



Growing Ant Colony Optimization with Intelligent and Dull Ants and its Application to TSP

Sho Shimomura
Dept. Electrical and Electronic Eng.,
Tokushima University
Email: s-sho@ee.tokushima-u.ac.jp

Haruna Matsushita
Dept. Reliability-based Information Systems Eng.,
Kagawa University
Email: haruna@eng.kagawa-u.ac.jp

Yoko Uwate
Dept. Electrical and Electronic Eng.,
Tokushima University
Email: uwate@ee.tokushima-u.ac.jp

Yoshifumi Nishio
Dept. Electrical and Electronic Eng.,
Tokushima University
Email: nishio@ee.tokushima-u.ac.jp

Abstract—This study proposes an Growing Ant Colony Optimization with Intelligent and Dull Ants (GIDACO). In the GIDACO algorithm, two kinds of ants coexist: *intelligent ants* and *dull ants*. In addition, the number of ants in the colony is increased with the passage of time. The new ant is added based on the best ant which has the best tour. Also, the added ant is decided whether the dull ant or intelligent ant by the probability. By growing the colony, the structure of GIDACO is changed flexibly and automatically depending on the problem. Moreover, the growing can also reduce the simulation time because the simulation time depends on the number of ants. We apply GIDACO to Traveling Salesman Problems (TSPs) and confirm that GIDACO obtains more effective results than the standard ACO which consists of only the intelligent ants.

I. INTRODUCTION

Ant Colony Optimization (ACO) [1][2] is proposed to solve difficult combinatorial optimization problems, such as Traveling Salesman Problem (TSP), graph coloring problem, routing in communications networks, Quadratic Assignment Problem (QAP) [3]-[7] and so on. TSP is a problem in combinatorial optimization studied in an operations research and a theoretical computer science. In TSP, given a list of cities and their pairwise distances, the task is to find the shortest possible tour that each city exactly visited once. In the ACO algorithm, multiple solutions called “ants” coexist, and the ants drop a substance called the *pheromone*. Pheromone trails are updated depending on the behavior of the ants. By communicating with other ants according to the pheromone strength, the algorithm tries to find the optimal solution. However, ACO has a problem which is to trap at local solutions. Therefore, it is important to enhance the algorithm performances by improving its flexibility.

Meanwhile, it has been reported that about 20 percent of the ants are unnecessary ants called “dull ant” in the real ant’s world [8]. The dull ant keeps dawdling its colony whereas the other ants in the colony perform feeding behavior. In

a computational experiment, the researchers performed the feeding behavior by using intelligent ants, which can trail the pheromone exactly, and dull ants which cannot trail the pheromone. From these results, the ants group including the dull ants can obtain more foods than the group containing only the intelligent ants. The dull ants are regarded as having task which find the new food sources by dawdling. It means that the coexistence of the intelligent and dull ant improves the effectiveness of the feeding behavior.

In this study, we propose a new type of the ACO algorithm called Growing Ant Colony Optimization with Intelligent and Dull Ants (GIDACO). The first important feature is that two kinds of ants coexist. The one is an *intelligent ant* and the another is a *dull ant*. The intelligent ant can trail the pheromone and the dull ants cannot trail the pheromone. The second important feature of GIDACO is that the number of ants in the colony is increased with the passage of time. In other words, there are a few ants in the initial colony, however, the colony grows depending on ants’ conditions. The new ants are added based on the ant which obtained the best tour. Also, the added ant is decided whether dull ant or intelligent ant by the probability. By growing the colony, the structure of GIDACO is changed flexibly and automatically depending on the problem. Moreover, the growing can also reduce the simulation time because the simulation time depends on the number of ants. Because their features are essentially similar to the real ant’s world, we can say that GIDACO algorithm is nearer to the real ant colony than the standard ACO algorithm.

