2017 IEEE 10th International Workshop on Computational Intelligence and Applications
November 11-12, 2017, Hiroshima, Japan

Cartesian Ant Programming Introducing Symbiotic Relationship between Ants and Aphids

Jun-ichi Kushida, Akira Hara and Tetsuyuki Takahama
Graduate School of Information Sciences
Hiroshima City University
Hiroshima, Japan
{kushida, ahara, takahama}@hiroshima-cu.ac.jp

Naoya Mimura
Faculty of Information Sciences
Hiroshima City University
Hiroshima, Japan
nmimura@ins.info.hiroshima-cu.ac.jp

Abstract—**Cartesian Ant Programming (CAP) is one of the swarm-based automatic programming method, which combines graph representations in Cartesian Genetic Programming with search mechanism of Ant Colony Optimization. CAP generates a graph structure program based on pheromone sprayed on routes. By using pheromone communication, ants can search the promising solution space intensively. However, once the pheromone is convergence, ants concentrate on the certain route and search ability of diversification decrease. In this research, in order to solve this problem we propose a modified CAP introducing symbiotic relationship between ants and aphids. In the CAP developed in this research, we consider nodes to be aphids coexisting with ants, and ants consider not only pheromone communication, but also secretions secreted by the aphid and generates a program. We examined the effectiveness of our proposed method by applying it to symbolic regression problems. Through experimental results, we confirmed that The performance of the proposed method outperformed conventional CAP.**

Index Terms—**Cartesian Ant Programming, Ant Colony Optimization, Symbiotic relationship**

I. Introduction

Cartesian Ant Programming (CAP) [1], [2] is a method that adopts both approaches of route selection method based on pheromone concentration in Ant Colony Optimization (ACO) [3], [4] and graph representation of individuals in Cartesian Genetic Programming (CGP) [5]. CAP uses a feedforward graph structure as an expression form and nodes with function symbols that are components of programs are arranged in a lattice pattern. Ants generate programs by moving in the node-network based on pheromone. The program generated by the CAP is a graph structure program, which has an advantage that the partial graph structure can be reused.

In CAP, the nonterminal symbols of each node are fixedly assigned beforehand at the beginning of search. Therefore, the expressive capacity of the program generated by ants depends on the initial placement of nonterminal symbols. Furthermore, with only search based on pheromone, ants tend to concentrate on specific routes, making it difficult to realize appropriate balance of diversification and concentration. Especially, in order to solve with recent large-scale and complicated optimization problems, it is necessary to cope with these problems. Therefore, we have proposed several modified CAP algorithms [6], [7].

In this research, we focus on the symbiotic relationship between ants and other insects in order to improve the search performance of CAP. Ants in nature are known not only to communicate between ants by pheromones but also to have a symbiotic relationship with other insects. Aphids are insects that live on plants and establishes symbiotic relationship with ants. Ants feed on the sugary honeydew provided by aphids. In exchange, the ants protect the aphids from predators or parasites. In order to build such a relationship of harmonious coexistence, aphids always need to secrete foods that attract many ants.

Even in CAP, if a node has a function symbol required by ants, many ants will visit to it. As a result, ants can also abundantly use the required symbols and generate better programs. In other words, the relationship between ants and aphids in the natural world can correspond to the relationship between ants and nodes in CAP. By considering such relationship of symbiosis and incorporating it into the algorithm of CAP, we propose a novel CAP to realize a good balance between intensification and diversification. In the proposed method, there are aphids in the environment of ants transition, and ant determines routes by not only pheromone but also honeydew of aphids.

This paper is organized as follows: In Section II, we explain algorithm of CAP. Next, in Section III, we propose a CAP introducing symbiotic relationship between ants and aphids. Then, in Section IV, we describe our experimental results and discussions. Finally, in Section V, we describe conclusions and future work.

II. Cartesian Ant Programming (CAP)

CAP is based on the graph representation in CGP has been combined with the route selection mechanism of ants based on the pheromone intensity in ACO. Similar to CGP, ACO requires the definition of the topology of the graph (i.e., number of rows and columns), input set, function set, number of outputs, number of input connections to a node and the level-back parameter.

A. Definitions of CAP parameters

The CAP algorithm is defined by using the following symbols.

