler to justify the parameters than before.

Genetic Algorithms(GA) has the capability of quick global research while it can't use the information of system's feedback. This character leads many useless redundancy iteration and low efficiency. TACA has the character of distribute, parallel and global convergence by the accumulation and updating of pheromone. Being lack of pheromone information during the starting period, it often has long search time.

In order to overcome the limitation of each algorithm and make advantage mutual-complementary, a new method is proposed. Firstly, Firstly a genetic algorithms procedure is used to solve the problem in specifying time, which has the character of quick global research. Secondly the solution having gotten is used to distribute the original pheromone. At the last TACA is used to search the optimal solution, which has the character of positive feedback and parallel. The new method is better than TACA in time efficiency, which can find better solution in specifying time.

It is arranged below: In section II time ant colony algorithm is described and some characters of algorithm is narrated. In section III the genetic algorithms used in new method is introduced. In section IV TACA with genetic algorithm is introduced. In section V the proof of convergence of new method is given. In section VI the comparative experimental results and the conclusion are given.

II. TIME ANT COLONY ALGORITHM

In according to thinking the real foraging process, that is all ants are working in full time, a novel ant colony algorithm based on time model is proposed, in which all ants are crawling in full time. It has been successfully used in n-cities TSP.

A. Basic Procedure

The hypothesis of this algorithm is as below:

D={ $d_{ij} \mid C_i, C_j \in C$, $1 \le i, j \le n$ } is the set of distances between cities and n is the number of cities. The shortest distance of set D is marked as d_{min}. All the ants are crawling in all time and their velocity is equal to d_{min} per time unit. The time of being decision and so on is omitted. Each ant travels the same distance in each cycle which is always d_{min}. TabuList$_i$ saves the cities which the ant numbered i have already visited. $\tau_{ij}(t)$ saves the pheromone density on the route between city i and city j at time t.

The pseudo-code is as below:

```
Step 1: Initialization, distribute the ants in the cities and add the city which the ant locate to its corresponding TabuList_i, set all $\tau_{ij}(0) = c$, where $1 \le i, j \le n$ and c is a constant.
Step2: Cycle the operation as below while the result hasn't been convergence or the cycle count is smaller than the given number:
Step3: Do the operation as below for all the ant k, where $1 \le k \le m$, m is the number of ants:
    if(ant numbered k can reach the city i when current cycle ends)
```

```
        reach the city i;// reachcity(i)
    else
        note the distance between the ant and the city i
        when this cycle ends;
```

Step4: Update all the pheromone on all the route according to the volatile coefficient,

$$\tau_{ij}(t) = (1-\rho)\cdot\tau_{ij}(t-1), \qquad (1)$$

cycle to step 2;

Step5: Output the result and the program ends.

The pseudo-code of function reachcity(i) is described below:

step1. Add the city i to the route list $TabuList_k$.

step2.Calculate the path length which have been visited, that is

$$Cost_k(t) = d_{ij} + Cost_k(t-1), \qquad (2)$$

where $Cost_k(t)$ is the cost of the ant numbered k at time t.

step3. Release the fix quantity pheromone on the route between the two latest visited city, nCurrentCity and nNextCity.

$$\tau_{ij}(t) = C + (1-\rho)\cdot\tau_{ij}(t-1), \qquad (3)$$

where C is a constant.

step4. Make the if-else decision as below:

```
If(the ant's tour has finished(all the cities have been visited))
{
    prepare for the next tour, clear the route list
        $TabuList_k$, add city i to $TabuList_k$;
    set the distance which have traveled zero,
        $Cost_k(t)=0$;
}
else
    There is no operation.
```

step5. Choose the next city nNextCity according to the state transition probability $P_{ij}^k(t)$, which will be traveled in the next cycle.

step6. The ant moves to the target city in the remaining time of this cycle and the distance between the ant and the city nNextCity is noted when this cycle finished.

step7. function reachcity(i) ends.