This paper is organized as follows. The basic method of ACO is introduced in the Section II. We explain the GIDACO algorithm in detail in the Section III. In the Section IV, we apply GIDACO to four kinds of TSPs whose number of cities are different. Furthermore, we investigate the behavior of GIDACO in detail. Performances are evaluated quantitatively in comparison with the standard ACO. We confirm that GIDACO

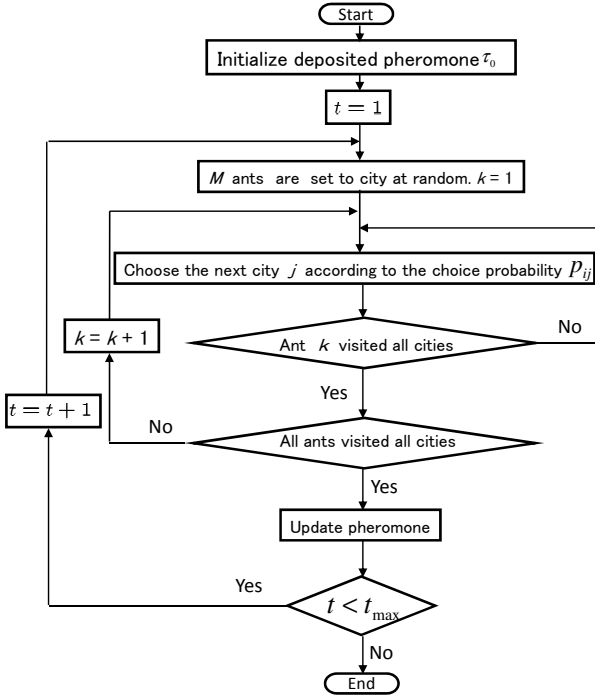


Fig. 1. Flowchart of the ACO algorithm.

can obtain a more effective results than the standard ACO. Finally, the Section V concludes this paper.

II. THE BASIC ALGORITHM OF ACO

ACO is popular search algorithms and applied to various applications. In this section, we introduce the basic algorithm of ACO.

A flowchart of the ACO algorithm is shown in Fig. 1. N -city of TSP is denoted as

$$S \equiv \{P_1, P_2, \dots, P_N\}, P_i \equiv (x_i, y_i), \quad (1)$$

where the data area is normalized from 0 to 1, and P_i denotes the i -th city position ($i = 1, 2, \dots, N$). Each ant k (total M) is deposited on a city selected at random.

[ACO1](Initialization): Let the iteration number $t = 0$. $\tau_{ij}(t)$ is the amount of pheromone deposited on the path (i, j) between the city i and j at time t , and $\tau_{ij}(t)$ is initially set to τ_0 .

[ACO2](Find tour): The visiting city is chosen by according to the probability $p_{ij}(t)$ as shown in Fig. 2. The probability of k -th ant moving from the city i to j is decided by

$$p_{kij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta}, \quad (2)$$

where $k = 1, 2, \dots, M$, and $1/\eta_{ij}$ is the distance of the path (i, j) . The adjustable parameters α and β control the weight of the pheromone intensity and of the city information, respectively. Therefore, the searching ability goes up and down by changing α and β . As Eq. (2), the ants judge next city by

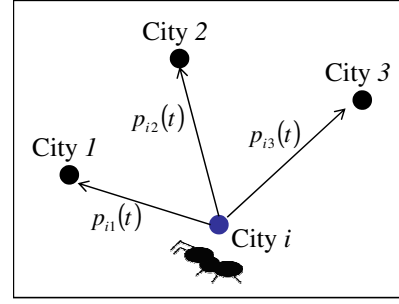


Fig. 2. Probability $p_{ij}(t)$ of ants. The visiting city is chosen by probability $p_{ij}(t)$.

both the pheromone and the distance from the present location. N_k is a set of the cities that k -th ant has never visited. The ants repeat choosing next city until all the cities are visited.

[ACO3](Pheromone update): After all ants have completed their tours, the amount of pheromone deposited on each path is updated. The tour length $L_k(t)$ is computed, and the amount of pheromone $\Delta\tau_{kij}(t)$ deposited on the path (i, j) by k -th ant is decided as

$$\Delta\tau_{kij}(t) = \begin{cases} 10/L_k, & \text{if } (i, j) \in T_k(t) \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where $T_k(t)$ is the tour obtained by k -th ant, and $L_k(t)$ is its length. Total amount of pheromone $\tau_{ij}(t)$ of each path (i, j) is updated depending on $\Delta\tau_{kij}(t)$;

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^M \Delta\tau_{kij}(t), \quad (4)$$

where $\rho \in [0, 1]$ is the rate of pheromone evaporation.