978-1-5386-0469-4/17/$31.00

T	The set of Terminals
n_i	The number of Terminals (program inputs)
n_o	The number of program outputs
n_n	The number of input connections for each node
F	The set of Functions
n_f	The number of Functions
n_r	The number of rows in the grid (fixed to 1)
n_c	The number of columns in the grid
P	Pheromone table
l	Level back parameter, which determines how many column ahead the ant can move to from the current column.
z_i	Current position of the ant i
S_a	The set of node numbers which ant a can move to in the next move.
m_{ni}	The number of each node's outputs (fixed to 1 in this paper)
M	The number of ants
t_{max}	Max time step

B. Program Construction

Nodes are arranged in a grid, and the nodes are numbered from input nodes to output nodes. The functions of respective nodes are sequentially and periodically assigned, and the assigned functional symbols are fixed during the search. Each ant starts from the program output, and it can move to the nodes with smaller number than its current node.

Consider the case where the ant a is in the k-th input of node j. Parameter l is the level back parameter, which determines how many column ahead the ant can move to from the current column. In the case of $l \geq n_c$, the ant can move to any nodes in the left-hand of the current position. Therefore, the set of node numbers which ant a can move to in the next move is $S_a = \{0, 1, \ldots, j-1\}$.

The next destination node $i \in S_a$ is determined probabilistically by the node transition rule in Eq.(1).

$$p_{j-k,i} = \frac{\tau_{ijk}}{\sum_{l \in S_a} \tau_{ljk}} \tag{1}$$

where τ_{ijk} is the pheromone intensity between the current position (k-th input of node j) and the node $i \in S_a$. The notation of pheromone τ_{ijk} is shown in Fig. 2. The ant moves to the next destination till it reaches to the terminal nodes. If it reaches to a terminal node, it backtracks to the node which has the unconnected inputs, and restarts to move from an unconnected input. If all nodes which have been passed through by the ant have no unconnected inputs, a graph structural program has been generated completely.

Fig. 1 shows an example of the search by an ant. In this figure, the number of terminals (program inputs) n_i is two and the number of program outputs n_o is one. The set of terminals T is $\{1.0, x\}$ and the set of functions F is $\{+, -, *\}$. The numbers of rows n_r and columns n_c in the grid are set to $n_r = 1$ and $n_c = 3$ respectively. The functions are assigned to the respective nodes sequentially. The levelback parameter is set to $l = n_c$. TABLE I shows the pheromone table corresponding to the environment in Fig. 1. In this table, the leftmost column means the ant's current position. The first row means the next destination of the ants. The ant starts from Node 5 (Output) toward the Terminals (Node 0 or 1) based on the pheromone information. The selected connections of nodes in Fig. 1 represent the function $f(x) = 1.0 - x^2$.

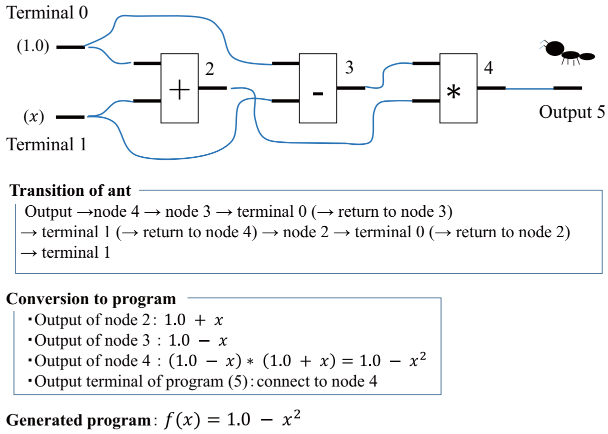

Fig. 1. An example of the transition of the ant in CAP.

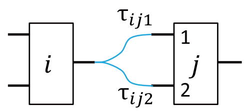

Fig. 2. Notation of pheromone τ_{ijk}.

C. Pheromone update

The graph structural programs generated by m ants are evaluated. Then, the pheromone table is updated based on the fitness of the best program in the step like Max-Min Ant System (MMAS) [8]. The pheromone update rule is shown in Eq.(2).

$$\tau_{ijk}(t+1) = (1-\rho)\tau_{ijk}(t) + \Delta\tau_{ijk}^{best}, \tag{2}$$

where the amount of additional pheromone $\Delta\tau_{ijk}^{best}$ is defined according to the problems. In addition, to avoid stagnation of search, the range of possible amount of pheromone on each edge is limited to an interval $[\tau_{\min}, \tau_{\max}]$ as similar to MMAS [8]. These procedures are repeated t_{max} times.

