2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

An Improved Ant Colony Optimization for Quadratic Assignment Problems

Kenya Jin'no[1], Mari Sato[2], and Kazuyuki Aihara[3,1]

1) Aihara Complexity Modeling Project, ERATO, JST
Room 607, An Bldg., Institute of Industrial Science, The University of Tokyo
4−6−1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
2) Graduate School of Electrical Engineering, Kanto Gakuin University
1-50-1 Mutsuura-higashi, Kanazawa-ku, Yokohama, 236-8501 Japan
3) Institute of Industrial Science, The University of Tokyo
4−6−1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
Email: jinno@ieee.org, m0842006@kanto-gakuin.ac.jp, aihara@sat.t.u-tokyo.ac.jp

Abstract—A large number of meta-heuristic algorithms have been developed for solving various kinds of combinatorial optmization problems. In order to improve the searching ability, we consider that nonlinear dymanics is applied to such algorithms. In this article, we propose an improved ant colony optimization algorithm. The ant colony optimization system can be classified into a kind of multi agent system. If each agent has nonlinear dynamics, the system may improve the ability to search the optimum solution. By using our proposed system, we try to solve quadratic assignment problems.

1. Introduction

Searching for the optimal value of the evaluation function to various problems is very important in the engineering field. Such problem is called optimization problems. In order to solve such optimization problems, various kinds of algorithms have been proposed.

In generally, to search an optimal solution of such optimization problems is required a lot of computation time. Therefore, in order to search speedy, many heuristic optimization algorithms have been proposed, for example, Simulated Anneling, Neural Networks, Genetic Algorithm, Genetic Programming, Particle Swarm Optimization, and so on.

Ant Colony Optimization (abbr. ACO), which was originally proposed by M.Dorigo[1],[2] which is called "Ant System", is one one of such heuristic algorithms. In Ref.[2], the Traveling Salesman Problem, which is a very famous combinatorial optimization problem, is solved by Ant Colony System[2]. The result indicates the Ant Colony System exhibits effective peformance for local optimization[2]. In this article, we consider the system which based on the Ant Colony System, and we call the system "ACO".

The ACO is based on the studies of Swarm Inteligence[3], is simulating an action which searches for the bait of the ant. The principle of ACO algorithm is based on the way ant searches for bait and finds their way back to the nest. The ant leaves a chemical which is called "pheromone" on the graoud. The pheromone configures a trail which leads ants toward to the bait. If an ant finds the shortest trail from the nest to the bait, other ants will follow the trail, and such trail means an optimal solution.

The ACO is very powerful searching algorithm to search an optimal combination which gives a optimal value of its coressponding evaluation function. For this system, each ant can be regarded as an agent, therefore, this algorihm consists with a lot of agents, and the swarm of agents search the optimal combination. By using this algorithm, we try to solve quadratic assignment problems.

2. Quadratic assignment problems

Quadratic Assignments Problems (abbr. QAP) have been introduced by Koopmans and Beckman in 1957[4] is one of combinatorial optimization problems.

The QAP can be described as the problem of assigning a set of facilities to a set of location with given distance between the locations and given flows between the facilities. In particular, the problem consists with n facilities and n locations. Figure 1 shows this situation for $n = 4$. In Figure

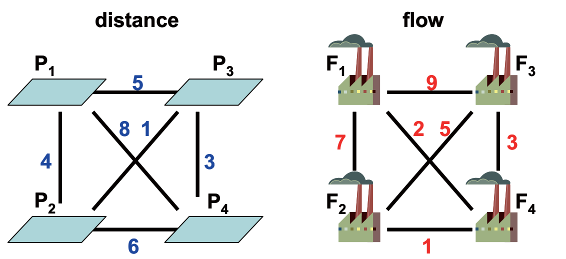


Figure 1: An example of distance and flow ($n = 4$)

1, the left figure denotes the location and each number indicates the distance between the location. Also, each number in the right figure denotes the flow between the facilities.

Figures 2 and 3 show examples of assignments of the facilities. The sum of the products of the flows and distances for the assignment of Figure 2 is 218. On the other hand, the sum for the assignment of Figure 3 is 180. In this case, the assingnment of Figure 3 is an optimal solution which gives the minimal sum.