[ACO4] Let $t = t + 1$. Go back to [ACO2] and repeat until $t = t_{\max}$.

III. GROWING ANT COLONY OPTIMIZATION WITH INTELLIGENT AND DULL ANTS

We explain the proposed GIDACO algorithm in detail. A flowchart of the GIDACO algorithm is shown in Fig. 3. The first important feature of GIDACO is that two kinds of ants coexist; *intelligent ant* which can exactly trail the pheromone and *dull ant* which cannot trail the pheromone for making the tour. The second important feature of GIDACO is that the number of ants is increased with the passage of time. In other words, the colony grows depending on the problem. N -city positions of TSP is set to according to Eq. (1). First, each ant k (total M_G) is deposited on a city selected at random. Ants are classified into a set of the intelligent ants S_{Intel} , namely, the initial colony consists of only the intelligent ants.

[GIDACO1](Initialization): Let iteration number $t = 0$ and initialize the pheromone $\tau_{ij}(t)$ according to [ACO1].

[GIDACO2](Find tour): For the intelligent ants and the dull ants, the visiting city is chosen by the probability $p_{ij,t}(t)$ and

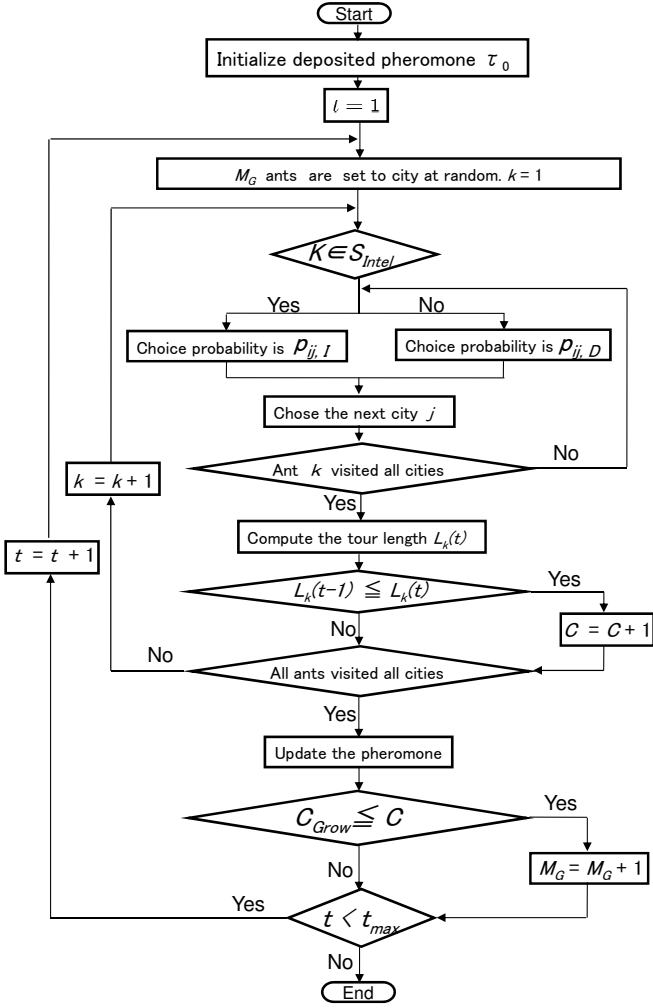


Fig. 3. Flowchart of the GIDACO algorithm.

$p_{ij,D}(t)$ as shown in Fig. 4. The probability of k -th ant moving from the city i to j is decided by

$$p_{k_{ij},D}(t) = \frac{[\eta_{ij}]^{\beta_D}}{\sum_{l \in N_k} [\eta_{il}]^{\beta_D}}, \quad \text{if } k \in S_{\text{dull}} \quad (5)$$

$$p_{k_{ij},I}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^{\beta_I}}{\sum_{l \in N_k} [\tau_{il}(t)]^\alpha [\eta_{il}]^{\beta_I}}. \quad \text{otherwise,} \quad (6)$$

The adjustable parameters β_I and β_D control the weight of the city information of the intelligent ant and of the dull ant, respectively. As Eq. (5) does not include the amount of deposited pheromone $\tau_{ij}(t)$ and $\tau_{il}(t)$, the dull ants cannot trail the pheromone. The dull ants judge next city depending on only the distance from the present location, in contrast to the intelligent ants which judge next city by the pheromone and the distance from the present location.

