American Journal of Software Engineering and Applications
2014; 3(2): 12-20
Published online June 30, 2014 (http://www.sciencepublishinggroup.com/j/ajsea)
doi: 10.11648/j.ajsea.20140302.11

Application methods of ant colony algorithm

Elnaz Shafigh Fard[1], Khalil Monfaredi[2], Mohammad H. Nadimi[1]

[1]Faculty of Computer Engineering, Najafabad branch, Islamic Azad University, Isfahan, Iran
[2]Engineering Faculty, Department of Electrical and Electronic Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran

Email addresses:
shafighfard@azaruniv.edu (E. S. Fard), khmonfaredi@azaruniv.ac.ir (K. Monfaredi), nadimi@iaun.ac.ir (M. H. Nadimi)

To cite this article:
Elnaz Shafigh Fard, Khalil Monfaredi, Mohammad H. Nadimi. Application Methods of Ant Colony Algorithm. *American Journal of Software Engineering and Applications.* Vol. 3, No. 2, 2014, pp. 12-20. doi: 10.11648/j.ajsea.20140302.11

Abstract: As one of the most prestigious and beneficial methods of artificial intelligence, ant colony takes the advantage of communal behavior of ants in nature for solving optimization problems in various fields. However, this useful algorithm requires extensive and repetitious computation, as a result, the processing duration of the present algorithm seems to be one of the most serious challenges about it. In order to solve optimization problems in which duration is very important, this paper attempts to review the previously applied methods and consider the advantages and the disadvantages of each method through highlighting the problems algorithm designers encounter.

Keywords: Ant Colony, Optimization, Process Duration, Artificial Intelligence, Nature

1. Introduction

Nowadays, mathematical and physical methods can easily resolve some issues; on the other hand, their analysis cannot be done via typical methods. Engineers encounter vast complications and alternative solutions in a way that pronouncing a way as the best alternative seems absolutely difficult. Thus, a bunch of new conceptions under the shelter of complementary computation have been proposed for being put into use in optimization process and high speed search. Ant colony is one of the mentioned methods which has been inspired by nature and functions in many fields such as vectoring peddlers, financial issues, management of affairs, designation of axes, etc.(C.Blum,A.Roli,2003). In this algorithm, ants can solve the problems related to optimization by their tendency of mutual collaboration, collusion, and self-renovation. In recent decades parallel applications for the elevation of algorithm performance have been focused. The major problem of these algorithms is that they are time consuming because some calculations are done repeatedly and investigation areas are vast. This can take long hours, even may last for days. Hence, the typical application of such optimized algorithm for most speed functions will be useless. In the following passage, ant colony algorithm and its function will be introduced. The second section will offer some software, hardware, and compound methods together with efficiencies and deficiencies. Possible problems and challenges of designing and applying ant colony have been discussed in the third section and are preceded by the conclusion.

1.1. Ant Colony

Ant colony is one of the most prestigious alternatives of artificial intelligence that was proposed for the first time by Dorrigo and Dicaroin1992. The intelligent element is a being that can perceive environment through sensors and can manipulate environment by operators. While walking, ants usually leave a kind of chemical substance behind themselves called pheromone. The mentioned substance gradually starts evaporating. Shortly after evaporation, ant footsteps remain on the ground. When ants decide on their directions, they usually choose a path containing maximum amount of pheromone or already the way for other ants to pass. In every single phase, each ant builds up a resolution independently and spontaneously based on the distribution probability by using an item which benefits pheromone and heuristic factors as the following formula.

$$pij = \frac{[\tau i,j]^{\alpha}[\eta i,j]^{\beta}}{\sum[\tau i,j]^{\alpha}[\eta i,j]^{\beta}} \qquad \text{Formula 1}$$

After finding a solution for each ant, others begin to update locally and after choosing the best direction, they attempt a universal updating as formula 2 and 3.

$$\tau ijnew=(1-\rho)\ \tau ij(old\ )+\rho\Delta\tau ij \qquad \text{Formula 2}$$

$$\Delta\tau ij=(n.\ [\![Lnn)]\!]\ (-1) \qquad \text{Formula 3}$$

$$\tau ij^{new}=(1-\rho)\ \tau ij^{(old)} \qquad \text{Formula 4}$$

Evaporation as formula 4 takes place in all routes because if it does not occur, the result will be divergent and gradually the best routes will lose their quality. The process is repeated until algorithm comes to a conclusion and termination in addition to the offering of the best solution to the applied program.

