PAPER

Multiple Ant Colonies Algorithm Based on Colony Level Interactions

Hidenori KAWAMURA†, **Masahito YAMAMOTO**†, ***Members***, **Keiji SUZUKI**†, ***Nonmember***, ***and*** **Azuma OHUCHI**†, ***Member***

SUMMARY Recently, researchers in various fields have shown interest in the behavior of creatures from the viewpoint of adaptiveness and flexibility. Ants, known as social insects, exhibit collective behavior in performing tasks that can not be carried out by an individual ant. In ant colonies, chemical substances, called pheromones, are used as a way to communicate important information on global behavior. For example, ants looking for food lay the way back to their nest with a specific type of pheromone. Other ants can follow the pheromone trail and find their way to baits efficiently. In 1991, Colorni et al. proposed the ant algorithm for Traveling Salesman Problems (TSPs) by using the analogy of such foraging behavior and pheromone communication. In the ant algorithm, there is a colony consisting of many simple ant agents that continuously visit TSP cities with opinions to prefer subtours connecting near cities and they lay strong pheromones. The ants completing their tours lay pheromones of various intensities with passed subtours according to distances. Namely, subtours in TSP tourns that have the possibility of being better tend to have strong pheromones, so the ant agents specify good regions in the search space by using this positive feedback mechanism. In this paper, we propose a multiple ant colonies algorithm that has been extended from the ant algorithm. This algorithm has several ant colonies for solving a TSP, while the original has only a single ant colony. Moreover, two kinds of pheromone effects, positive and negative pheromone effects, are introduced as the colony-level interactions. As a result of colony-level interactions, the colonies can exchange good schemata for solving a problem and can maintain their own variation in the search process. The proposed algorithm shows better performance than the original algorithm with almost the same agent strategy used in both algorithms except for the introduction of colony-level interactions.

key words: *multi-agent system, ant algorithm, traveling salesman problems, combinatorial optimization problems*

1. Introduction

Thre are many kinds of creatures that show adaptive and flexible behavior to achieve various tasks. Ants, so-called social insects, are one example. Although ants have simple abilities, they exhibit collective behavior to perform tasks that can not be carried out by one individual, such as foraging, building a nest, caring for offspring, and defending their colony from enemies. As a result of cooperation in many kinds of micro-scale behavior, macro-scale complex behavior seems to emerge without any central or hierarchical control. In ant colonies, chemical substances called pheromones are used as a way of communicating important information concerning their colonies [1]. For example, ants looking for food lay the way back to their nest with a specific type of pheromone. Other ants can follow the pheromone trail and find their way to baits efficiently.

Studies on the behavior of ants in colonies and on pheromone communication may be useful for the development of artificial intelligence. Researchers in various fields have studied the behavior of ants. Nakamura, Suzuki and Mikami studied on multiagent systems based on the behavior of ants and their organization ability based on pheromone-style communication [14], [16], [20]. For optimizations, we have constructed multiagent-based optimization algorithms based on pheromone-style communication for Vehicle Routing Problems and Nurse Scheduling Problems [11], [21]. Subramanian proposed distributed routing algorithms based on simple biological ants for routing in packet-swiched communication networks [19]. Caro also proposed an approach using mobile agents based on the AntNet for adaptive communication networks routing [3]. Kuntz applied ant-like agents to partitioning problems in VLSI technology [12].

The ant algorithm, originally proposed by Colorni et al. in 1991 for Traveling Salesman Problems (TSPs) [4], [5], [8], [9], is also one of optimization algorithms inspired from the analogy of the foraging behavior of ants and interactions between ants in a colony. The optimization of the ant algorithm is based on low-level interactions among a large number of cooperating simple agents, like ants, which are not aware of their cooperative behavior. To solve TSPs, the ant agents continuously move from one city to another unvisited city in a TSP with opinions to prefer subtours connecting near cities and they lay strong pheromones. The ant agents visiting all cities lay some intensity of the pheromones with subtours included in their completed tours according to the distances of tours. That is, subtours that have a possibility of being better in TSP tours tend to have strong pheromones, and the ant agents specify the good regions in the search space by using this feedback mechanism. Several variations of the ant algorithm, such as the ANT-Q algorithm by Gambardella, the MAX-MIN ant system by Stützle and the rank-based version of the ant system have been proposed [2], [10], [18]. Moreover, the ant algorithm has

Manuscript received April 9, 1999.
Manuscript revised September 20, 1999.
†The authors are with the Graduate School of Engineering, Hokkaido University, Sapporo-shi, 060–8628 Japan.

been applied not only to TSPs but also to Quadratic Assignment Problems by Maniezzo, Graph Coloring Problems by Costa and Job Shop Problems by Colorni [6], [7], [13].

We have extended the original ant algorithm to a multiple ant colonies algorithm in order to improve the basic performance of the algorithm. This algorithm consists of some independent colonies that basically correspond to the original ant algorithm colony, but colony-level interactions have been introduced. Similar ideas were introduced in the parallel Genetic Algorithm, in which interactions among subpopulations are generally practiced by the exchanging operation of some individuals [15]. In our proposed algorithm, interactions among colonies are more naturally introduced by exchanging information on the pheromones as the schemata for solved problems. The pheromones belonging to one colony have different meanings for other colonies, i.e., there are *positive pheromone effects* and *negative pheromone effects*. The positive pheromone effects force ant agents to choose the way laid on, and the negative pheromone effects make the ant agents avoid to choose the way. This mechanism enables ant agents to provide good schemata to agents in other colonies and to share search regions with each other. Moreover, this mechanism seems to be appropriate for parallel computing in the sense of allocating one processor to one colony.