[GIDACO3](Pheromone update): After all ants have completed their tours, the amount of deposited pheromone on each path is updated. We should note that the dull ants can deposit the pheromone on the path, though, they cannot trail

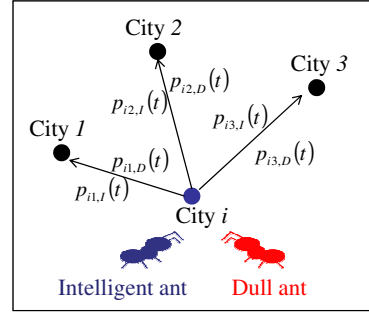


Fig. 4. Probability $p_{ij}(t)$ of Intelligent and Dull ants. The visiting city of intelligent ant is chosen by the probability $p_{ij,I}(t)$. The visiting city of dull ant is chosen by the probability $p_{ij,D}(t)$ which does not include the amount of pheromone.

the pheromone. Then, the tour length $L_k(t)$ is computed for both the intelligent and dull ants, and the amount of pheromone $\Delta\tau_{k_{ij}}(t)$ deposited by k -th ant on the path (i, j) is decided according to Eq. (3), Update $\tau_{ij}(t)$ of each path (i, j) according to Eq. (4)

[GIDACO4](Growing): After the their tour length $L_k(t)$ is computed, we evaluate the tour length $L_k(t)$ and consider whether the colony should be grown. A growing rule is as follows;

$$C = \begin{cases} C + 1, & \text{if } L_k(t-1) \leq L_k(t) \\ C, & \text{otherwise,} \end{cases} \quad (7)$$

where C is a counter which counts the number of times which obtained results are worse than that of last iteration. The number of ants is decided by

$$M_G = \begin{cases} M_G + 1 \text{ and } C = 0, & \text{if } C \geq C_{\text{Grow}} \\ M_G. & \text{otherwise,} \end{cases} \quad (8)$$

The adjustable parameter C_{Grow} controls the rate of ants increase. The added ant is decided whether dull ant or intelligent ant by the probability. The dull ant is added by according to the probability P_m , in contrast the intelligent ant is added by the probability $(1 - P_m)$. In addition, the added ant is affected by the best ant. Namely, the new ant is deposited in the same city of ant which obtained best tour.

[GIDACO5] Let $t = t + 1$. Go back to [GIDACO2] and repeat until $t = t_{\text{max}}$.

IV. NUMERICAL EXPERIMENTS OF TSP

In order to evaluate a performance of GIDACO and to investigate its behavior, we apply GIDACO to various TSPs. We compare GIDACO with the standard ACO.

In the experiments, the number of ants M in the standard ACO is set to the same as the number of cities, namely $M = N$. However, the initial number of ants M_G in the GIDACO is set to the one-tenth of the number of cities, namely $M_G = N/10$. Therefore, the number of ants M_G in GIDACO is changed from $N/10$ to N . The standard ACO contains ants whose choice probability is decided by Eq. (2). GIDACO includes dull ants and intelligent ants whose choice probability

TABLE I
RESULTS OF THE STANDARD ACO AND GIDACO.

	att48	eil51	kroC100	gr120
Average of ACO	2.87%	7.43%	10.59%	7.05%
Average of GIDACO	1.65%	3.74%	9.78%	5.22%
Improved rate of GIDACO from ACO	42.5%	49.66%	7.65%	25.96%

is decided by Eq. (6). We repeat the simulation 20 times for all the problems. The parameters of two methods are set as follows;

$$\tau_0 = 10, \rho = 0.3, \alpha = 1, \beta = \beta_I = \beta_D = 5,$$

$$P_m = 0.2, C_{Grow} = 30, t_{max} = 2000,$$

where the evaporation rate ρ , the weight of pheromone α , the weight of distance β , β_I and β_D , probability of dull ant P_m , the rate of ants increase C_{Grow} and the search limit $t = t_{max}$ are fixed values. In order to compare obtained solutions with the optimal solution, we use an error rate as follow;

$$\begin{aligned} & \text{Error rate}[\%] \\ &= \frac{(\text{obtained solution}) - (\text{optimal solution})}{(\text{optimal solution})} \times 100. \end{aligned} \quad (9)$$