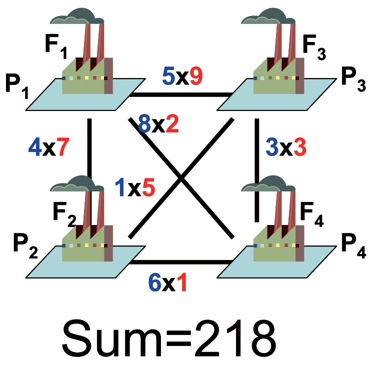


Figure 2: An assignment example.

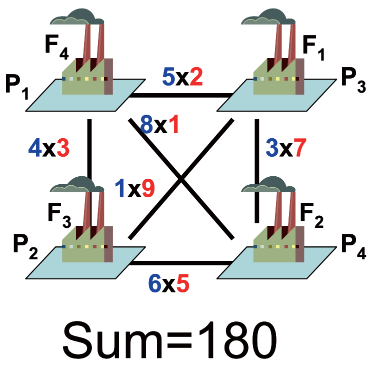


Figure 3: An optimal assignment example.

We can consider two $n \times n$ matrices $\boldsymbol{A} = [a_{ij}]$ and $\boldsymbol{B} = [b_{ij}]$, where a_{ij} denotes the flow between facilities i and j and b_{ij} denotes the distance between location i and j. The purpose of the problem is to find an assignment such that the sum of the products of the flows and distances is the minimal, therefore the objective function $f(\boldsymbol{\pi})$ of the problem can be described as

$$\min_{\boldsymbol{\pi}\in\Pi(n)} f(\boldsymbol{\pi}) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j}, \tag{1}$$

where $\Pi(n)$ is the set of permutation of n elements.

For the problem size n, the corresponding space of solution of the permutation is given by $n!$. The QAP can be classified into NP-Hard problem, therefore, to serach an optimal solution is required a lot of computation time.

Only few combinatorial optimization problems can be solved exaxtly for relatively large instances. QAP, however, is a quite hard to solve, because the QAP instance of size larger than 20 are considered intractable. The use of heuristic algorithm for solving large QAP instance is currently the only practicable solution.

3. Ant Colony Optimization

The algorithm of Ant Colony Optimization (abbr. ACO) is simulating an action which searches for the bait of the ant, and is one of meta-heuristics algorithm which searches an optimum solution of given objective function. The principle of this algorithm is based on the way ant searches for bait and finds their way back to the nest. The ant leaves a chemical which is called "pheromone" on the graoud. A trail is configured by the pheromone. The pheromone trail plays to guide other ants toward to the bait from the nest. The amount of the pheromone depends on the distance between the nest and the location of the bait because the pheromne has a volatile characteristic. Since the pheromone of the shortest trail will become strong, the most of ants select such same trail. As for such strong trail, the possibility to be a best trail is high. The algorithm of ACO uses such property of the action of ants. In this article, we consider that we try to solve QAP by using ACO.

Here, we consider a complete graph G which consists with n verteces V_i as shown in Figure 1.

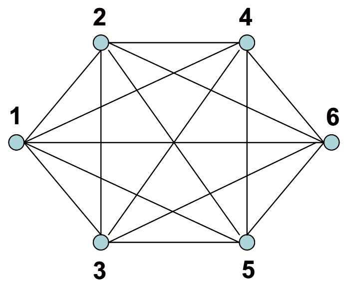
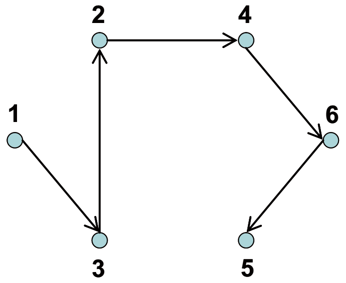


Figure 4: A complete graph ($n = 6$)

Because the graph G is the complete graph, the number of edges of the graph G is $n(n-1)/2$. Each edge denotes E_{ij} which means the edge between i-th vertex V_i and j-th vertex. In order to solve QAP, each element of the distance matrix B of QAP is allocated on the corresponding edge of the graph. Also, the pheromone τ_{ij} is arranged on the corresponding edge E_{ij}. ACO algorithm is applied to the such graph. The algorithm select a trail where an ant visits all vertices only once. In other word, this algorithm outputs a visit order of the vertices. It creates the permutation of the flow of QAP using the visit order, and it computes the evaluation value of the trail of the ant. The visit order denotes a permutation of the flow, then $\pi_i = 1$ means that the first visit vertex is i-th vertex. An example of a visit order of 6 verteces graph is shown in Figure 5. In the case of Figure 5, the permutation is given as $\boldsymbol{\pi} = (1, 3, 2, 4, 6, 5)$.