In Sect. 2 of this paper, we introduce the multiple ant colonies algorithm for TSPs. In Sect. 3, we present the results of computer experiments on some TSPs. Our proposals are discussed in Sect. 4, and conclusions are presented in Sect. 5.

2. Multiple Ant Colonies Algorithm

In this section, the extension of the original ant algorithm to a multiple ant colonies algorithm is described. This new algorithm has several independent colonies that basically correspond to the original ant colonies except that the behavior of ants in one colony is influenced by the pheromones in other colonies through colony-level interactions. There are two types of pheromone effect: a positive effect and a negative one. The positive pheromone effect enables ant agents to determine good schemata in their colony and provide such schemata to other colonies, whereas the negative pheromone effect prevents the ant agents from using the search regions indicated by other colonies' schemata and from sharing the regions with each other. These effects can be managed with the control parameters in the algorithm. These colony-level interactions may enable implementation of the ant algorithm to parallel computing and may enable improvement of the basic performance of the ant algorithm. Figure 1 shows an image of the multiple ant colonies algorithm with four colonies.

Here, the brief definition of a TSP is described [17].

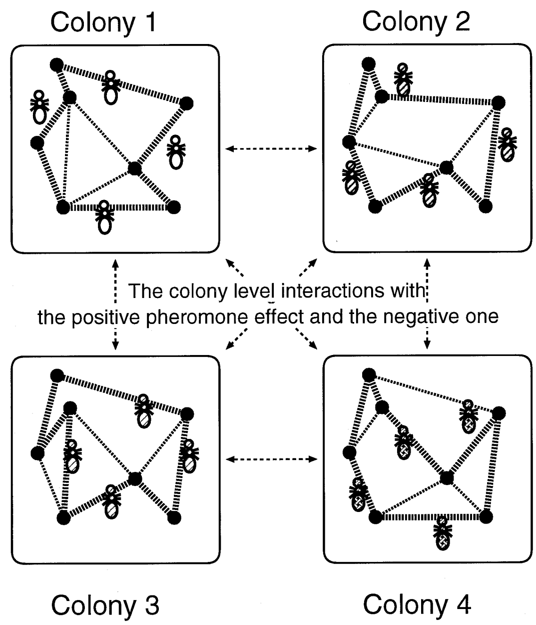


Fig. 1 An image of the multiple ant colonies algorithm for a TSP with four colonies.

Given a set of n cities, and distances d_{ij} between two cities, the TSP is a problem of finding the minimal length of a closed tour that passes through each city just once. An example of a TSP is given by a graph (N, E), where $N(|N| = n)$ is the set of cities and E is the set of edges between cities (a fully connected graph in Euclidean TSP). TSPs become symmetric in the case of $d_{ij} = d_{ji}$ and asymmetric in the case of $d_{ij} \neq d_{ji}$.

Next, we describe the multiple ant colonies algorithm. Let M and m be the number of colonies and the number of ant agents in each colony, respectively. The k-th ant agent in the h-th colony is denoted as the "(h, k) ant." The number of $M \cdot m$ ants search for TSPs at each time t. In the case of $M = 1$, this algorithm is the same as the original ant algorithm without colony-level interactions. Each ant agent is a simple agent with the following characteristics:

- It continuously moves from one city i to the next city j in the time interval between t and $t + 1$; the next city j is chosen according to probability that is a function of the distance d_{ij} and intensity of the pheromone present on the connecting edge in each colony;
- It visits only unvisited city by itself in preceding times; this property, implemented to force the ant agents to make legal tours, holds until its tour is completed; then the ant memory is cleared to begin its tour again;
- After completing its tour (i.e., after visiting all cities), it lays pheromones of various intensities along each edges included in its tour; the intensity of the pheromone laid is decided as a function of the total distance of its tour.

- It behaves continuously until stop time t_{MAX}.

To extend a single colony pheromone to multiple-colony pheromones, let $\tau_{ij}^h(t)$ be the intensity of the pheromone on the edge (i,j) in the h-th colony at time t. The intensity of the pheromone on the edges in each colony is initialized to a small positive value at time 0. After each ant has completed its tour in n time intervals, the intensity of the pheromone $\tau_{ij}^h(t)$ becomes

$$\tau_{ij}^h(t+n) = \rho \cdot \tau_{ij}^h(t) + \Delta\tau_{ij}^h, \tag{1}$$

where ρ is a coefficient such that $(1-\rho)$ represents the evaporation rate of the pheromones between time t and $t+n$ as in the original ant algorithm. The value of ρ must be set to a value less than 1 to avoid unlimited accumulation of the pheromone. The modified intensity of the pheromone $\Delta\tau_{ij}^h$ is defined as

$$\Delta\tau_{ij}^h = \sum_{k=1}^{m} \Delta\tau_{ij}^{hk}, \tag{2}$$

where $\Delta\tau_{ij}^{hk}$ represents the intensity per unit of length of the edge (i,j) along which the pheromove is laid by ant (h,k) between time t and $t+1$, and this is given as

$$\Delta\tau_{ij}^{hk} = \begin{cases} Q/L^{hk} & \begin{array}{l}\text{if ant } (h,k) \text{ uses edge} \\ (i,j) \text{ on its tour between} \\ \text{time } t \text{ and } t+n\end{array} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here, Q is a constant and scarcely affects the behavior of the algorithm, as is the case in the original algorithm. L^{hk} is the total tour length of and (h,k); i.e., the ant completing a tour with a shorter length is able to lay a larger intensity of the pheromome along edges included in its tour. The pheromones laid on probably better subtours would take large value as a result of whole behavior.