In the algorithm the state transition probability $P_{ij}^k(t)$ is calculated as below:

$$P_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha}{\sum_{k\in allowed_k}[\tau_{ij}(t)]^\alpha} & if\ j \in allowed_k \\ 0 & otherwise \end{cases} \qquad (4)$$

B. Analysis of Algorithm Complexity

The complexity of step1 is $O(n^2\cdot m)$. The complexity of step3 is $O(m\cdot n^2)$. The complexity of step4 is $O(n^2)$. If the algorithm is finished after cycle NC times, the time complexity of the entire algorithm is $O(NC\cdot m\cdot n^2+NC\cdot n^2)$. Because n is comparative to m, the time complexity of the algorithm is $O(NC\cdot n^3)$, which is the same as basic ant colony algorithm(abbr. as BACA).

C. Explain on Parameters

Parameters used in BACA are the number of ants m and α, β, η related to transition probability and volatile coefficient ρ and total quantity Q related to pheromone[1]. There are less parameters used in novel time model ant colony algorithm than that of BACA, which are the number of ants m, α related to transition probability and volatile coefficient ρ and constant C. Because of less parameters the novel algorithm can be easily grasped by rookies. The time unit being not the same as that of BACE, there is more cycle for the ant to reach the next city than before. The pheromone being gotten more effect, $\rho<0.1$ is been commended.

III. Genetic Algorithms

The usual form of genetic algorithm was described by Mitsuo Gen[6] as below:

```
Procedure: Genetic Algorithms
begin
    t←0;
    initialize P(t); %let P(t) be parents in current generation t
    evaluate P(t);
    while (not termination condition) do
        recombine P(t) to yield C(t);
        evaluate C(t);  %let C(t) be offspring
                        %in current generation t.
        select P(t+1) from P(t) and C(t);
        t←t+1;
    end
end
```

Genetic algorithms are stochastic search techniques based on the mechanism of natural selection and natural genetics. Genetic algorithms start with an initial set of random solutions called population. Each individual in the population is called a chromosome, representing a solution to the problem. A chromosome is a string of symbols, which is usually a binary bit string. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measures of fitness. To create the next generation, new chromosomes, called offspring, are formed by either (a) merging two chromosomes from current generation using a crossover operator or (b) rejecting others so as to keep the population size constant. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimum of suboptimal solution to the problem.

IV. Entire Algorithm Procedure

There are some modification on the base of TACA and the new method GATimeACA(GATACA) was born. In the first step function GADistribute is used to distribute the original pheromone. The others is the same to the TACA. The entire algorithm procedure is as below:

```
Procedure GATIMEACA
Step 1: Function GADistribute is used to distribute the original pheromone. distribute the ants in the cities and add the city which the ant locate to its corresponding TabuList_i.
Step2: Cycle the operation as below while the result hasn't been convergence or the cycle count is smaller than the given number:
Step3: Do the operation as below for all the ant k, where $1 \le k \le m$, m is the number of ants:
    if(ant numbered k can reach the city i when current cycle ends)
        reach the city i;// reachcity(i)
    else
        note the distance between the ant and the city i when this cycle ends;
Step4: Update all the pheromone on all the route according to the volatile coefficient,
    $\tau_{ij}(t) = (1-\rho)\cdot\tau_{ij}(t-1)$
    cycle to step 2;
Step5: Output the result and the program ends.
```

The pseudo-code of function GADistribute() is described below:

```
step1. Initialization: create $P_0$ randomly, which is coded in real number;
step2. Evaluation: evaluate the population of generation n ($P_n$), the function of fitness is the reciprocal of total tour length;
step3. if the generation is equal to 30 then goto step9;
step4. select the individuals to crossover, in which the chromosome string (i,j) and (i+1,j+1) are used to replace the original string (i,i+1) and (j,j+1);
step5. evaluate the expected offspring;
step6. select P(n+1) from P(n) and C(n);
step7. n = n+1;
step8. goto step2;
step9. a group of optimum solution is produced, which are several trip route;
step10. The ants crawling on the got optimum route and dispose the fixed pheromone C on it. The original pheromone is distributed.
    $\tau_{ij}= 0$
    for (m ants) do
    {
        $\tau_{ij}= C + \tau_{ij}$ ,which i,j is belong to the set of
                optimum route
    }
step11. Function end.
```