Since the selection of each edge depends on the amount of pheromone on the trail, the most important component of ACO is the management of the amount of pheromone trail. Initially no information is contained in the pheromone trail, meaning that all pheromone τ_{ij} have an identical value, e.g. $\tau_{ij} = 1$.



$\pi_1 = 1, \pi_2 = 3, \pi_3 = 2, \pi_4 = 4, \pi_5 = 6, \pi_6 = 5$

Figure 5: An example of a tour. $\boldsymbol{\pi} = (1, 3, 2, 4, 6, 5)$

When the m-th ant locates on the i-th vertex, the probability of the ant visits the j-th vertex which is allowed to move from the i-th vertex, P_{ij}^m, is calculated as

$$P_{ij}^m = \begin{cases} \dfrac{\tau_{ij}(\eta_{ij})^\beta}{\sum_{k \in N_i} \tau_{ik}(\eta_{ik})^\beta}, & \text{if } j \in N_i^m \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

where N_i^m means a set of verteces never accessed of m-th ant from i-th vertex. When the ant visits the i-th vertex to the π_i-th, η_{ij} is a parameter which associated with the cost between the i-th vertex and the j-th vertex.

$$\eta_{ij} = \frac{1}{a_{ij} b_{\pi_i \pi_i + 1}}. \qquad (3)$$

$\beta > 0$ is a parameter the relative importance of pheromone versus the cost.

η is adopted as a heuristic parameter, and the parameter improves the searching performance.

Based on above probability, the state transition rule is determined. In the state transition rule of the ACO, two rules are existed. One is an exploitation, other one is a biased exploration. The m-th ant located on the i-th vertex selects the j-th vertex to move to by applying the rule given by

$$j = \begin{cases} \arg\max_{k \in N_i} P_{ik}^m & \text{if } q \leq q_0 (\text{exploitation}) \\ J & \text{otherwise (biased exploration)} \end{cases} \qquad (4)$$

where q is a number which is generated by logistic map, q_0 is a thereshold ($0 \leq q_0 \leq 1$), and J is a random variable in N_i^m selected according to the probability distribution in (2).

Next, we consider the pheromone updating rule. In this article, we consider two pheromone updating rules.

The fisrt updating rule is that the pheromone is made allocation to all edges according to the evaluation value. Once all ants have build their tour, pheromone is updated on all edges according to

$$\tau_{ij} \leftarrow (1 - \alpha)\tau_{ij} + \sum_{k=1}^{M} \Delta_{ij}^k \qquad (5)$$

where M denotes the number of ants, and Δ_{ij}^k means an increment corresponding to each ant which propotionates to its evaluation value.

Δ_{ij}^k is defined as

$$\Delta_{ij}^k = \begin{cases} \dfrac{\gamma_k}{\sum_{j=1}^{M} \gamma_j} \cdot \dfrac{\alpha \sum_{i=1}^{n} \sum_{j=1}^{n} \tau_{ij}}{n} & \text{if } (i, j) \in \text{tour done by ant } k \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

where, γ_k means a parameter for the k-th ant which is determined by its evaluation value.

The γ_k is given by

$$\gamma_k = \frac{E_k - \min_j E_j}{\max_j E_j - \min_j E_j} \qquad (7)$$

where, E_k denotes an evaluation value for the k-th ant which is calculated as

$$E_k = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i^k \pi_j^k} \qquad (8)$$

The second updating rule is that the pheromone is made allocation to edges which comprise the most effective tour.

$$\tau_{ij} \leftarrow (1 - \alpha)\tau_{ij} + \Delta_{ij} \qquad (9)$$

Δ_{ij} is defined as

$$\Delta_{ij} = \begin{cases} \dfrac{\alpha \sum_{i=1}^{n} \sum_{j=1}^{n} \tau_{ij}}{n} & \text{if } (i, j) \in \text{tour done by ant } k \text{, and} \\ & k = \arg\max_j E_j. \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

Note that, using above pheromone updating rules, the total amount of the pheromone must be a constant, e.g $n \times n$.

4. Simulation

First, we confirm the effect of the parameter α which means an attenuation coefficient. Most of the test problems can be found in QAPLIB[5]. We use `Nug14` in the QAPLIB for our simulation. Table 1 shows the result of the numerical experiment. The upper limit of the iteration in the experiments is 100000. "iteration" in Table 1 means the number of iterations when the best evaluation value is obtained. Therefore, if the "iteration" is small, the system converges a steady state speedy, and the parameter α controls the convergence speed. It, however, is said to be the search of the optimal solution is insufficient in converging too early. In order to search efficiently, $\alpha = 0.9999$ is applied to simulations hereafter.