The new transition probability from city i to city j for ant (h,k) including colony-level interactions is defined as

$$p_{ij}^{hk}(t) = \begin{cases} \pi_{ij}^h(t) \Big/ \sum\limits_{l \notin \boldsymbol{tabu}^{hk}(t)} \pi_{il}^h(t) & \text{if } j \notin \boldsymbol{tabu}^{hk}(t) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\pi_{ij}^h(t) = \left\{ \prod_{l=1}^{M} \left[\tau_{ij}^l(t) + C(h)\right]^{\alpha(h,l)} \right\} \cdot [\eta_{ij}]^{\beta(h)} \tag{5}$$

$$\eta_{ij} = 1/d_{ij} \tag{6}$$

Here, $\boldsymbol{tabu}^{hk}(t)$ indicates the tabu list of ant (h,k). This list consists of cities that have already been visited cities until time t, and the ant is forbidden to choose such cities repeatedly. This is reset to ϕ when the ant visits all cities and completes its tour, so the ant agents can usually keep legal tours. $\pi_{ij}^h(t)$ means the degree of preference for an edge connected to city j. If this value is large, ant (h,k) tends to choose city j as the next one to visit. The parameter $\alpha(h,l)$, which is must be set in advance, determines the type of pheromone effect and the degree of influence from colony l. If $\alpha(h,l)$ is set to a positive value, i.e., a positive pheromone effect, ant (h,k) tends to prefer schemata indicated by the pheromone in the l-th colony. Conversely, if $\alpha(h,l)$ is negative, i.e., a negative pheromone effect, ant (h,k) tends to avoid choosing edges that have been frequently used in the l-th colony. If the value of $\alpha(h,l)$ is zero, the pheromone in the l-th colony has no effect on the behavior of ants in the h-th colony. The absolute value of $\alpha(h,l)$ indicates the degree of the pheromone effect. The effect becomes stronger as the value increases and becomes weaker as the value decreases. In particular, $\alpha(h,h)$, the effect from itself, must be positive in order to cause a positive feedback in the colony. $C(h)$ represents the degree of insensitivity to the pheromones, and this parameter is introduced so that the ant agents will avoid overly sensitive effects towards the pheromones near zero. η_{ij} is used as a greedy heuristic term that makes the ants prefer nearby cities, and it seems to be effective for TSPs. The parameter $\beta(h)$ represents how important η_{ij} is for the ants in the h-th colony. Figure 2 shows the structure of the ant agents with the control parameters, and Fig. 3 shows the procedure for the proposed algorithm.

The computational complexities from step 1 to step 5 are $O(Mn^2 + M^2n^2)$, $O(Mmn)$, $O(Mmn^2)$, $O(Mn + Mmn^2)$ and $O(Mn^2 + M^2n^2)$, respectively. Here, we suppose $M \ll m$ as the setting of the algorithm, and the computational complexity of the proposed algorithm is of $O(MTmn^2)$ where T is the number of cycles in the algorithm. The order of original ant algorithm is $O(Tmn^2)$ given by the case of $M = 1$ in the above. Therefore, the order of multiple ant colonies