2. Proposed Methods

In this phase, parallel applications for this algorithm are introduced. They fall under three categories: software, hardware, and compound.

2.1. Software Methods

In the last twenty years, researches have been searching for the techniques to improve traditional method of ant colony and to increase its speed. By categorizing ant colony into groups and sending each group to the processors, the speed application for this beneficial serial algorithm has been optimized (F. Glover, G. Kochenberger, 2003). This application not only improves the speed of algorithm but also presents a new derivation which has positive impact on the results. The major component is usually between the minor particles and grand particles of algorithm application. The most classic suggestion in relation to the parallel method of application such as the applying on multi-processor s, graphic processors, and grade environments are good opportunities for parallel estimations for optimizing the ant colony results and reducing the time of application (Martin, 2011Pedemonte). Various methods have been proposed which can be divided in five groups: 1. Manorial system 2. Cellular 3. Independent run 4. Multi colony 5. Compounds.

2.1.1. Initial Works

In early 1990s, the ideas were focused on the parallel application of ant colony. The first proposal for the application by Dorrigo attempted to develop the quality of algorithm. After Dorrigo, the first application was conducted by minor particle technique (M. Bolondi, M. Bondaza, 1993). An ant was dedicated to a single processor; however, because the deliverance of information should have been done, load of information exchange was a sort of shortcoming and encountered criteria problem. Then, in hierarchical application, several ants were specified to a processor. After the information distribution in hierarchical form, the information exchange was initiated simultaneously. In this case, the problem of criteria is solved but there is still the barrier of exchange (M. Dorigo, G. Di Caro, 1999).

2.1.2. Manorial System

This method is famous for its simplicity and facility of its application. Micro-processors were used as the tenants that undertake the application solutions and local searches have sent the information to landlords for finding the best ant which proved to be better in speed and quality(E.Talbi, O.Roux, 2001). But in minor particles for the maximum burden, the speed is roughly reduced. The problem has been solved by using shared memories (P.Delisle, M.Gravel, 2005). Of course, the existing landlord in grand particles as a gate can always be raised and it requires more observation.

2.1.3. Cellular Method

Ant colony is located within the pheromone matrix. In this method, each ant is corresponded to a cell and the resolution act of each ant is gradually spread over the entire matrix. This method is not a parallel method. Just an investigation of parallelism has been presented which used distribution memory. That is the best among parallel methods.(M.Pedemonte, H.Cancela, 2010)

2.1.4. Independent Run Method

In this method, several micro-processors focus on a colony and several serial ants run independently in each micro-processor. Different models of it have been proposed for application that the basis for independent run is a colony among many processors in a way that MMAS runs on GPU. Each thread is allocated to an ant and initialing CPU and main algorithm on graphic CPU was conducted on each thread before entering to the critical area that had been blocked so that the former thread finish its task. Specially, in some methods for lack of communication of processors with each other the speed was high. Nevertheless, the existing critical area in this field deserves further study. The model is one of the parallel methods in which several colonies collaborate and their collaboration in a form of ace crone has been illustrated. Most applications have utilized shared intelligence in the platform clustering. Recent studies have aimed to create a balance between ants colony while information transferring. Such an application will be used in greater scale and yield better results in comparison with series. There is something wrong with this algorithm; the expenses of communication cannot be ignored (H.Bai, D.Ouyang, 2009).

2.1.5. Multi colony

This method is one of the parallel methods in which several colonies collaborate. And their collaboration in a form of ace crone has been elaborated in the studies. Most applications have utilized shared intelligence in the platform clustering. Recent studies have aimed to create a balance between ants colony while data transferring. Such an application will be useful in greater scale and will yield better results in comparison with series (R. Michel, M. Middendorf, 1998).

2.1.6. Discussion and Comparison of the Applied Methods

The overview of mentioned applications shows that the methods of grand particles manorial and multi-colony are more reliable. The multi-colony methods have a degree of flexibility which allows them to partake in several parallel cases without wasting their functionality. Due to their

flexibility and measurability, they can be conducted in grand particles. This method has the capacity of hardware application. The functionality of the minor and major partition of manorial method stem from the value and frequency the information present in each tenant. Generally, the yielded results indicate that grand partition methods are better than minor ones but when the number of tenants increase, excessive communication will attack landlord and conversions will occur in route. The researchers have suggested the solution of ace crone. The minor scale manorial method was highly proposed so that the application of minor scale method in Graphic processors using shared memory saved in pheromone was proposed. However, the reduction of communication between central processor and GPU must be considered. Parallel cellular method and ace crone have the highest speed and performance.