Table 1: The effect of the parameter α for `Nug14` in QAPLIB[5]. "iteration" means the numbet of iterations when the best evaluation value is obtained.

	$\min_k E_k$ (iteration)	
α	$\beta = 0.00$	
0.20000	1138	(10)
0.10000	1072	(35)
0.01000	1064	(477)
0.00100	1040	(5818)
0.00010	1040	(35524)
0.00001	1056	(74335)

Next, we consider the effect of the parameter β and the updating rule. First, we consider the the rule is that the pheromone is made allocation to all edges acoording to the evalution value. Table 2 shows the result that Eq.(5) is applied for the updating rule. As shown in Table 2, the system cannot find an optimal solution within 100000 iterations. In this case, the system exhibits the most effective result around $\beta = 0.5$. Next, we consider the case where the updaing rule is that the pheromone is made allocation to edges which comprise the most effect tour. In this case, the system can find the optimal solution which denotes the boldface number. These results indicate that the second updating rule is more efficient than the first one. According to Ref[2], the heuristic function η is fundamental in making the algorithm find good solutions in a reasonable time. The result seems that $\beta = 0.5$ is the most effective value. Note that, η is configured by the corresponding cost of a part of trail in this article. The system, however, may exhibit more remarkable performance if η can be determined appropriate function. Finding such function is a future problem.

Table 2: The effect of the parameter β for the first updating rule that the pheromone is made allocation to all edges acoording to the evalution value.

	$\min_k E_k$ (iteration)	
β	$\alpha = 0.0001$	$\alpha = 0.0100$
0.0	1056 (74335)	1040 (7785)
0.1	1058 (57610)	1044 (6516)
0.2	1064 (48713)	1050 (6339)
0.3	1050 (36617)	1056 (4267)
0.4	1050 (36617)	1080 (1838)
0.5	1050 (36617)	1094 (2749)
0.6	1050 (36617)	1094 (113)
0.7	1056 (78873)	1096 (1113)
0.8	1056 (78873)	1108 (968)
0.9	1064 (29771)	1104 (425)
1.0	1058 (21872)	1104 (425)
1.5	1074 (36080)	1118 (411)
2.0	1086 (6639)	1128 (50)

Table 3: The effect of the parameter β. Applying update rule is the pheromone is made allocation to edges which comprise the most effect tour. The boldface denotes it is the optimal value.

	$\min_k E_k$ (iteration)	
β	$\alpha = 0.0001$	$\alpha = 0.0100$
0.0	1040 (35524)	1064 (477)
0.1	**1014** (30493)	1044 (251)
0.2	1024 (20592)	1024 (414)
0.3	1024 (26343)	1040 (419)
0.4	**1014** (37176)	1044 (278)
0.5	**1014** (38543)	1024 (339)
0.6	**1014** (39019)	1050 (211)
0.7	**1014** (39019)	1040 (368)
0.8	1024 (27074)	1050 (302)
0.9	1024 (20375)	1050 (267)
1.0	1024 (18015)	1056 (395)
1.5	1050 (15071)	1096 (190)
2.0	1074 (14877)	1096 (227)

5. Conclusion

The ACO consists with many control parameter, then we confirmed the effects of each parameter, and we proposed the effective value of these parameters. The most of other proposed ACO uses another heuristics for local search[2]. Correspondigly, note that our system uses only ACO.

References

[1] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: Optimization by a colony of co-operating agents," *IEEE Trans. Systems, Man and Cybernetics – Part B,* Vol.26, No.1, pp.1-13, 1996.

[2] M. Dorigo, L.M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem," *IEEE Trans. Evolutionary Computation*, Vol.1, pp.53-66, 1997.

[3] E. Bonabeau, M. Dorigo, and G.Theraulaz, "Inspiration for Optimization from Social Insect Behaviour," *Nature* Vol.406, pp.39-42, 2000.

[4] T.C. Koopmans, and M.J. Eckmann, "Assignment Problems and the Location of Economic Activities", Econometrica, vol.25, pp.53-76 , 1957.

[5] R.E. Burkard, E. Cela, S.E. Karisch and F. Rendl,"QAPLIB - A Quadratic Assignment Problem Library, `http://www.seas.upenn.edu/qaplib/`