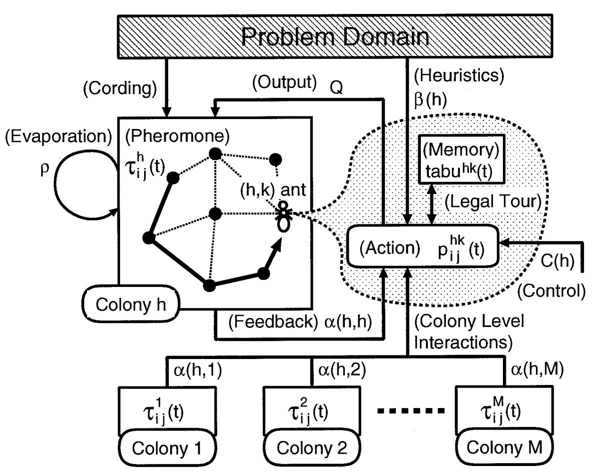


Fig. 2 Relationship between ant agents, colonies and problem domain with control parameters.

```
1. Initialize
   Set t := 0
   For h := 1 to M do
      For every edge (i, j)
         Set τ^h_ij(t) := initialvalue
2. Set all ants at the starting city
   Set s := 0
   Empty all tabu lists
   For h := 1 to M do
      For k := 1 to m do
         Place the starting city of the (h, k) ant,
and insert it in tabu^hk(t)
3. Repeat until all cities have been visited
   For s := 1 to n − 1 do
      For h := 1 to M do
         For k := 1 to m do
            Choose the city j to move to,
with probability p^hk_ij(t + s) given by equation (4)
            Move the (h, k) ant to the city j
            Insert the city j in tabu^hk(t + s)
4. Compute changes of the pheromones in every colony
   For h := 1 to M do
      For k := 1 to m do
         Move the the (h, k) ant to the starting city
         Compute tour length L^hk by the (h, k) ant
         Update the shortest tour found
   For h := 1 to M do
      For every edge (i, j)
         Δτ^h_ij := 0
         For k := 1 to m do
            Compute Δτ^hk_ij according to equation (3)
            Δτ^h_ij := Δτ^h_ij + Δτ^hk_ij
5. Update the pheromones in every colony
   For h := 1 to M do
      For every edge (i, j)
         τ^h_ij(t + n) := ρ · τ^h_ij(t) + Δτ^h_ij
   Set t := t + n
6. Check stop time
   If (t < t_MAX)
      then
         Goto step 2
      else
         Print shortest tour
         Stop
```

Fig. 3 The algorithm of the multiple ant colonies.

algorithm corresponds to the order of the ant algorithm in M times running. Namely, if it is showed that the multiple ant colonies algorithm has better performance than the ant algorithm in M times running, the proposed algorithm is superior to the original ant algorithm.

In this algorithm, more parameters than those in the original algorithm must be determined in advance, but trials of various relationships among colonies can be done by controlling these parameters. The management of colony structure is very important in the case of multiple colonies. In addition, we don't focus on exquisite agent strategies or heuristics specialized to the solved problem but a very simple heuristic in this paper. The objective of this study was how to extend the original ant algorithm to one for multiple ant colonies in order to improve the basic performance of the algorithm, and such specializations will be discussed in a future paper.

3. Computer Experiments

We performed some computer experiments to investigate the basic performance and the characteristics of the proposed algorithm compared with the original algorithm. The first experiment was carried out on a TSP with a form like a doughnut, and interesting results were obtained. In the second experiment, the pheromone effects based on colony-level interactions with eil101 benchmark TSP in TSPLIB were investigated. In the third experiment, the performance of the proposed algorithm was compared with that of the original one for some TSPs in TSPLIB. The values of parameters in each experiment were determined by reference to the original settings.

3.1 Experiment 1

The first experimental problem is a symmetric TSP in which 40 cities are located in a doughnut-like shape. The radii of the inside and outside circles are 15 and 20, respectively. This type of TSP is known to have several local optimal solutions, such as "C" type tours and "O" type tours. The total distance of a "C" type tour is a little longer than that of an "O" type tour in our experimental setting. That is, the "C" type tour has local optimal solutions and the "O" type tour has a global optimal.

In preliminary experiments, the original ant algorithm always found "C" type tours with all combinations of $\alpha = \{1, 2\}$ and $\beta = \{1, 2, 5, 10\}$ in many trials. This is because incomplete "O" type tours tend to have larger total distances than do incomplete "C" type tours and because the intensities of the pheromone along the edges of the "C" type tour would be greater. This indicates that the existence of unsearched regions would be a characteristic of the original ant algorithm with a simple heuristic. Thus, care should be taken in using heuristics in families of ant algorithms with conscious of problem features.

We applied the proposed algorithm to this TSP using the parameters listed in Table 1. With these settings, only negative pheromone effects were used as colony-level interactions. The first colony was not affected by other colonies, and the ants in the first colony could search at will. The colony was affected only by the first colony, and the third colony was affected by the first and second colonies (which can be called a *hierarchical colony structure*). $C(3)$ took a larger value than $C(1)$ and $C(2)$ because the third colony would be strongly influenced by two colonies.

Interesting results are shown in Fig. 4. Figures 4 (a)–(c) represent the transition process of the pheromones at each time. The left, middle and right figures are related to the intensities of pheromones in the first, second and third colonies respectively. The width of each line represents the intensity of the pheromone on each edge. In (a), ant agents in the first colony put effective pheromone on "C" type tours rapidly, then ant agents in the second colony searched different regions from those searched in the first due to the negative pheromone effect. Finally in (c), ant agents in the third colony succeeded to specify different regions from those in the first and second colonies. Figure 4 (d) shows the best solutions obtained in each colony through this experiment. The first colony found a solution with a distance of 218.1, and a global optimal solution with a distance of 209.5 was found by the second and third colonies. Namely, global optimal solutions that could not be found by the first colony were found by the second and third colonies, and three local and global optimal solutions could be found at same time. These results are very interesting, although we admit that this is an adhock case.

Table 1 Values of parameters used in experiment 1.

