Hardware-oriented Ant Colony Optimization Considering Intensification and Diversification

Masaya Yoshikawa
Meijo University
Japan

1. Introduction

Swarm intelligence is the technological modeling of behaviors of social insects, such as the ant or the honeybee. Although each element comprising swarm intelligence is simple, high grade intelligence emerges when the elements gather to form a swarm. Ant Colony Optimization (Dorigo, M, et al., 1997), which is called ACO, is one of the swarm intelligence and has been attracting much attention recently. The ACO represents a general name of the algorithm inspired by feeding behavior of ants. It has been applied to various combinatorial optimization problems (Ramkumar, A.S. et al., 2006), including the travelling salesman problem (TSP), the floorplanning problem (Luo, R., et al., 2007) and the scheduling problem (Sankar, S.S., et al., 2005). The basic model of the ACO is the ant system (AS) that was proposed by Dorigo et al. (1996), and many ACOs applied to TSP are based on the AS. However, these ACOs require a lot of calculation time, because the optimization process is based on repetitive searches by plural numbers of ants.
In this chapter, a novel hardware-oriented ACO (H-ACO) is proposed to achieve high-speed optimization based on mechanism of ACO algorithm. The characteristics of the H-ACO is as follows: (1) all calculations can be performed with only addition, subtraction, and shift operation, instead of the floating point arithmetic and power calculation which are adopted in conventional ACO; (2) a new technique using Look-Up-Table (LUT) is introduced; and (3) in addition to upper and lower limits, benchmarks are set to the pheromone amount. Experiments using benchmark data prove effectiveness of the proposed algorithm.
The organization of this chapter is as follows: Section 2 describes the search mechanism of ACO and briefly surveys the ACO research in terms of the computational time. Section 3 explains H-ACO. Section 4 reports the results of computer simulations applied to travelling salesman problem. Section 5 summarizes the chapter.

2. Preliminaries

2.1 Ant colony optimization

Ant Colony System is one of the expansion algorithm of AS, and it shows better capability than genetic algorithm and simulated annealing when applying to TSP. Therefore, we adopt Ant Colony System as a base algorithm and the target problem is TSP. Hereafter, ACO indicates Ant Colony System.

Source: Advances in Greedy Algorithms, Book edited by: Witold Bednorz,
ISBN 978-953-7619-27-5, pp. 586, November 2008, I-Tech, Vienna, Austria

The search mechanism of ACO utilizes the static evaluation value and the dynamic one. The static evaluation value called heuristic value is peculiar information of the target problem, and the dynamic evaluation value is pheromone amount. Usually, a reciprocal number of the distance is adopted as the heuristic value, when ACO is applied to TSP. Specifically, ant k in city i selects the move to city j according to probability p^k and it is defined as follows.

$$p^k(i,j) = \frac{[\tau(i,j)][\eta(i,j)]^{\beta}}{\sum_{l \in n^k}[\tau(l,j)][\eta(l,j)]^{\beta}} \tag{1}$$

Where, $\tau(i,j)$ is a pheromone value between city i and city j, $\eta(i,j)$ is a reciprocal number of the distance between city i and city j, β is a parameter which controls the balance between static evaluation value and dynamic one, and n^k is a set of un-visit cities.
On the other hand, a pheromone amount on each route is calculated by using two pheromone update rules. One is a local update rule and the other is a global update rule. The local update rule is applied to the route which is selected by equation (1), and it is defined as follows.

$$\tau(i,j) \leftarrow (1-\psi)\tau(i,j) + \psi\tau_0 \tag{2}$$

Where, ψ is a decay parameter in the local update rule, $\tau 0$ is initial value of pheromone. Thus, the local update rule adds the pheromone to the selected route when the ant moves. The global update rule adds pheromone to the best tour (the completed route) of all tours. The best tour usually indicates the shortest tour. The global update rule is defined as follows.

$$\tau(i,j) \leftarrow (1-\rho)\tau(i,j) + \rho\Delta\tau(i,j)$$

$$\Delta\tau(i,j) = \begin{cases} 1/L^+ & if\ (i,j) \in T^+ \\ 0 & otherwise \end{cases} \tag{3}$$

Where, T^+ is the best tour, and L^+ is the total distance of the best tour.