2.2. Hardware Methods

The problem of this fiscal algorithm will not be solved even if it is installed as software or a powerful microprocessor because the presence of an operator and fetch of hardware will consume time of decoding and executing.

2.2.1. The Hardware Models of Multi colony Algorithm

The hardware methods of multi-colony algorithms the hardware application has high speed in comparison with software. There is no limitation of running the main ant colony algorithm for the flexibility of software but in case of hardware, it is challenging to apply hardware method for the following reasons:

1) For the amount of pheromone and the produced random quantity need to be illustrated in decimal form regarding application limits on logic arrays, there is a problem of designation.
2) Evaporation and heuristic factor that require multiplication are not supportable by most fpga arrays.
3) For choosing the next city based on the distribution probability factor. Total sum of the previously left factors should be estimated and for each pheromone matrix a circuit should be provided. If the number of ns in the problem sours up, there will be the limitations of time and space appearing on the array surface (B. Scheuerman, K. So, M. Guntsch, M. Middendorf and H. Schmeck, 2004). Large decimals and repetitions in addition to divisions, the ant colony should be changed in a way that the result is not influenced by it. Hence, in some algorithms, the heuristic factors in choosing next city are ignored instead of decimals. Definite numbers have been used for powered amounts α andβ are suppositions and shift will function and will be replaced by multiplication. The entire functions resulting in decimals are limited.

2.2.2. Reconfigurable Mesh Method

It is a standard method for models of processors capable of recurrent configurations including processor elements in which they are located in grades of K*N. Every single element has got four parts in order to communicate with its neighbors. Elements utilize their ports for reading and writing. In case several elements initiate writing stimulations, a logical ok is formed among the information. Each element recognizes the position of row and columns containing few registers (R. Vaidyanathan, L. Trahan, 2004). A mesh with two dimensions consisting minor particles is considered for connection organization for applying hardware algorithm of ant colony. The pheromone chart in this N*N is located as figure 1.

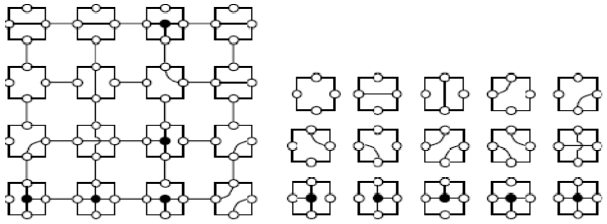

Figure 1. *16 mesh models and communication processor elements*

On the other hand, any processor element knows whether the existing item in the column has been selected or not. Second ant starts selecting his target item followed by first ant considering that there is only one row which makes a distance between them. There are two exceptions in this method that it does not obey the ant colony regulation:

1. Coming to the end of the row and selecting his solution for m-1/2, ant waits till it is checked for the position of solution among the previously selected best solutions then the full update is initiated. While other ant is on the selection route.
2. Instead of using pheromone item directly for selecting the next item, supermom and minimum for speed elevation is used in running algorithm.

In formula 5 t>0, h>l>0 is considered and for each item jεS the very formula is defined.

This algorithm is ideal for elevating the speed of ant colony algorithm application but comparison and pheromone update may be applied several times.

2.2.3. Population Method Based on Field Program Array

In this method, minimum space is allocated to designation and application related to the vivacity of previously conducted researches on the field. It has high speed in comparison with software method. Ants scatter into pheromone chart instead of an n*n matrix which runs repeatedly. A population chart is modeled on the pheromone chart that is about 1-8.This chart includes the population of the best routes gone in the former repetitions. The ants use this newly created chart for direction choice. By an n*k switch, n is the number of the problem items and k the number of the best routes. Each ant is allocated to a row by choosing an item in that row; the ant goes to the

next row. This process continues till the number of rows reach to n, and then a new cycle begins. As illustrated in figure 2, this method includes 3 modules. The first module is the population module which is located between the two items of i and j belonging to pheromone factor. That is known as qij. The second module, which is the solution module, is divided into 3 sections: 1. S- array 2.match buffer 3.Selector. The third module is special for evaluation (B.Scheuerman, K. So, M. Guntsch, M. Middendorf and H. Schmeck, 2004).