M	m	ρ	Q	$\tau_{ij}^h(0)$	t_{MAX}
3	40	0.65	200	5	240

	1	2	3
$\alpha(1,-)$	2	0	0
$\alpha(2,-)$	−0.5	2	0
$\alpha(3,-)$	−0.2	−0.2	2

$\beta(1)$	5
$\beta(2)$	5
$\beta(3)$	5

$C(1)$	1
$C(2)$	2
$C(3)$	20

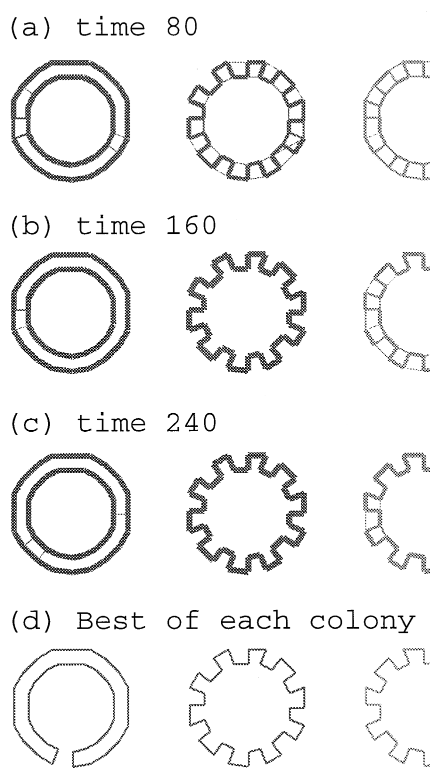


Fig. 4 The transition of the pheromone and the best solutions obtained in experiment 1.

3.2 Experiment 2

Experiment 2 was performed to confirm the general properties of colony-level interactions with each setting of parameter α for eil101, 101 cities symmetric benchmark in TSPLIB. For this confirmation, we prepared ten colonies with one-way interactions, in which the first colony was not affected by other colonies, and other colonies were affected by only the first colony with various values of α. The parameter values used in this experiment are listed in Table 2. In addition, the values of $\alpha(h,l)$, which are not listed in the table, were set to zero. The second, third, fourth and fifth colonies would receive negative pheromone effects from the first, while the sixth colony was neutral, and the seventh, eighth, ninth and tenth colonies would receive positive pheromone effects from the first.

Table 3 shows the results averaged over 40 trials with the average distance of the best solution, the quality of a known global optimal solution (642.3), the standard deviation, and the ratio of duplicate subtours compared with the best solution of the first colony. The ratio of duplicate subtours between the first and the sixth colonies, i.e., a pair with no interaction, was 76.6%, which indicates that the original ant algorithm has a tendency to always generate solutions with about 76.6% of the same subtours in each trial. The ratios between the first and second colnies and between the first and third colonies were very low, but the second and third colonies found worse solutions in strongly avoiding subtours well used by the first colony. The fifth colony which received a slightly negative pheromone effect, generated solutions of almost the same quality as those of the first colony. The ratio between the first and fifth colonies was lower than between the first and sixth colonies, and different regions were searched by the fifth colony. The seventh, eighth, ninth and tenth colonies had solutions of about the same quality with higher ratios; that is, these colonies searched similar regions due to the positive pheromone effects from the first colony.

We showed that it is possible to control the searching behavior of each colony with colony-level interactions based on parameter α. Strong negative

Table 2 Values of parameters used in experiment 2.

M	m	ρ	Q	$\tau_{ij}^h(0)$	t_{MAX}	$\beta(h)$	$C(h)$
10	101	0.65	200	5	303000	1.5	1

$\alpha(2,1)$	$\alpha(3,1)$	$\alpha(4,1)$	$\alpha(5,1)$	$\alpha(6,1)$
−1.0	−0.5	−0.2	−0.1	0

$\alpha(7,1)$	$\alpha(8,1)$	$\alpha(9,1)$	$\alpha(10,1)$	$\alpha(h,h)$
0.1	0.2	0.5	1.0	1.5

Table 3 Results of experiment 2.

Colony No	1	2	3	4	5	6	7	8	9	10
Average	691.9	896.0	881.5	724.3	693.9	685.5	685.9	689.1	693.1	693.7
Quality	107.7%	139.5%	137.2%	112.8%	108.0%	106.7%	106.8%	107.3%	107.9%	108.0%
Std. Dev.	13.1	20.6	17.9	15.5	20.0	15.7	12.5	13.6	12.0	10.1
Ratio Dup.	100%	1.3%	2.5%	40.3%	61.4%	76.6%	86.4%	88.8%	88.9%	88.4%

Table 4 Results of experiment 3 (1) using a network colony structure.

	The Multiple Ant Colonies Algorithm with Network Colony Structure							Ant Algorithm
Colony No	1	2	3	4	5	6	Best	Best
Average	692.2	683.6	695.5	685.7	689.7	**680.1**	**665.4**	676.6
Quality	107.8%	106.4%	108.3%	106.7%	107.4%	**105.9%**	**103.6%**	105.3%
Std. Dev.	21.5	18.9	15.3	19.4	15.7	11.0	9.1	8.0

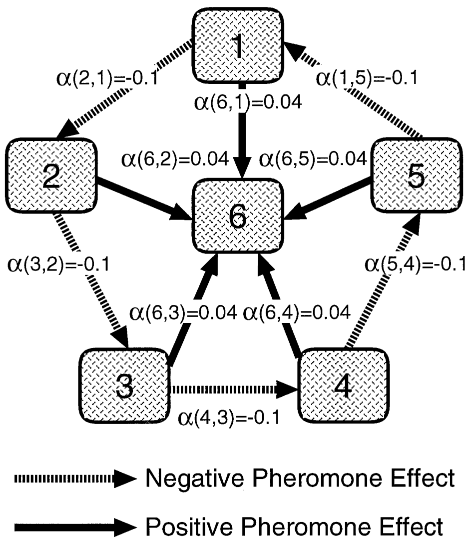


Fig. 5 A network colony structure with six colonies. $\alpha(h,h) = 1.5$. The values of $\alpha(h,l)$ not shown were all set to zero.

pheromone effects are not effective and lead to solutions of inferior quality, while a slight negative pheromone effect, such as setting $\alpha(h,l)$ to 0.1, seems to be useful for various searches while maintaining the quality of the solution.