2.2 Related work

Examples of dedicated hardware for ACO are found by Haibin et al. (2007), Nakano et al. (2006), and others. Haibin et al. (2007) proposed the hardware architecture that combines genetic algorithm and ACO, and they showed the validity of the combined architecture. Nakano et al. (2006) implemented the partial function of ACO on FPGA (Filed Programmable Gate Array), and they demonstrated more high-speed than software processing. The authors also proposed the dedicated hardware of ACO (Yoshikawa, M., et al., 2007). However, the developed hardware was based on ordinary ACO algorithm, and adopted floating point arithmetic as calculation of pheromone update rules.
Regarding other meta-heuristic algorithm, the dedicated hardware approach to reduce calculation time are reported by Scotte et al. (1995), Imai at al. (2002), and Frye et al.(1991). Scott et al. (1995) developed a hardware-based GA and demonstrated its superiority to software in speed and solution quality. Imai et al. (2002) proposed a processor element constituting parallel GA, and achieved the parallelism due to the number effect. Frye et al. (1991) developed the dedicated hardware for neural network using analog device.

Thus, no studies have ever seen, to our knowledge, the hardware oriented ACO algorithm which does not utilize floating point arithmetic.

3. Hardware-oriented ACO

3.1 Pheromone update rule

In the H-ACO, in order to control the trade-off between intensification (exploitation of the previous solutions) and diversification (exploration of the search space), upper and lower limits are set in a manner similar to the Max-Min AS. Here, a new benchmark of the pheromone is also introduced. Using this benchmark, an increment of the pheromone is determined. An example in which the pheromone is added is shown in Fig.1, where the horizontal axis indicates the time and the vertical axis indicates the pheromone value.

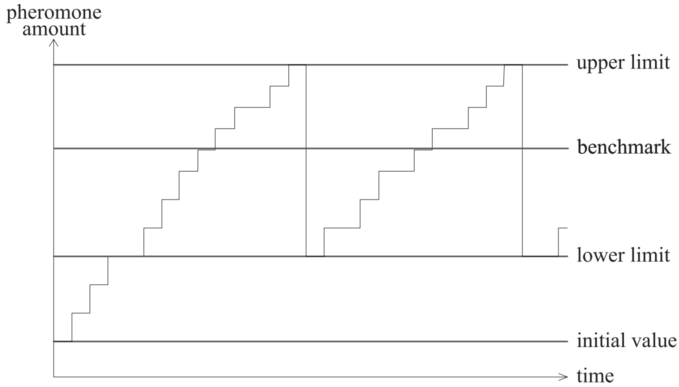


Fig. 1. Example of transition of pheromone

The pheromone value (pheromone amount) is added by performing the search starting from the initial value $\tau 0$. When the pheromone value is smaller than the benchmark, a large number of the pheromones are added from a viewpoint of intensification.

On the other hand, when the pheromone value is larger than the benchmark, a small number of pheromones are added to diversify the search space.

When a particular tour (route) has a large number of pheromones, this indicates that the tour is often selected. When the pheromone value reached the upper limit, the pheromone value is reduced to the lower limit. By this operation, the probability of other tours being selected is increased from a viewpoint of diversification. In other words, the H-ACO is controlled to perform a global search in this case.

A new equation (4) is introduced to the local update rule and it is defined as follows.

$$\tau(i,j) = \tau + \beta$$

$$\beta = \begin{cases} 8 & if \;\; 8 < \tau < std \\ 4 & if \;\; std < \tau < max \\ 0 & if \;\; 8 = \tau \end{cases} \tag{4}$$

Where, *std* represents the benchmark, *max* is upper limit, and the initial value of pheromone ($\tau 0$) is set to 8. In the local update rule, if $\tau(i,j) = 8$, the increment is 0, i.e., no local update will

be performed. In other words, trapping by a local optimal solution can be avoided without adding excessive pheromones to the tour at early stages of the search.
Moreover, high-speed processing can be also realized by reducing the number of processing steps. For example, when the numbers of ants (agents) and cities are denoted as m and n, respectively, the number of processing steps required for the local update rule of $m \times n$ times can be reduced in the first search.
The global update rule is defined by the equation (5).