This equation shows how close to the optimal solution the ACOs obtain the tour length. Thus, the error rate nearer 0 is more desirable. Furthermore, in order to evaluate how well the solution of GIDACO are improved from that of ACO, we use an improved rate as follow;

$$\begin{aligned} & \text{Improved rate}[\%] = \\ & \frac{(\text{Avg. of Error of ACO}) - (\text{Avg. of Error of GIDACO})}{(\text{Avg. of Error of ACO})} \times 100. \end{aligned} \quad (10)$$

The TSPs are conducted on *att48* (composed of 48 cities), *eil51* (composed of 51 cities), *kroC100* (composed of 100 cities) and *gr120* (composed of 120 cities). These problems are quoted from TSPLIB [9]. The simulation results of the standard ACO and GIDACO are shown in Table I.

In this table, the average values nearer 0 are more desirable. We can confirm that the proposed GIDACO obtained the best results than the standard ACO for all the problems. Furthermore, we investigate the relationship between the performance and the number of ants of GIDACO for *eil51*. Figure 5 shows the simulated result for 1 trial. We can see that obtained solutions are improved by increasing the number of ants. Also, GIDACO can reduce the simulation time because the simulation time depends on the number of ants. From these results, we can say that GIDACO is more effective algorithm than the standard ACO in solving TSPs.

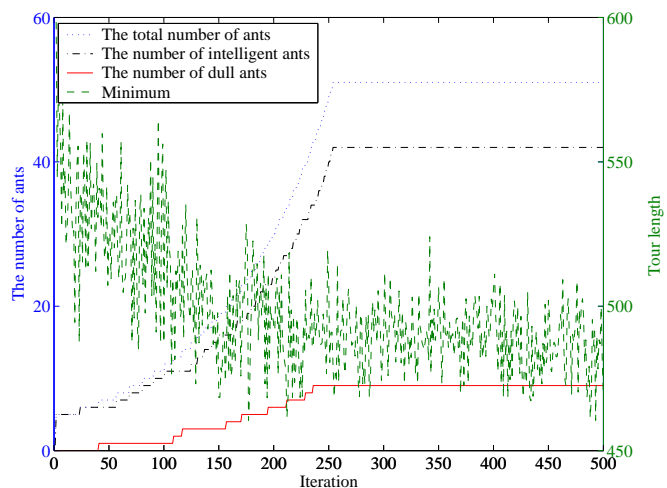


Fig. 5. Relationship between the performance and the number of ants in the simulation of *eil51*.

V. CONCLUSION

In this study, we have proposed Growing Ant Colony Optimization with Intelligent and Dull Ants (GIDACO). We have investigated the performances of GIDACO by applying it to four TSPs. We have confirmed that GIDACO including the dull ants obtained better results than the standard ACO which containing only the intelligent ants. We considered that the structure of GIDACO which changed flexibly and automatically depending on the problem is effective. From these results, we can say that GIDACO is more effective algorithm than the standard ACO in solving TSPs.

REFERENCES

- [1] M. Dorigo and T. Stutzle, *Ant Colony Optimization*, Bradford Books, 2004.
- [2] M. Dorigo, V. Maniezzo and A. Colomi, "Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetic*, 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] M. Dorigo, G. D. Caro and L. M. Gambardella, "Ant Algorithms for Discrete Optimization," *Artificial Life* vol. 5 no. 2, pp. 137-172, 1999.
- [5] Blum and Merkel (eds.), *Swarm Intelligence*, Springer, 2008.
- [6] Inspiration for optimization from social insect behavior. E. Bonabeau, M. Dorigo and G. Theraulaz.
- [7] V. Maniezzo and A.Colomi, "The ant system applied to the quadratic assignment problem," *IEEE Transactions on Knowledge and Data Engineering* vol. 11 no. 5, pp. 769-778, 1999.
- [8] H. Hasegawa, "Optimization of GROUP Behavior," *Japan Ethological Society Newsletter*, no. 43, pp. 22-23, 2004 (in Japanese).
- [9] G. Reinelt, "TSPLIB-A traveling salesman problem library," *ORSA J. Comput.* Vol. 3: pp.376-384, 1991.