3.3 Experiment 3

In experiment 2, the negative pheromone effects were found to be useful for various searches while maintaining the quality of the solution, so we constructed *the network colony structure* shown in Fig. 5. In this structure, five colonies placed around colony 6 were related to neighboring colonies with slightly negative pheromone effects, and the central sixth colony received positive pheromone effects from all the surrounding colonies. We expected that in this structure the surrounding colonies could each search various regions and that the central colony would recieve good schemata from the surrounding colonies due to positive pheromone effects. The values of parameter α are shown in Fig. 5, and m was set to 101, the number of cities. The values of other parameters were the same as those in experiment 2.

The results for eil101 averaged over 20 trials are shown in Table 4. The results obtained from the original ant algorithm are shown in the last column for comparison with those obtained from the multiple ant colonies algorithm, and the results of the original ant algorithm show the best solution obtained from 6 independent run in each trial. That is, it gave the same number of solutions as those of the multiple colonies algorithm in each trial. The columns labeled "Best" show the best solutions obtained from 6 colonies in each trial. In the multiple ant colonies algorithm, the sixth colony discovered the best solutions 6 times and showed the best performance of all colonies, as was expected. The sixth colony succeeded to gather good schemata. Moreover, the best solutions obtained by the multiple ant colonies algorithm were of better quality thatn those obtained by the original ant algorithm. Examples of the best solutions obtained by each colony are shown in Fig. 6. It is clear from those examples that the colony-level interactions and the network colony structure are very effective.

Next, two experiments 3(2) and 3(3) were performed in order to confirm the effectiveness of negative pheromone effects in the network colony structure. In experiment 3(2), the values of parameters were the same as those in Fig. 5 except having the positive pheromone effects instead of the negative effects, i.e., $\alpha(2,1) = \alpha(3,2) = \alpha(4,3) = \alpha(5,4) = \alpha(1,5) = 0.1$. The settings of experiment 3(3) were also the same as those in Fig. 5 except having no effects, i.e., $\alpha(2,1) = \alpha(3,2) = \alpha(4,3) = \alpha(5,4) = \alpha(1,5) = 0$. The results of both experiments are shown in Tables 5 and 6. In both cases, the surrounding colonies (No. 1–5) could generate solutions with similar quality to the experiment 2, but the center colony (No. 6) could generate worse qualities. Moreover, the best solutions of both cases had similar quality to the original ant algorithm. These results indicate that the network colony structure without the negative pheromone effects tend to search wasteful regions without varieties. Namely, the surrounding colonies behaved similarly, and the cen-

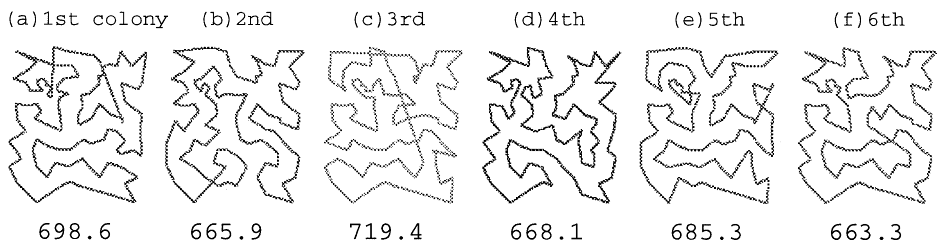


Fig. 6 Examples of best solutions obtained by each colony in experiment 3 (1).

Table 5 Results of experiment 3 (2) using a network colony structure based on the positive pheromone effects.

Colony No	1	2	3	4	5	6	Best
Average	703.6	688.6	686.9	685.9	696.1	692.9	677.5
Quality	109.5%	107.2%	106.9%	106.8%	108.3%	107.8%	105.5%
Std. Dev.	21.8	16.6	12.8	14.2	13.1	17.8	11.4

Table 6 Results of experiment 3 (3) using a network colony structure based on no effects.

Colony No	1	2	3	4	5	6	Best
Average	698.5	684.2	689.0	686.8	695.9	691.8	673.4
Quality	108.7%	106.5%	107.3%	106.9%	108.3%	107.7%	104.8%
Std. Dev.	17.5	10.8	13.6	10.7	11.8	21.1	9.1

Table 7 Results for other TSPs.