$$\tau(i,j) = \tau + \varepsilon$$

$$\varepsilon = \begin{cases} 8 & if \quad \tau < std \\ 0 & otherwise \end{cases} \tag{5}$$

As regards the global update rule, if it is applied only when the pheromone value is smaller than the benchmark, a larger global search can be performed.
Fig. 2 shows an example in which the local and global update rules are applied. As shown in the figure, the local update rule is only performed to the tours that pheromone have been added to in the global update rule.

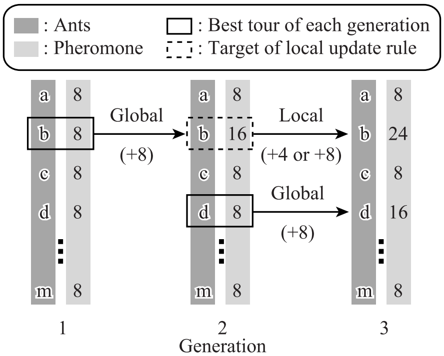


Fig. 2. Example of local update rule and global update rule

That is, in the H-ACO, a global search is performed at the early stage of search, and a local search is performed as the search progressed. Thus, H-ACO can achieve not only speed-up of the pheromone update procedure by reducing the number of processing steps, but also effective search by controlling of intensification and diversification.

3.2 Selection method using Look-Up-Table technique

In the general ACO, when the city of the next destination is selected, a power calculation, as shown in equation (1), is required. In addition, the heuristics value, as the information peculiar to a certain problem, is a decimal because it is the reciprocal of distance. Therefore, when the heuristics value is realized in a dedicated hardware, a floating point arithmetic unit is required. To simplify hardware resources, the number of processing steps and the control, however, power calculations and floating point arithmetic operations are not suitable for the hardware.
In the H-ACO, a selection technique based on the LUT system is introduced. As the heuristics value, the value that is obtained by converting the reciprocal of distance to the

positive integer is used. As the pheromone value, only the values that are multiples of 4, as shown in the equation (4) and (5) of the pheromone update rule, are used. Then, the LUT, into which both the heuristics and pheromone values are input, is created. Fig.3 shows an example of the LUT. By using the integral value and the LUT system, it ensures that all the operations can be performed by simple addition, subtraction, and shift operations.

$\eta(i, j)$ \ $\tau(i, j)$	8	16	24	32	· · ·
0 - 99	8	16	24	32	· · ·
100 - 199	8	16	24	32	· · ·
200 - 299	16	24	32	40	· · ·
300 - 399	24	32	40	48	· · ·
	⋮	⋮	⋮	⋮	

Fig. 3. Example of Look-Up-Table

4. Experiments and discussion

In order to verify the validity of the H-ACO, several comparative experiments are conducted. First, in order to evaluate search performance, the H-ACO is compared with the conventional ACO described in Section 2. 1. The experimental platform is a Pentium 4 3.0 GHz and the program is described by the C language. As experimental data, the original data from 50 cities, in which the optimal solution is already known, and the travelling salesman problem library (TSP.LIB) benchmark data of 100 cities are used. The experimental results are shown in Figs. 4 and 5.

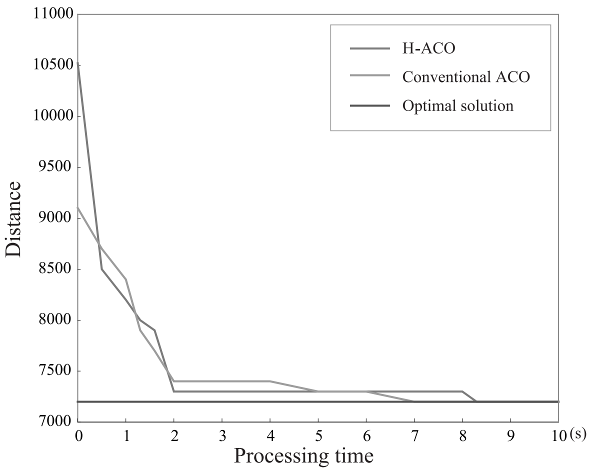


Fig. 4. Result of 50 cities

In both figures, the horizontal axis indicates the processing time and the vertical axis indicates the total distance of the route. As shown in both figures, the search performance of