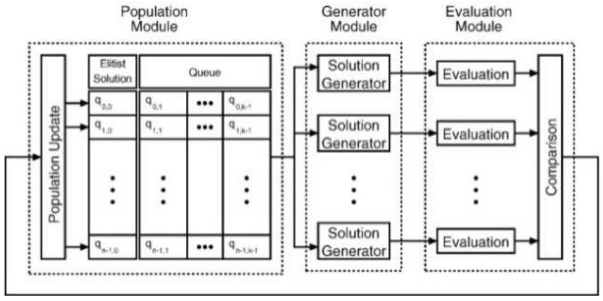

Figure 2. *The population based method*

Here, the population module bears the responsibility of providing the producer module with an n*k matrix presenting the pheromone amount. Eventually, it places the best solution in zero column and the remaining solutions in population matrix as the first entrance, first exeunt. In figure 3, the overall flow chart for this method has been demonstrated.

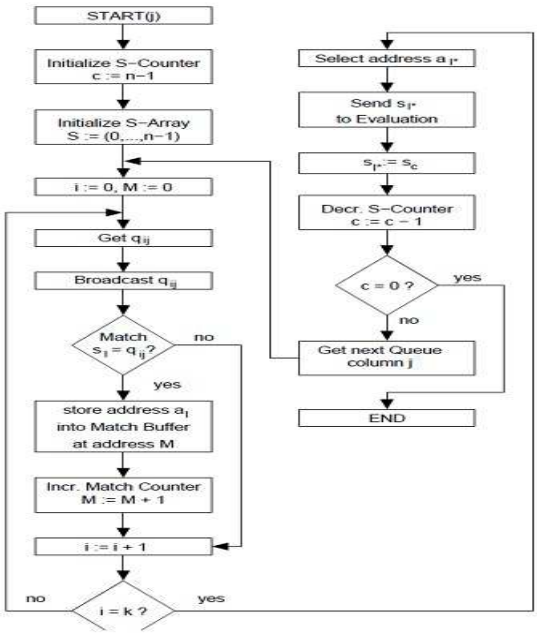

Figure 3. *Flowchart of solution*

Generally in the flowchart of figure 3, after the passage of each ant from each queue and the comparison of available items in queue i of pheromone matrix with s components, if they were the same, the address for saving in match buffer in case of correspondence of match, the number production duration will be randomly produced and the item is selected. This continues till all the items of s are selected and a solution is given. S-Array includes all the items of the problem. Every single component of pheromone matrix is compared to s- array and at the end of each queue. The matched items are saved in match buffer even if the saving process occurs several times. The quantity of the matched item M and c=s-l are sent to the selector according to R=c+MΔ is identified between 0 to R for producing random numbers. Selection of an item from a match buffer and one item out of an s- array is previously selected. Selecting an item out of an array, it shifts to right and a flag special to the selected items are disabled. This process goes on till all the s members are selected and the solution to the selection of ants is terminated.

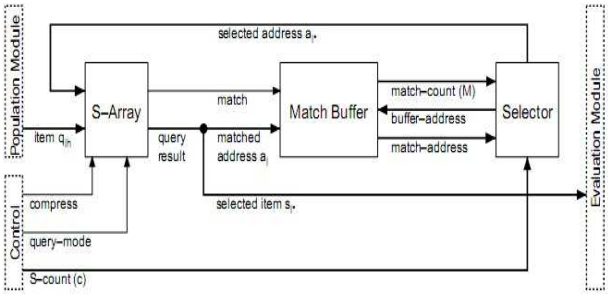

Figure 4. *Diagram blocks producing solution.*

The route of the next city has been illustrated in figure 5. If r-c causes overflow and r›c is a collection of match buffer or the previously selected address that has been omitted from match buffer.

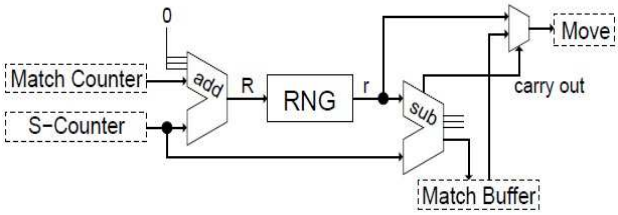

Figure 5. *Selector block*

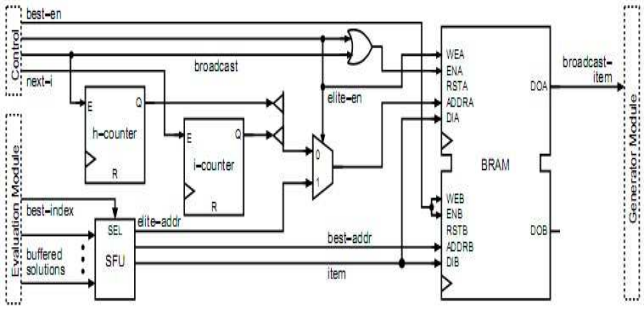

Figure 6. *Population module*

After all the ants found their solutions, in evaluation module, the best solution is found based on comparison. In BRAM, that pheromone matrix is inside it, via port A the best solution is written in j=o queue. The consideration is