Problem		Multiple Ant Colonies Algorithm				Ant Algorithm			
Name	Dimension	Best Avg.	Quality	Std. Dev.	Time Avg.	Best Avg.	Quality	Std. Dev	Time Avg.
att48	48	**34972.1**	**104.3**%	222.2	139.6 sec	35169.2	104.9%	232.5	56.1 sec
eil51	51	**448.3**	**104.3**%	4.8	196.6 sec	457.7	106.5%	2.6	146.7 sec
st70	70	**710.9**	**104.8**%	8.3	319.2 sec	727.9	107.3%	15.9	341.1 sec
gr96	96	**543.9**	**106.2**%	2.9	956.5 sec	550.3	107.4%	14.1	848.8 sec
ch130	130	**6348.6**	**103.9**%	43.8	1398.4 sec	6423.1	105.1%	81.6	1757.2 sec

ter colony couldn't gather effective schemata concerned with the search space. Therefore, we can say that the network colony structure with the negative pheromone effects are very effective against the original ant algorithm.

Table 7 shows the results averaged over 20 trials for other TSPs in TSPLIB with the same colony structure used in experiment 3. In all cases, the multiple ant colonies algorithm showed better performance than did the original ant algorithm. In addition, "Time Avg." columns in Table 7 show the time to find best solutions by Pentium III 500 MHz IBM-PC computer based on Linux Operating System. This result indicates the use of the multiple ant colonies algorithm is effective with better qualities against the original one because the time of both algorithms are almost the same. The time to find the best solutions increases linearly as the size of the problems increases. and it shows the possibility to apply the proposed algorithm to large TSPs.

However, the qualities of solution obtained by both algorithms do not seem to be better than the quality of solutions obtained by approximate methods, such as Genetic Algorithm, Simulated Annealing, and Tabu Search, due to a lack of careful adfustment to the parameters for solving these TSPs. In particular, the ant agents in both algorithms generated tours that included many cross-subtours in each trial. For example, this weakness in TSPs could be broken to regard solutions that are generated by the ant agents as genotype solutions and then these genotypes could be translated into phenotype solutions in such a way as to apply operations such as 2-opt to each cross-subtour point; however, this idea was not used in this study for only comparisons between the proposed algorithm and the original one.

4. Discussion

Generally, approximate methods with hand-crafted heuristics tend to have weak points that search regions would be specified by properties of used heuristics but these heuristics often make search efficient. The original ant algorithm also has this weakness, and we should therefore handle heuristics with conscious of problem features. We consider that the proposed multiple ant colonies algorithm can overcome this weakness by the introduction of colony-level interactions without requiring exquisite heuristics. The negative pheromone effects enable the maintenance in the search process, and colonies can easily exchange good schemata with each other due to the positive pheromone effects. In our proposals, we didn't dare to include well tuned up ant agent strategies for the sake to extend a basic performance of the ant algorithm, and the performance will be still better with nice strategies using in the ant algorithm families.

Moreover, the proposed algorithm can be applied to a parallel computing, such as cluster computing, without the need to consider synchronization of operations, and it would have better performance than that of the original ant algorithm simply running parallel. However, further investigation of the construction of colony structures and determination of interactions between colonies is needed. It is desirable for the algorithm to autonomously, adaptively, and dynamically organize its interactions in accordance with states, although we decided as hand-crafted in the experiments, and the interactions worked effectively in a network colony structure. One idea for the control of interactions is to introduce a super ant agent, i.e., the queen ant agent, that observes the behavior of the ant agents and manages the colony-level interactions. The queen ant agent could be designed to focus on uni-path behavior, i.e., the situation in which all ant agents make the same tour [5]. It will be effective that the queen ant agent operates the changing mechanisms of the parameters according to the whole behaviors in its colony.

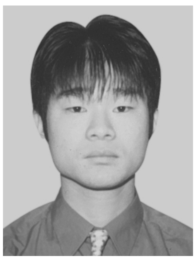
5. Conclusions

We have extended the original ant algorithm to a multiple ant colonies algorithm with introducing the colony-level interactions such as positive and the negative pheromone effects. The negative pheromone effects enable maintenance of variation in the search process while maintaining quality in the solutions, and colonies are able to exchange good schemata for solving a problem due to the positive pheromone effects. Through computer experiments, we showed that the proposed algorithm has better performance than the original ant algorithm with almost the same agent strategies. In a future study, we will attempt to apply the proposed algorithm to cluster computing with well-defined heuristics or strategies and also to huge combinatorial optimization problems such as a TSP problem involving several thousand cities.