the H-ACO, which does not use decimal operation and power calculation, is similar to that of the conventional ACO, which does require their use.
Next, an experiment to evaluate the controllability of parameters is conducted. In the conventional ACO, the balance of the pheromone value and the heuristics value is controlled by parameter β in the equation (1) and the balance of the information on the past behaviour and that on the new behavior is controlled by decay parameters ψ and ρ (the evaporation rate) in the equations (2) and (3).
In the H-ACO, the LUT value, upper limit, lower limit, and benchmark are used as the parameters, instead of parameters ψ, ρ and β, to control these balances.
The experimental results with various LUT values are shown in Fig. 6. Fig.6 (1) shows the result of setting the pheromone value larger than the heuristics value.

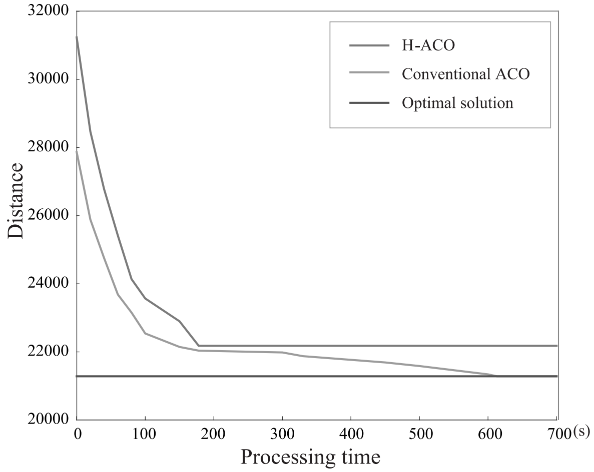
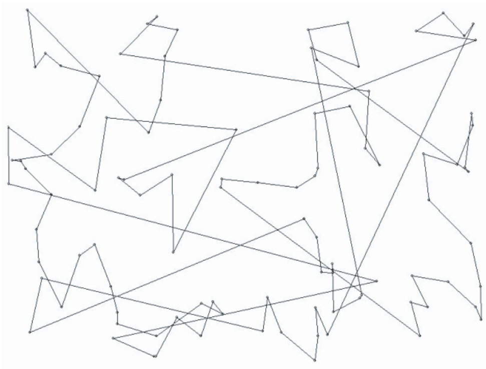
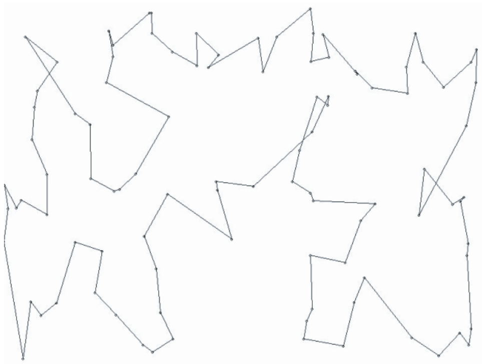