V. Proof of convergence

Imitating the method of [7], the proof of convergence is given below:

Proposition 1 For any τ_{ij} it holds:

$$\lim_{\theta\to\infty}\tau_{ij}(\theta)<\tau_{\max}=\frac{mC}{\rho}$$

Proof The maximum possible amount of pheromone added to any arc (i, j) after any iteration is $q_f(s^*)$. In this algorithm $q_f(s^*)$ is equal to mC, which means all ants passed through the arc (i, j) and reached the city numbered j. Clearly, at iteration 1 the maximum possible pheromone trail is

$$\tau_{ij}(1)=(1-\rho)\tau_0+Q(1)\le(1-\rho)\tau_0+q_f(s^*), \tag{5}$$

where Q(1) is the amount of pheromone added to arc (i, j) after iteration 1.

At iteration 2 it is

$$\tau_{ij}(2)=(1-\rho)\tau_1+Q(2)\le(1-\rho)\tau_1+q_f(s^*) \tag{6}$$

$$\le(1-\rho)^2\tau_0+(1-\rho)q_f(s^*)+q_f(s^*), \tag{7}$$

where Q(2) is the amount of pheromone added to arc (i, j) after iteration 2 and so on. Hence, due to pheromone evaporation, the pheromone trail at iteration θ is bounded by

$$\tau_{ij}(\theta)=(1-\rho)\tau_{ij}(\theta-1)+Q(\theta-1) \tag{8}$$

$$\le(1-\rho)\tau_{ij}(\theta-1)+q_f(s^*) \tag{9}$$

$$\le(1-\rho)^{\theta}\tau_0+\sum_{i=1}^{\theta}(1-\rho)^{\theta-i}q_f(s^*) \tag{10}$$

$$\text{Let } \tau_{\max}=\lim_{\theta\to\infty}(1-\rho)^{\theta}\tau_0+\sum_{i=1}^{\theta}(1-\rho)^{\theta-i}q_f(s^*) \tag{11}$$

As $0<\rho\le 1$, then

$$\lim_{\theta\to\infty}(1-\rho)^{\theta}\tau_0=0, \tag{12}$$

$$\lim_{\theta\to\infty}\sum_{i=1}^{\theta}(1-\rho)^{\theta-i}=1/\rho, \tag{13}$$

equation 16 converges asymptotically to

$$\tau_{\max}=\frac{q_f(s^*)}{\rho}. \tag{14}$$

While in the algorithm it is impossible that the amount of pheromone added to any arc (i, j) after each iteration is always $q_f(s^*)$, then from (9) to (10) the token = should be deleted. That is

$$\lim_{\theta\to\infty}\tau_{ij}(\theta)<\tau_{\max}=\frac{mC}{\rho}. \tag{15}$$

Proof end.

Theorem Let $P^*(\theta)$ be the probability that the algorithm finds an optimal solution at least once within the first θ iterations. Then, for an arbitrarily small $\varepsilon>0$ and for a sufficiently large θ it holds that

$$P^*(\theta)\ge 1-\varepsilon.$$

and, by definition,

$$\lim_{\theta\to\infty} P^*(\theta) = 1$$

Proof Due to the pheromone trail limits and $\tau_{\min}$ and $\tau_{\max}$ we can guarantee that any feasible choice in (9) for any partial solution x_h is made with a probability $p_{\min} > 0$. A trivial lower bound for $p_{\min}$ is given by

$$p_{\min} \geq \hat{p}_{\min} = \frac{\tau_{\min}^{\alpha}}{(n-1)\tau_{\max}^{\alpha} + \tau_{\min}^{\alpha}}, \quad (16)$$

where n is the cardinality of the set C of components.(for the derivation of this bound we consider the following "worst-case" situation: the pheromone trail associated with the desired decision is $\tau_{\min}$, while all the other feasible choices – there are at most n-1---have an associated pheromone trail of $\tau_{\max}$) Then , any generic solution s', including any optimal solution $s^* \in S^*$ (S^* is a non-empty set of optimal solutions),can be generated with a probability $\hat{p} \geq \hat{p}_{\min} > 0$, where $n < +\infty$ is the maximum length of a sequence. Because it is sufficient that one ant finds an optimal solution,