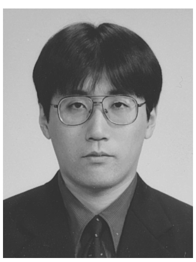
References

[1] W.C. Agosta, Chemical Communication – The Language of Pheromone, W.H. Freeman and Company, New York, 1992.
[2] B. Bullnheimer, R.F. Hartl, and C. Strauss, A New Rank Based Version of the Ant System: A Computational Study, Working paper, Institute of Management Science, University of Vienna, Australia, 1997.
[3] G.D. Caro and M. Dorigo, "AntNet: A mobile agents approach to adaptive routing," Technical Report 97/12 of IRIDIA, Université Libre de Bruxelles, 1997.
[4] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," Proc. ECAL91-European Conf. on Artificial Life, Paris, France, eds. F. Varela and P. Bourgine, pp.134–142, Elsevier Publishing, 1991.
[5] A. Colorni, M. Dorigo, and V. Maniezzo, "An investigation of some properties of an ant algorithm," Proc. the Parallel Problem Solving from Nature Conf., Brussels, Belgium, eds. R.Manner and B. Manderick, pp.509–520, Elsevier Publishing, 1992.
[6] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian, "Ant system for job-shop scheduling," Belgian Journal of Operations Research, Statistics and Computer Science, vol.34, pp.39–53, 1994.
[7] D. Costa and D. Snyers, "Ants can colour graphs," Journal of the Operational Research Society, vol.48, pp.295–305, 1997.
[8] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant sysytem: Optimization by a colony of cooperating agents," IEEE Trans. Syst., Man. & Cybern.-Part B, vol.26, no.1, pp.1–13, 1996.
[9] M. Dorigo and L.M. Gambardella, Ant Colonies for the Travelling Salesman Problem, BioSystems 43, pp.73–81, Elsevier, 1997.
[10] L.M. Gambardella and M. Dorigo, "Ant-Q: A reinforcement learning approach to the traveling salesman problem," Proc. ML-95, Twelfth International Conf. on Machine Learning, pp.252–260, 1995.
[11] H. Kawamura, M. Yamamoto, T. Mitamura, K. Suzuki, and A. Ohuchi, "Cooperative search based on pheromone communication for vehicle routing problems," IEICE Trans. Fundamentals, vol.E81-A, no.6, pp.1089–1096, June 1998.
[12] P. Kuntz, P. Layzell, and D. Snyers, "A colony of ant-like agents for partitioning in VLSI technology," Proc. Forth European Conf. on Artificial Life, eds. P. Husbands and I. Howey, pp.417–424, 1997.
[13] V. Maniezzo, A. Colorni, and M. Dorigo, "The ant system applied to the quadratic assignment problem," Technical Report 94/28 of IRIDIA, Université Libre de Bruxelles, 1994.
[14] T. Mikami, S. Mikami, and M. Wada, "Ant-like collective sorting for massive data analysis – A quantitative study," Proc. Intelligent Autonomous Systems (IAS-5), eds. Y. Kakazu, M. Wada, and T. Sato, pp.686–693, 1998.
[15] H. Mühlenbein, M. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," Proc. Fourth International Conf. on Genetic Algorithm, eds. R.K. Belew and L.B. Booker, pp.271–278, 1991.
[16] M. Nakamura and K. Kurumatani, "Formation mehcanism of pheromone pattern and control of foraging behavior in an ant colony model," Proc. Fifth International Workshop on the Synthesis and Simulation of Living Systems, eds. C.G. Langton and K. Shimohara, pp.67–74, The MIT Press, 1997.
[17] G. Reinelt, "The traveling salesman – Computational solutions for TSP applications," in Lecture Notes in Computer Science 840, eds. G. Goos and J. Hartmanis, Springer-Verlag, 1994.
[18] T. Stützle and H. Hoos, "The max-min ant system and local search for the traveling salesman problem," Proc. IEEE International Conf. of Evolutionary Computation (ICEC '97), pp.308–313, 1997.
[19] D. Subramanian, P. Drushcel, and J. Chen, "Ants and reinforcement learning: A case study in routing in dynamic networks," Proc. Fifteenth International Joint Conf. on Artificial Intelligence (IJCAI-97), pp.832–838, 1997.

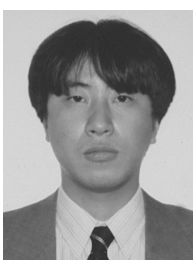
[20] K. Suzuki and A. Ohuchi, "Reorganizations of agents with pheromone style communication in multiple monkey banana problem," Proc. Intelligent Autonomous Systems (IAS-5), eds. Y. Kakazu, M. Wada, and T. Sato, pp.615–622, 1998.
[21] M. Yamamoto, H. Kawamura, K. Suzuki, and A. Ohuchi, "Collective approarch for combinatorial optimization problems," Proc. The International Technical Conf. on Circuits/Systems, Computers and Communications (ITC-CSCC98), pp.1479–1482, 1998.



Hidenori Kawamura received the M.C. degree from Division of System and Information Engineering, Graduate School of Engineering, Hokkaido University in 1998. He is currently D2 in Division of System and Information Engineering, Graduate School of Engineering, Hokkaido University. His research interests include Multi-Agent systems and combinatorial optimization problems. He is a member of IPSJ and JSAI.



Masahito Yamamoto received the Ph.D. degree from Institute of Information Engineering, Graduate School of Engineering, Hokkaido University in 1996. He is currently an Asistant in the Graduate School of Engineering, Hokkaido University. His research interests include automated reasoning and optimization and search. He is a member of IPSJ, SICE, JSAI and ORSJ.



Keiji Suzuki received the Ph.D. degree from Institute of Precision Engineering, Graduate School of Engineering, Hokkaido University in 1993. He is currently an Associate Professor in the Graduate School of Engineering, Hokkaido University. His research interests include artificial life, evolutionary computation, and multi-agent systems. He is a member of IPSJ, JSAI, and JSME.

Azuma Ohuchi received his Ph.D. degree in 1974 from Hokkaido University. He is currently a Professor in the Graduate School of Engineering, Hokkaido University. His research interestes include systems information engineering, applied artificial intelligence, and medical systems. He is a member of the IPSJ, JSAI, IEEJ, ORSJ, Soc. Contr. Eng., Jap. OR Soc., Soc. Med. Informatics, Hosp. Manag., and IEEE-SMC.