LETTER

Ant Colony Optimization with Genetic Operation and Its Application to Traveling Salesman Problem

Rong-Long WANG[†a)], *Member*, Xiao-Fan ZHOU[†], *Nonmember*, *and* Kozo OKAZAKI[†], *Member*

SUMMARY Ant colony optimization (ACO) algorithms are a recently developed, population-based approach which has been successfully applied to optimization problems. However, in the ACO algorithms it is difficult to adjust the balance between intensification and diversification and thus the performance is not always very well. In this work, we propose an improved ACO algorithm in which some of ants can evolve by performing genetic operation, and the balance between intensification and diversification can be adjusted by numbers of ants which perform genetic operation. The proposed algorithm is tested by simulating the Traveling Salesman Problem (TSP). Experimental studies show that the proposed ACO algorithm with genetic operation has superior performance when compared to other existing ACO algorithms.

key words: *ant colony optimization, combinatorial optimization problems, traveling salesman problem, genetic algorithm, genetic operation*

1. Introduction

Combinatorial optimization problems are of high importance both for the industrial world as well as for the scientific world. It arises in many different fields such as economy, commerce, engineering, industry or medicine. Ant colony optimization (ACO) is one of the most recent techniques for solving combinatorial optimization problems, and has been unexpectedly successful in recent years [1], [2]. The inspiring source of ACO algorithms are real ant colonies. More specifically, ACO is inspired by the ants' foraging behavior. At the core of this behavior is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short paths between their nest and food sources. This characteristic of real ant colonies is exploited in ACO algorithms in order to solve, for example, discrete optimization problems.

The first ACO algorithm, called Ant System (AS) was proposed by Dorigo [3] in 1992. Since then, the ACO algorithm attracted the attention of more researchers and a number of other ACO algorithms were introduced. The first improvement over AS was obtained by the Elitist AS (EAS) [4], which is obtained by instantiating pheromone update rule, that is, by using all the solutions that were generated in the respective iteration, and in addition the best-so-far solution, for updating the pheromone values. Ant Colony System (ACS) [5], [6] developed by Dorigo adopts a modified selection rule called pseudo random proportional rule which favors transitions towards nodes connected by short edges and with a large amount of pheromone. ACS also differs from AS in that the updating rule is applied only to edges belonging to the global best tour. Rank-based AS was introduced in [7]. In Rank-based AS, the best-so-far solution has the highest influence on the pheromone update, while a selection of the best solutions constructed at that current iteration influences the update depending on their ranks. Another improved version is Max-Min Ant System (MMAS) [8], [9], which is characterized as follows. Possible trail values are restricted to the interval $[\tau_{max}, \tau_{min}]$, where these two parameters are set up in a problem dependent way. This modification prevents ants from converging to local optimum. MMAS also adopts a concept of elitism in which only the best ant at each iteration updates trails. These are the main differences between MMAS and AS, aiming to achieve good balance between exploitation and exploration. Despite these improvements, the balance between intensification and diversification is still the most important theme in the study of ACO algorithm.

Hybridization is nowadays recognized to be an essential aspect of high performing algorithms. Pure algorithms are almost always inferior to hybridizations. The earliest type of hybridization was the incorporation of local search based methods such as local search, tabu search, or iterated local search, into ACO. However, these hybridizations often reach their limits when highly constrained problems for which it is difficult to find feasible solutions are concerned. Some researchers investigated the incorporation the other population-based search algorithms into ACO algorithms. Pilat and White [10] proposed two modified ACS by using genetic algorithm (GA). Their first algorithm is a hybrid between ACS-TSP and a GA (called ACSGA-TSP) that encodes experimental variables in ants. However their simulation results found that the algorithm does not perform as well as ACS-TSP with respect to finding the optimal solution. Their second algorithm uses a GA to evolve the values of experimental variable used in ACS. By using the algorithm, they found the good values of variables used in ACS. Acan [11] proposed a hybrid algorithm GAACO, which combines GA and ACO. In their GAACO algorithm, GA and ACO work parallel within the same environment to solve the problem. When one of the algorithms finds a better solution, migration occurs between two algorithms. This information share can improve the convergence speed of algorithms. However, these hybrid algorithms have not the ability of controlling the balance between intensification and diversification in search process.

Manuscript received September 4, 2008.