Fig. 5. Result of 100 cities

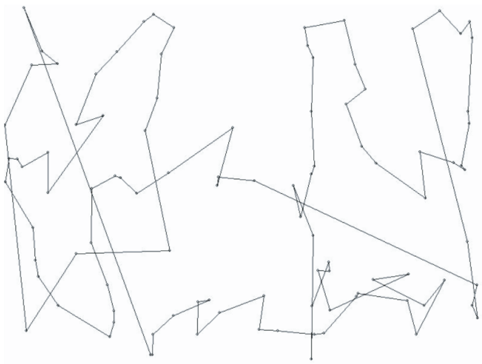
Fig.6. (3) shows the result when the pheromone value is set to be smaller than the heuristics value.
As shown in Fig.6 (1), since the influence of the heuristics value (in this case, it is the distance between the cities) is small, this search is similar to a random search and many tours are intersected.
As shown in Fig.6 (3), since the influence of the distance between the cities is great, the cities (the distance between which is small) are selected in the initial stage of the route construction; that is, a behavior similar to the greedy algorithm is observed. Therefore, in the final stage, to complete the route, cities with large distances between them are selected.
As shown in Fig.6 (2), the pheromone value and the heuristics value are well-balanced, and an effective search is realized. Thus, the technique of using the LUT value, which has been newly introduced to the H-ACO, instead of parameter β and power calculation, which are used in the conventional ACO, is clearly shown to provide an effective selection of the cities with well-balanced pheromone and heuristics values.



(1) $\tau(i, j)$ is larger than $\eta(i, j)$.



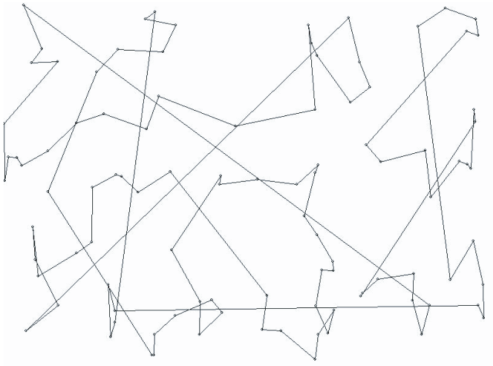
(2) Good balance between $\tau(i, j)$ and $\eta(i, j)$.



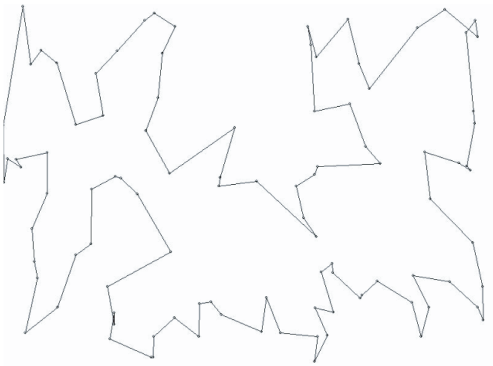
(3) $\tau(i, j)$ is smaller than $\eta(i, j)$.

Fig. 6. Result of several kinds of LUTs

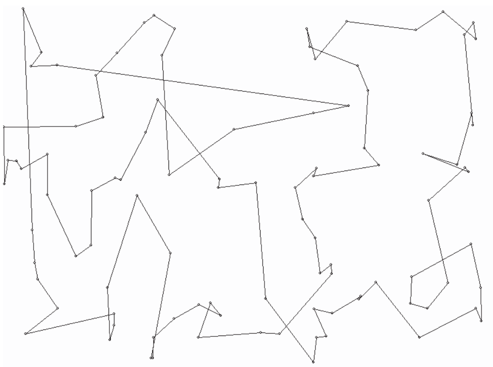
Fig. 7 shows the results of the experiments in which various upper limits and benchmarks of the pheromone value are used.



(1) Both value are high.



(2) Both value are middle.



(3) Both value are low.

Fig. 7. Result of several pairs of upper limit and benchmark

In these experiments, since the relative relationship between the upper limit and the benchmark is considered as maintained even if the lower limit is fixed, only the upper limit and the benchmark are changed. Figs.7 (1), (3), and (2) show the results of the experiments, in which the upper limit and the benchmark are set to be high, low, and intermediate, respectively.

As shown in Fig.7 (1), since the upper limit was set high, pheromone is accumulated over a long period. This means that, since the past information is considered important, no progress is observed in the search.

As shown in Fig.7 (3), since the distance between the upper limit and the benchmark is small due to the low upper limit, this search is similar to a random search.

In contrast, as shown in Fig.7 (2), when the upper limit and the benchmark are well-balanced, a satisfactory solution is obtained.

Thus, simply by adjusting the upper limit and the benchmark, the same effect as using the decay parameters, which controlled the information on the past behavior and the information on the new behavior, can be realized. Based on the above experimental results, the proposed H-ACO is confirmed to provide a similar solution searching mechanism and ability as seen with the conventional ACO, without the need for floating point arithmetic operations and power calculations.