TABLE I
PHEROMONE TABLE FOR THE ENVIRONMENT IN FIG. 1.

	0	1	2	3	4
2-1	τ_{021}	τ_{121}	-	-	-
2-2	τ_{022}	τ_{122}	-	-	-
3-1	τ_{031}	τ_{131}	τ_{231}	-	-
3-2	τ_{032}	τ_{132}	τ_{232}	-	-
4-1	τ_{041}	τ_{141}	τ_{241}	τ_{341}	-
4-2	τ_{042}	τ_{142}	τ_{242}	τ_{342}	-
5 (Output)	τ_{05}	τ_{15}	τ_{25}	τ_{35}	τ_{45}

III. CAP INTRODUCING SYMBIOTIC RELATIONSHIP BETWEEN ANTS AND APHIDS

Ants and aphids in nature have a symbiotic relationship. Ants feed on the honeydew provided by aphids. In exchange, the ants protect the aphids from predators or parasites. In order to build such a relationship of harmonious coexistence, aphids always need to secrete foods that attract many ants. We incorporate such a relationship of symbiosis into the algorithm of CAP, and propose a novel CAP to realized a good balance between intensification and diversification. In the proposed CAP, there is an aphid at each node, and it produces an honeydew corresponding to the function symbol. Each ant moves within the environment based on the amount of pheromone on the route and the amount of honeydew of each node, and generates a graph structure program. At the beginning of each step, the amount of honeydew of each node is set to same value. In the node visited by ants, the honeydew decreases by a certain amount. Fig. 3 shows the environment of ants in the proposed CAP.

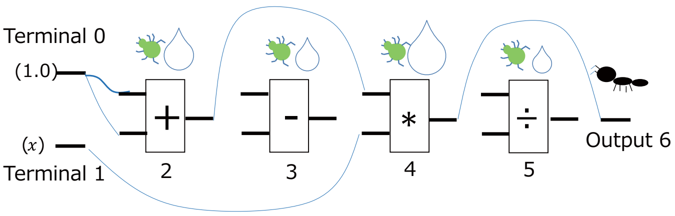

Fig. 3. Environment of ants in the proposed CAP

The procedure of the proposed method is as follows.

(1) Initialize pheromone and honeydew

(2) The following processing is repeated until the termination condition is satisfied

 (3.1) Generate program based on transition rule using pheromone and amount of honeydew

 (3.2) Evaluate the program generated by each ant

 (3.3) Update pheromone

 (3.4) Update honeydew

 (3.5) Replace function symbols for nodes with a low utilization

(4) Output the generated best solution

In the proposed method, update honeydew and replacement of function symbols are added to the conventional CAP. Details of each processes are described in the following sections.

A. Transition rule

In step t, transition rule from k-th input of node j to node $i \in S_a$ is as follows:

$$p_{j-k,i} = \frac{[\tau_{ijk}]^\alpha [\eta_i^{t-1}]^\beta}{\sum_{l \in \mathcal{S}^a} [\tau_{ljk}]^\alpha [\eta_l^{t-1}]^\beta}, \tag{3}$$

where τ_{ijk} is the pheromone intensity between the current position (k-th input of node j) and the node $i \in S_a$, and η_i^{t-1} represents the amount of honeydew of node i at previous step (step $t-1$). α and β are the weight of pheromone and honeydew, respectively.

B. Update of honeydew

The amount of honeydew of each node is set to 1 at the start of the step. The update of honeydew is performed after evaluation of program is finished. In order to greatly reduce the honeydew of the node where many ants visited, the amount of honeydew in each node is updated by the following equation.

$$\eta_i^t = 1 - \frac{c_i^t}{M+1} \tag{4}$$

where, c_i^t is the number of ants which visited node i in step t, except backtracking. In step t, the amount of honeydew at the previous step, η_i^{t-1}, is used in the route selection, η_i^t is obtained by Eq.(4) for next step.