[†]The authors are with the Faculty of Engineering, University of Fukui, Fukui-shi, 910-8507 Japan.

a) E-mail: wang@u-fukui.ac.jp

DOI: 10.1587/transfun.E92.A.1368

Copyright © 2009 The Institute of Electronics, Information and Communication Engineers

In this paper, focusing on the balance between intensification and diversification, we propose an improved ACO algorithm with genetic operation. Different to the ACSGA-TSP and the GAACO, in the proposed ACO algorithm, GA is not used to searching the values of parameters or speeding up the convergence of the ACO algorithm. In the proposed ACO algorithm, genetic operation is performed on some of ants to provide the diversity and help the ACO algorithm to move out of local optima. Additionally, the proposed ACO algorithm also provides a mechanism for adjusting the balance between intensification and diversification by controlling the numbers of ants which perform genetic operation. To evaluate the performance of the proposed improved ACO algorithm, we simulate some TSPLIB benchmark problems [12]. The simulation results show that the proposed algorithm produces better results over the other existing ACO algorithm.

2. Ant Colony Optimization

The first ACO algorithm, called Ant System (AS) was firstly applied to the traveling salesman problem (TSP). In TSP, a given set of n cities has to be traversed so that every city is visited exactly once and the tour ends in the initial city. The optimization goal is to find a shortest possible tour. Let d_{ij} be the distance between city i and city j and the τ_{ij} the amount of pheromone in the edge that connects i and j. Each of m ants decides independently on the city to be visited next based on the intensity of pheromone trail τ_{ij} and a heuristic value η_{ij}, until the tour is completed. Each ant is placed on a random start city, and builds a solution going from city to city, until it has visited all of them. The probability that an ant k in a city i chooses to go to a city j next is given by:

$$p_{ij} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum\limits_{l \in J_i^k(t)} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}} & if\ j \in J_i^k(t) \\ 0 & otherwise \end{cases} \tag{1}$$

where the heuristic value $\eta_{ij} = 1/d_{ij}$, the parameters α and β determine the relative influence of pheromone and distance. J_i^k is the set of cities that remain to be visited by ant k positioned on city i. Once all ants have built a tour, ants perform following pheromone update rule:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{2}$$

Equation (2) consists of two parts. The left part makes the pheromone on all edges decay. The speed of this decay is defined by ρ, the evaporation parameter. The right part increases the pheromone on all the edges visited by ants. The amount of pheromone an ant k deposits on an edge (i,j) is defined by $L_k(t)$, the length of the tour created by that ant at iteration t.

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L^k(t)} & edge(i,j)\ is\ used\ by\ ant\ k \\ 0 & otherwise \end{cases} \tag{3}$$

where the Q is constant. In this way, the increase of pheromone for an edge depends on the number of ants that use this edge, and on the quality of the solutions found by those ants.

Even though the original AS algorithm achieved encouraging results for the TSP problem, it was found to be inferior to state-of-the-art algorithms for the TSP as well as for other problems. Therefore, several extensions and improvements of the original AS algorithm were introduced over the years. One of the famous extensions is the Rank-based AS [7]. In Rank-based AS, always the global-best tour is used to update the pheromone trails. Additionally, a number of best ants of the current iteration are allowed to add pheromone. To this aim the ants are sorted by tour length, and the quantity of pheromone an ant may deposit is weighted according to the rank r of the ant. Only the $(w-1)$ best ants of each iteration are allowed to deposit pheromone. The global best solution, which gives the strongest feedback, is given weight w. The rth best ant of the current iteration contributes to pheromone updating with a weight given by $max\{0, w-r\}$. Thus the improved update rule is:

$$\begin{aligned} \tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) &+ \sum_{r=1}^{w-1} (w-r) \cdot \Delta\tau_{ij}^{r}(t) \\ &+ w \cdot \Delta\tau_{ij}^{gb}(t). \end{aligned} \tag{4}$$

where $\Delta\tau_{ij}^{r}(t) = Q/L^{r}(t)$ and $\Delta\tau_{ij}^{gb}(t) = Q/L^{gb}(t)$.