5. Conclusion

In this chapter, we proposed a novel hardware-oriented ACO algorithm. The proposed algorithm introduced new pheromone update rules using the LUT. It enabled all calculations for optimization with only addition, subtraction, and shift operation. Moreover, it controlled the trade-off between exploitation of the previous solutions and exploration of the search space effectively. As a result, the proposed algorithm achieved not only high speed processing, but also maintenance of the quality of solutions. Experiments using benchmark data proved effectiveness of the proposed algorithm.

6. References

Dorigo, M., Maniezzo, V. & Colorni,A.(1996). Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol.26, No.1,pp.29-41.

Dorigo, M. & Gambardella, L.M.(1997). Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, Vol.1, No.1, pp.53-66.

Frye, R.C., Rietman, E.A. & Wong, C.C. (1991). Back-propagation learning and nonidealities in analog neural network hardware, *IEEE Transactions on Neural Networks*, Vol.2, No.1, pp.110-117.

Haibin, D. & Xiufen, Y. (2007). Progresses and Challenges of Ant Colony Optimization-Based Evolvable Hardware, *Proceedings of IEEE Workshop on Evolvable and Adaptive Hardware*, pp.67-71.

Imai, T., Yoshikawa, M., Terai, H. & Yamauchi, H. (2002).Scalable GA-Processor Architecture and Its Implementation of Processor Element, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol.3, pp.3148-3151.

Luo, R. & Sun, P.(2007). A Novel Ant Colony Optimization Based Temperature-Aware Floorplanning Algorithm, *Proceedings of Third International Conference on Natural Computation*, Vol.4, pp.751-755.

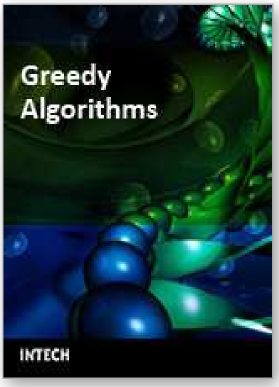
Nakano, M., Iida, M. & Sueyoshi, T. (2006). An Implementation of Ant Colony Optimization for the MaTriX Processing Engine, *Proceedings of IEICE-RECONF2006-27*, Vol.106, No.247, pp.1-6.

Ramkumar, A.S. & Ponnambalam, S.G. (2006). Hybrid Ant colony System for solving Quadratic Assignment Formulation of Machine Layout Problems, *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems*, pp.1-5.

Sankar, S.S., Ponnambalam, S.G., Rathinavel, V. & Visveshvaren, M.S. (2005). Scheduling in parallel machine shop: an ant colony optimization approach, *Proceedings of IEEE International Conference on Industrial Technology*, pp.276-280.

Scott, S.D., Samal, A. & Seth, S. (1995). HGA:A Hardware-Based Genetic Algorithm, *Proceedings of Iternational Symposium on Field-Programmable Gate Array*, pp.53-59.

Yoshikawa, M. & Terai,H. (2007). Architecture for high-speed Ant Colony Optimization, *Proceedings of IEEE International Conference on Information Reuse and Integration*, pp.95-100.



Greedy Algorithms
Edited by Witold Bednorz

ISBN 978-953-7619-27-5
Hard cover, 586 pages
Publisher InTech
Published online 01, November, 2008
Published in print edition November, 2008

Each chapter comprises a separate study on some optimization problem giving both an introductory look into the theory the problem comes from and some new developments invented by author(s). Usually some elementary knowledge is assumed, yet all the required facts are quoted mostly in examples, remarks or theorems.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Masaya Yoshikawa (2008). Hardware-oriented Ant Colony Optimization Considering Intensification and Diversification, Greedy Algorithms, Witold Bednorz (Ed.), ISBN: 978-953-7619-27-5, InTech, Available from: http://www.intechopen.com/books/greedy_algorithms/hardware-oriented_ant_colony_optimization_considering_intensification_and_diversification

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821