$$P^*(\theta) = 1 - \prod_{i=1}^{\theta}(1-\hat{p}_i) \quad (17)$$

$$\geq 1 - (1-\hat{p}_{\min})^{\theta} = \hat{P}^*(\theta) \quad (18)$$

Because of $1 > \hat{p}_{\min} > 0$, by choosing a sufficiently large θ this probability can be made larger than any value $1-\varepsilon$. Hence, we have that

$$\lim_{\theta\to\infty} \hat{P}^*(\theta) = 1, \quad (19)$$

then

$$\lim_{\theta\to\infty} P^*(\theta) = 1. \quad (20)$$

Proof end.

VI. Experiment and Conclusion

To testify the validity of the novel algorithm some special TSP experiment was taken. In Fig. 2 the distance between two neighboring cities is 10 and the distance between two cities can be calculated by geometry in which (a) is for 9-cities TSP, (b) is for 16-cities TSP and (c) is for 25-cities TSP. Later the novel algorithm was used to solve the problem: CTSP31 [8,9], CHN144[10], CHN150[11], TSP225[11] and GR666[11]. The result is in table 1[12,5].

The result shows that performance of TACA is as well as that of BACA. Being more occasion to find the global optimization, TACAE and GATACA has successfully been used in several TSP and its performance is better than that of BACA and TACA. It is testified as a novel effective optimization method. The further work is its application in real time simulation and parallel computing.

Acknowledgment

Zuo H.H. thanks his collogue Sa Li, Bai Shilei and Ding Jing for discussing some issues about this paper.

TABLE 1
EXPERIMENT RESULT

Algorithm \ Datasets		CTSP31	CHN144	CHN150	GR666
BACA	Average of 20 experiments	4.2264	30472.75	6558.9	294763.7
	Optimization found	4.1632	30355.12	6527	294358
TACA	Average of 20 experiments	4.2009	30450.00	6552.95	294698.8
	Optimization found	4.1662	30355.12	6527	294358
GATACA	Average of 20 experiments	4.1945	30435.83	6548.8	294631.5
	Optimization found	4.1588	30355.12	6527	294358

References

[1] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies", Proc. First European Conference on Artificial Life, Paris, France, 1992, pp. 134-142.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 26, no. 1, pp. 29-41, 1996.

[3] M. Dorigo and L.M. Gambardella, "Ant colonies for the traveling salesman problem", BioSystems, vol. 43, pp. 73-81, 1997.

[4] Marco Dorigo and Luca Maria Gambardella, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J],IEEE Transaction On Evolutionary Computation, 1(1):53~66, 1997.

[5] Zuo Honghao, Xiong Fanlun, A novel ant colony algorithm based on time model, Journal of Pattern Recognition and Artificial Intelligence(China), in press.

[6] Mitsuo Gen and Runwei Cheng, Genetic Algorithms and Engineering Optimization, Wiley, New York, 2000.

[7] Marco Dorigo, Thomas Stützle. Ant Colony Optimization, pp.128-130. London, England, The MIT Press, 2004.

[8] Jin Pan, Fan Junbo and Tan Yongdong. Neural Network and Neural

Computer, 375-377. Chengdu, China: Southwest Transportation University Publication, 1991.

[9] Jin Pan. Basic of Neural Computational Intelligence, 300-308. Chengdu, China: Southwest Transportation University Publication, 2000.

[10] Kang Lishan, Xie Yun, You Shiyong, et al., Non-Numerical Parallel Algorithm, Simulated Annealing Algorithm. BeiJing, China: Science Publishing House, 1994, pp150~151.

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

[12] Zuo Honghao, Xiong Fanlun, Novel ant colony algorithm based on local optima clustering and its application in Chinese traveling salesman problem, Journal of Tongji University (China), vol.32, no. SUPPL, pp. 142-144, October, 2004.