3. ACO with Genetic Operation

The same as in other heuristic algorithm, one of the most important themes in the study of ACO algorithm is the balance between intensification and diversification. In detail, too much emphasis on the intensification can make ants converge to a local minimum and too much emphasis on the diversification can cause an unstable state. In Sect. 2, we surveyed the original ACO algorithm and some improvements. We note that most of existing ACO algorithms are aiming to adjust the diversity in depositing pheromone. However it is difficult to control the balance between intensification and diversification using only adjusting the pheromone. In this section, we propose an improved ACO algorithm in which genetic operation is performed on some of ants to provide the diversity, and by controlling the numbers of ants which perform genetic operation the balance between intensification and diversification can be adjusted directly.

The outline of proposed improved ACO is depicted in Fig. 1. Let N_t denotes the total number of ants. First we randomly divide N_t ants into two groups (G_1 and G_2). Then the ants in G_1 build solutions based on the intensity of pheromone trail τ_{ij} and a heuristic value η_{ij} in ACO algorithm. The same as in the original ACO, each ant is placed on a random start city, and builds a solution going from city to city, until it has visited all of them. The probability that an ant k in a city i chooses to go to a city j next is given by Eq. (1). On the other hand, each ant in G_2 corresponds to a solution of TSP, and these ants evolve by performing genetic operation which is crossover in genetic algorithm (GA). Our approach for representing chromosome of an ant (solution) in G_2 is to simply list the order in which the cites are visited. Some crossover methods were proposed for the TSP, such as order 1 crossover (OX1) [13], position crossover (PX) [14], asexual crossover (AX) [15]. AX crossover uses a single chromosome as a parent. Two crossover points are selected at random and the elements at those points are swapped. In this work, we use AX crossover for the TSP.

Once all ants in G_1 and G_2 have built a tour, the pheromone information will be updated. In the proposed method, we propose a pheromone update procedure which is modified from that of Rank-based AS [7]. Only w_1 "elite ants" in G_1 and w_2 "elite ants" in G_2 are allowed to deposit pheromone based on tour length. At the same time, the global best tour from the beginning of the trial is also updated in consideration of exploitation. The updating rule is described as follows:

$$\begin{aligned}\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) & \\ + \sum_{r=1}^{w_1-1}(w_1-r)\cdot\Delta\tau_{ij}^{r,G_1}(t) & \\ + \sum_{r=1}^{w_2-1}(w_2-r)\cdot\Delta\tau_{ij}^{r,G_2}(t) & \\ + w_1\cdot\Delta\tau_{ij}^{gb}(t) & \end{aligned} \tag{5}$$

$$\Delta\tau_{ij}^{r,G_1}(t) = \begin{cases} Q/L^{r,G_1}(t) & if\ (i,j)\in T^{r,G_1}(t) \\ 0 & otherwise \end{cases} \tag{6}$$

$$\Delta\tau_{ij}^{r,G_2}(t) = \begin{cases} Q/L^{r,G_2}(t) & if\ (i,j)\in T^{r,G_2}(t) \\ 0 & otherwise \end{cases} \tag{7}$$

$$\Delta\tau_{ij}^{gb}(t) = \begin{cases} Q/L^{gb}(t) & if\ (i,j)\in T^{gb}(t) \\ 0 & otherwise \end{cases} \tag{8}$$

where Q is constant, $L^{r,G_1}(t)$ and $L^{r,G_2}(t)$ are the length of the tour generated by the rth best ant in G_1 and G_2 at iteration t, $T^{r,G_1}(t)$ and $T^{r,G_2}(t)$ are the set of edges constituting the tour, drespectively. L^{gb} is the length of the tour generated by the global best ant and T^{gb} is the set of edges constituting it. In Eq. (5), the parameter $\rho \in (0, 1]$ is a parameter that regulates the pheromone evaporation. After the pheromone information is updated, the ants are regrouped into G_1 and G_2. The above procedure is repeated a predefined number of times.

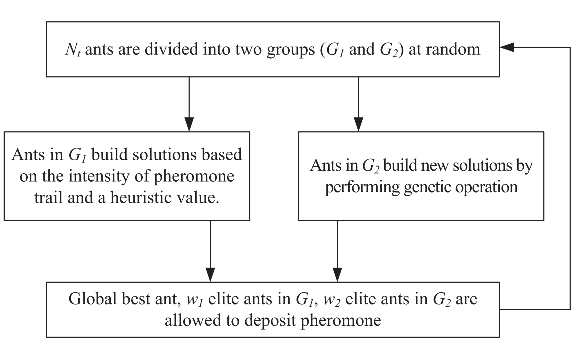

Fig. 1 Outline of proposed improved ACO.