In step t, previous step's η_i^{t-1} is used in the route selection, and η_i^t is updated for next step. That is, if ants concentrate to a certain node, the node is less likely to be used in the next step. By repeating such a transition of honeydew, ants will visit various nodes in the environment periodically.

C. Replacement of function symbols of nodes

Aphids ' ingredients of honeydew is greatly related to the induction of ants and aphids change the ingredients according to ant preference. In the proposed method, we introduce this habit of aphids into the CAP and randomly change function symbols of nodes with low utilization. In addition, the node that changed the symbol increases the amount of pheromone connected to it and encourages visit of ant.

(1) At step t, if node i satisfies the following condition, node i is regarded as an unnecessary node.

$$\tau_{ijk} = \tau_{min} \ \ \forall_{j,k} \in T_i \tag{5}$$

where, T_i is a set of all routes connecting the node i and the output side. $count_i$ is the number of steps that become consecutive unnecessary nodes.

(2) For each node, the number of consecutive unnecessary nodes is counted. If node i is an unnecessary node, $count_i$ increases by 1. Otherwise, set the $count_i$ to 0. The node i with $count_i >= 100$ becomes target of replacement of the function symbol.

(3) If there is a node to be replaced, it's symbol is randomly replaced by another symbol. If there are multiple nodes for replacement, one node is selected among them. To select node for replacement, the following three methods are used.

 (a) Select the node closest to the terminal symbol

 (b) Select the node closest to the output

 (c) Select randomly from whole target nodes

 For node i with function symbol replaced, count i is set to zero.

(4) The amount of pheromone of all paths connected to the output side from the node i with the symbol replaced are adjusted by the following equation.

$$\tau_{ijk} = \tau_{min} + (\tau_{max} - \tau_{min}) * \epsilon \ \ \forall_{j,k} \in T_i \tag{6}$$

where ϵ is a parameter to adjust the increase of pheromone.

IV. Experimental Results and Discussions

A. Experimental Settings

In this section, we examine the effectiveness of our proposed method by comparing the search performance with the conventional CAP in symbolic regression problems. The goal of the symbolic regression is to find an approximating numerical expression of objective function $f(x)$ from given input-output data. In our experiments, the objective functions are follows.

- $f_1 = 11x^2 - 13x + 17$
- $f_2 = x^5 + x^4 + x^3 + x^2 + x$
- $f_3 = \sin(x^2)\cos(x) - 1$

Fitness $fitness$ is calculated by the sum of absolute errors between the objective function $f(x)$ and the generated function $g(x)$ for 20 random values of x in $[-1.0, 1.0]$ as shown in Eq. (7).

$$fitness = \sum_{d=1}^{20} |f(x_d) - g(x_d)|, \tag{7}$$

where $x_d (d = 1, \ldots, 20)$ represents an input variable.

Pheromone update rules for node connections is performed by Eq.(8).

$$\Delta\tau_{ijk}^{best} = \begin{cases} \frac{1}{fitness^{best}+1} & \text{if } (i,j,k) \in T^{best} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $fitness^{best}$ is the fitness of the best ants, and T^{best} is the best program. (i, j, k) is the connection between the k-th input of node j and the node i.

TABLE II shows common parameters in our experiments. In the proposed method, the ratio between α and β greatly affects the search performance. α is set to 1 and the ratio is determined by changing the β. For the parameter β, we try two setting method: one uses fixed value (fixed β), and the other decreases β from β_{max} to 0 during the searching process (decreasing β) by Eq. (9).

$$\beta = \beta_{max} \times \frac{t_{max} - t}{t_{max}}, \tag{9}$$

where t represents the current step. The weight decreases gradually as the search proceeds.

TABLE II

Parameter Settings.