In the proposed ACO algorithm, genetic operation is conducted to provide the diversity of solutions and adjust the balance between intensification and diversification. For this reason, when we divide ant into two groups, we set the number of ant in G_2 according to the following equation.

$$N_{G_2} = \frac{N_t}{2}\cdot\mu^t \tag{9}$$

where N_t is the total numbers of ants, $0 < \mu < 1$ is a parameter and t denotes the t# iteration. From the above equation, we can know that the numbers (N_{G_2}) of ants in G_2 is set to $N_t/2$ as initial value, as the procedure of the proposed ACO algorithm, it is decreased slowly. Clearly, at the earlier stage, N_{G_2} has a relative large value, and many ants perform genetic operation. Thus at this stage, the proposed ACO algorithm could give more diversity and search global optimal solution more efficiently than the conventional ACO algorithm. As the procedure of the proposed ACO algorithm, N_{G_2} becomes smaller and smaller and the numbers of the ants which perform genetic operation also become smaller and smaller. Thus the proposed ACO algorithm could have good local search ability at the final stage. The value of μ controls the speed of decreasing the numbers of ants in G_2, thus by selecting the parameter μ, we can adjust the balance between intensification and diversification in the search process. We note that by introducing genetic operation into ACO algorithm, although the balance between intensification and diversification can be adjusted by selecting the parameter μ and the premature convergence of the algorithm can be avoided, the running time of algorithm also becomes longer than the original ACO algorithm. Besides, it is also worth noting that although the proposed ACO algorithm is introduced by using the TSP, it can be used to solve other combinatorial optimization problems.

4. Simulation Results

In order to assess the effectiveness of the proposed ACO algorithm, extensive simulations were carried out over some TSPLIB benchmark problems on a PC Station. The simulation results are shown in Tables 1 and 2. Because the pheromone update procedure of the proposed ACO algorithm is modified from that of Rank-based AS [7], parameters setting suggested in [7] was used in simulations. The number of ants was set to the number of the cities. ACO parameters were set to $\alpha = 1, \beta = 5, \rho = 0.5, Q = 100, w_1 = 6$. The numbers ($w_2$) of "elite ants" in G_2 was set according to the following equation:

$$w_2 = \begin{cases} 6 & N_{G_2} > 6 \\ N_{G_2} & otherwise \end{cases} \tag{10}$$

In the proposed ACO algorithm, parameter μ controls the speed of decreasing the numbers of ants in G_2 and thus, it can adjust the balance between intensification and diversification in the search process. Too small value of μ causes

Table 1 Simulation results.

Problem	Optimum	GA	EP	SA	AS_{rank}	Proposed algorithm
Eil50 (50-city)	425 (N/A)	428 (N/A) [25000]	426 (427.86) [100000]	443 (N/A) [68512]	441 (435.13) [5000]	425 (427.85) [1043]
Eil75 (75-city)	535 (N/A)	545 (N/A) [80000]	542 (549.18) [325000]	580 (N/A) [173250]	564 (566.88) [5000]	535 (542.30) [2699]
KroA100 (100-city)	21282 (N/A)	21761 (N/A) [103000]	N/A (N/A) [N/A]	N/A (N/A) [N/A]	23278 (23599.12) [5000]	21282 21285.44 [2471]

Table 2 Simulation results on Oliver30 using different μ.

μ	Best	Rate (%)	Average
0.8	423.74	4	431.28
0.9	423.74	2	430.63
0.92	423.74	5	429.11
0.94	423.74	2	428.71
0.96	423.74	7	427.43
0.98	423.73	19	425.66
0.99	423.74	27	424.88
0.999	423.74	40	424.31
0.999	423.74	94	423.78

that the decrease of ants in G_2 becomes too rapid and diversification decreases at an early stage. On the other hand, too large value of μ causes that at the final stage, there are still many ants which perform genetic operation. In this case, it will be difficult for the algorithm to have good local search ability at the final stage. To see the influence of parameter μ to the performance, we tested the proposed ACO algorithm on problem Oliver30 including 30 cities (its optimal tour length is 423.74) using different values of parameter μ. For each case, 100 runs were performed. The simulation results are summarized in Table 2. In the Table we give the best solutions, the rates to find the best solution and the average solutions. For the Table, we can know that the proposed ACO algorithm can find optimal solution with high probability when $\mu > 0.98$. It is evident that the range of reasonable value of the μ is very larger. In following simulations, we set the value of μ with 0.999.