Parameter	Value
The number of program outputs n_o	1
The number of Terminals n_i	2
The set of Terminals T	$\{1.0, x\}$
The set of Functions F	$\{+, -, *, div\}$
The number of rows in the grid n_r	1
The number of columns in the grid n_c	100
Level back parameter l	$n_i + n_c$
Max time step t_{max}	10000
The number of ants M	50
Max pheromone τ_{max}	10
Min pheromone τ_{min}	0.01
Weight for pheromone information α	1.0
Evaporation rate ρ	0.1
The number of trials W	30

B. Experimental Results

First, we confirm impact of β in the proposed CAP. We compare fixed β and decreasing β using the proposed CAP without replacement of function symbol. Each method ran 30 trials for each problem with β and $\beta_{max} = \{0, 1, 2, 3, 4, 5, 6, 7, 10, 15\}$. Fig. 4, 5 and 6 show the comparisons of mean fitness of fixed β and decreasing β for the objective functions f_1, f_2 and f_3 respectively. For all functions, it can be confirmed that decreasing β shows better results than fixed β. This seems to be because the balance of intensification and diversification is appropriately changed according to the search progress by Eq. (9). From these results, at the early stage of search various paths generated by considering honeydew is required, and concentration to promising path using pheromones is effective at the end of the search.

Next, we compare conventional CAP, the CAP without replacement and CAP with replacement using β and $\beta_{max} = 5$. TABLE III and IV show the comparisons of mean fitness of fixed β and decreasing β for the objective functions f_1 and and f_3 respectively. In function f_2, since the success rate is 100% in all the methods, the result is not shown. In these tables, "Best" is the average value of the best solution obtained in each trial. "Std" is the standard deviation of the best solution for each trial. "Success Rate" means the rate of finding the optimal solution over all trials. "#Effective Nodes" means the number of nodes which are connected to the output nodes directly or implicitly in the acquired best program. Also, "Terminal", "Output" and "Random" indicate that nodes of function symbol to be replaced are selected as terminal symbol side, output side, random, respectively.

In several setting, CAP with replacement shows better performance than CAP without replacement. Especially, in function f_3 all setting of CAP with replacement outperformed clearly conventional CAP. Furthermore, the #Effective Nodes of the proposed method is smaller than conventional CAP, and it can be seen that the proposed method is able to generate a compact program. However, in function f_1 if the node of terminal side is selected for replacement, the performance of the proposed method degrades. The node of terminal side corresponds to to the node near the root of tree structure program. Therefore, the change of the node of terminal side brings big change in the program and it may cause slow convergence to the promising route. For these reasons, to incorporate replacement of the function symbol, we think that it is better to target the node of the output side and set a small ϵ.

V. Conclusion

Ants in nature develop symbiotic relationships with other organisms, and one of the example is ants and aphids. Ants feed on the honeydew provided by aphids. In exchange, the ants protect the aphids from predators or parasites. In this

TABLE III

RESULT OF EACH METHOD IN FUNCTION f_1

	Best	Std	#Effective Nodes	Success Rate
CAP	1.572	1.912	30.9	16%
CAP w/o replacement	0.857	1.136	24.3	46%
Proposed CAP, Terminal,ϵ=0	0.905	1.230	25.2	43%
Proposed CAP, Terminal,ϵ=0.05	0.808	1.083	21.7	56%
Proposed CAP, Terminal,ϵ=0.1	2.025	2.016	20.2	40%
Proposed CAP, Output,ϵ=0	0.950	1.338	23.8	50%
Proposed CAP, Output,ϵ=0.05	0.634	1.085	21.5	**70%**
Proposed CAP, Output,ϵ=0.1	1.518	1.777	19.4	66%
Proposed CAP, Random,ϵ=0	**0.549**	1.007	22.0	63%
Proposed CAP, Random,ϵ=0.05	0.867	1.262	22.0	63%
Proposed CAP, Random,ϵ=0.1	1.505	1.697	21.5	50%

TABLE IV

RESULT OF EACH METHOD IN FUNCTION f_3

	Best	Std	#Effective Nodes	Success Rate
CAP	0.212	0.317	30.600	3%
CAP w/o replacement	0.079	0.074	26.233	10%
Proposed CAP, Terminal,ϵ=0	0.060	0.055	25.966	13%
Proposed CAP, Terminal,ϵ=0.05	0.061	0.054	25.433	3%
Proposed CAP, Terminal,ϵ=0.1	0.064	0.037	19.766	6%
Proposed CAP, Output,ϵ=0	0.057	0.067	24.833	**23%**
Proposed CAP, Output,ϵ=0.05	**0.055**	0.057	26.733	13%
Proposed CAP, Output,ϵ=0.1	0.077	0.054	20.166	3%
Proposed CAP, Random,ϵ=0	0.068	0.061	27.466	10%
Proposed CAP, Random,ϵ=0.05	0.056	0.056	26.633	3%
Proposed CAP, Random,ϵ=0.1	0.060	0.050	19.100	10%

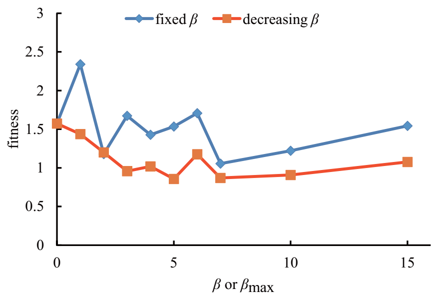

Fig. 4. Mean fitness of fixed β and decreasing β in function f_1

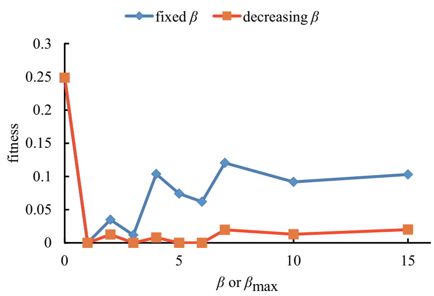

Fig. 5. Mean fitness of fixed β and decreasing β in function f_2

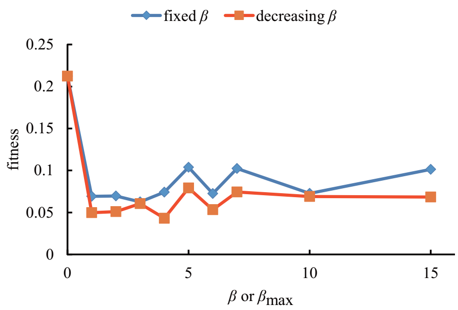

Fig. 6. Mean fitness of fixed β and decresing β in function f_3

paper, we incorporated such a relationship of symbiosis into the algorithm of CAP, and proposed a novel CAP to realized a good balance between intensification and diversification. In the proposed method, we introduced new transition rule that takes into consideration the amount of honeydew on each node, and added replacement of function symbol at unnecessary node. We examined the effectiveness of our proposed method in symbolic regression problems. The performance of the proposed method outperformed conventional CAP in all objective functions. However, the search performance of the proposed method depends on setting method of parameter β and replacement method of function symbol. In the future we plan to propose a method to adaptively control these parameters. Furthermore, we will apply the proposed method to various types of problems such as automatic circuit design or multi-agent control and verify the effectiveness.

REFERENCES

[1] A. Hara, M. Watanabe, T. Takahama, "Cartesian Ant Programming", Proceeding of 2011 IEEE International Conference on Systems,Man,and Cybernetics, pp.3161-3166, 2011.

[2] S. Luis and M. V. dos Santos, "On the Evolvability of a Hybrid Ant Colony-Cartesian Genetic Programming Methodology", Proceedings of 16th European Conference on Genetic Programming (EuroGP2013), LNCS 7831, pp.109-120, 2013.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence, Oxford University Press, 1999.

[4] M. Dorigo, and T. Stutzle, Ant Colony Optimization, MIT Press, 2004

[5] J. F. Miller and P. Thomson, "Cartesian Genetic Programming", Proceedings of the Third European Conference on Genetic Programming (EuroGP2000), pp.121-130, Springer-Verlag, 2000.

[6] A. Hara, J. Kushida, K. Fukuhara and T. Takahama, "Cartesian Ant Programming with adaptive node replacements", Proceedings of IEEE 7th International Workshop on Computational Intelligence and Applications (IWCIA 2014), pp. 119-124, 2014.

[7] A. Hara, J. Kushida, K. Fukuhara and T. Takahama, "Cartesian ant programming with transition rule considering internode distance", Proceedings of IEEE 9th International Workshop on Computational Intelligence and Applications (IWCIA 2016), pp. 101-105, 2016.

[8] T. Stutzle and H. Hoos,"Improving the Ant System: A Detailed Report on the $\mathcal{MAX}-\mathcal{MIN}$ Ant System", Technical Report AIDA-96-12, FG Intellektik, TH Darmstadt, Aug. 1996.