To see the ability of global search and local converge, we tested the proposed ACO algorithm on some TSPLIB benchmark problems. For each of instances, 100 simulation runs were performed. The results are shown in Table 1 where the results produced by the original Rank-based AS (ASrank) [7], genetic algorithm (GA), evolutionary programming (EP), simulated annealing (SA) are also listed for comparison. We report the best integer tour length, the best real tour length (in parentheses) and the number of tours required to find the best integer tour length (in square brackets). The difference between integer and real tour length is that in the first case distances between cities are measured by integer numbers, while in the second case by floating point approximations of real numbers. Note that for the original Rank-based AS [7], the same parameter setting as in the proposed algorithm was used and the results using GA, EP and SA are from [6]. For the Table, it is clear that the proposed ACO algorithm outperforms other algorithms in both the solution quality and the computation cost.

5. Conclusions

An improved ACO algorithm with genetic operation for efficiently solving combinatorial optimization problems was proposed in this paper. In the proposed ACO algorithm, some of ants can evolve by performing genetic operation, and the balance between intensification and diversification can be adjusted by numbers of ants which perform genetic operation. The proposed ACO algorithm was evaluated experimentally through simulating some TSP benchmark problems. From the simulation results, we noted that the proposed improved ACO has superior performance compared with original Rank-based AS and some other methods. It is worth noting that the idea adjusting the balance between intensification and diversification can also be applied to other ACO algorithms.

References

[1] M. Dorigo and T. Stutzle, Ant Colony Optimization, The MIT Press, 2004.
[2] C. Blum, "Ant colony optimization: Introduction and recent trends," Physics of Life Reviews, vol.2, no.4, pp.353–373, 2005.
[3] M. Dorigo, Optimization, learning and natural algorithms, PhD Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
[4] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," IEEE Trans. Syst. Man Cybern., vol.26, no.1, pp.29–41, 1996.
[5] M. Dorigo and L.M. Gambardella, "Ant colonies for the traveling salesman problem," BioSystems, vol.43, no.2, pp.73–81, 1997.
[6] M. Dorigo and L.M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," IEEE Trans. Evol. Comput., vol.1, no.1, pp.53–66, 1997.
[7] B. Bullnheimer, R. Hartl, and C. Strauss, "A new rank based version of the Ant system," Central European Journal of Operations Research, vol.7, no.1, pp.25–38, 1999.
[8] T. Stutzle and H.H. Hoos, "MAX-MIN ant system," Future Gener. Comput. Syst., vol.16, no.8, pp.889–914, 2000.
[9] T. Stetzle and M. Dorigo, "A short convergence proof for a class of ACO algorithms," IEEE Trans. Evol. Comput., vol.6, no.4, pp.358–365, 2002.
[10] M.L. Pilat and T. White, "Using genetic algorithms to optimize ACS-TSP," Proc. Ant Algorithms: Third International Workshop, ANTS 2002, Brussels, Belgium, Sept. 2002, eds. M. Dorigo et al., Lecture Notes Comput. Sci., vol.2463, pp.282–287, Springer-Verlag, 2002.
[11] A. Acan, "GAACO: A GA + ACO hybrid for faster and better search capability," Proc. Ant Algorithms: Third International Workshop, ANTS 2002, Brussels, Belgium, Sept. 2002, eds. M. Dorigo et al.,

Lect. Notes Comput. Sci., vol.2463, pp.300–301, Springer-Verlag, 2002.
[12] G. Reinelt, "A traveling salesman problem library," ORSA Journal on Computer, vol.3, no.4, pp.376–384, 1991.
[13] L. Davis, "Job shop scheduling with genetic algorithm," Proc. International Conference on Genetic Algorithm, pp.136–140, Lawrence Erlbaum, London, 1985.
[14] T. Starkweather, S. McDaniel, K. Mathias, D. Whitley, and C. Whitley, "A comparison of genetic sequence operators," Proc. 4th Int. Conf. on Genetic Algorithms, pp.69–76, Morgan Kaufmann, 1991.
[15] S. Chatterjee, C. Carrera, and L. Lynch, "Genetic algorithms and traveling salesman problems," Eur. J. Oper. Res., vol.93, pp.490–510, 1996.