that the best solution is not better that the former one when (FIFO) is saved via port B. As is shown in the picture, the two counters of column and queue are sent to the solution module via port A so that a new cycle of solution is started by ants.

As previously mentioned, the size of this matrix is fixed and the oldest solution is deleted from matrix when a new solution is added to it. (In case there is no best solution.) It is noticeable that the pheromone matrix for k times has not been used. The randomly produced quantities will result in the selection of an address including previously selected items that are not in s array. This wastes time. And heuristic factors in selecting the new next city are not interfered.

2.2.4. Duplicators on Field Program Array

The parallel running of hardware application is done with chief purpose of speed elevation. Below are the two types of parallel applications.

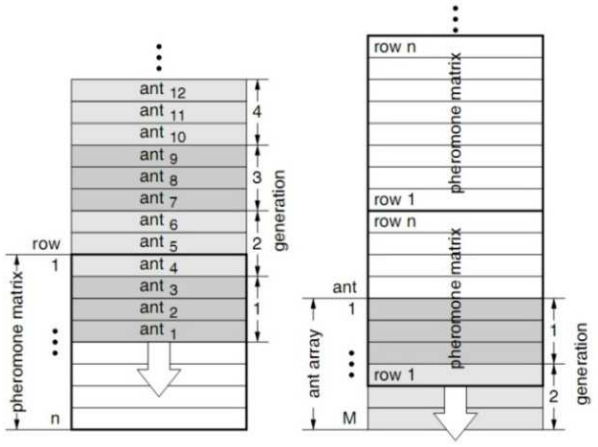

Figure 7. *Parallel method: right is pheromone pipe route, left is the ant route*

In this method pheromone matrix is projected on the array and ants enter linearly into pheromone matrix. Each ant selects an item in each route together with other ants. They intend to select a solution sequentially. However, the ants were not active in series. In these streams, ants are projected on the array stably and pheromone passes the location like a stream. In counter application, the search for algorithm parallel in all phases such as building a solution, evaluation, comparison and upgrading are increased.

Solution setup:

For setting up a solution each ant has to follow these steps:

- ✓ Selection probability calculation: In this method the distribution probability of heuristic items and unselected of pheromones are used instead of calculating the probability of single item.
- ✓ Random number production: A proper number [0: B_i-1] $r_i\varepsilon$ which is the subprime for B_i=max hεs p r_ih, i.e. the maximum amount resulted in the sum of multiplication. O (1) is produced as the random number.
- ✓ Selection: After producing random quantity for each deselected cell, the statement $pr_{i,\ j-1}<r_i<pr_{i,j}$ is calculated. If the mentioned statement in the cell is not compatible, the cell item will be selected and be deleted from s.
- ✓ Evaluation
- ✓ Comparison
- ✓ Pheromone update

2.2.5. Projection of Proposed Method on Fpga

For each cell a circuit is designed in which each ant and the item members of s enters pheromone matrix projected on fpga decadently.

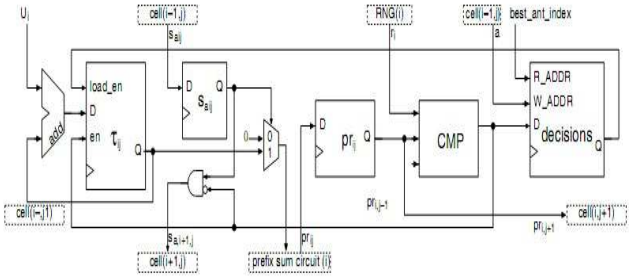

Figure 8. *Cell circuit (i and j)*

As is shown in the figure, if each member of si group is not selected, it is 1. If they have been selected earlier, it is 0. If the amount is 1, it participates in summing the multiplication factor and is compared with random numbers and sum of amount factors resulted from cell. If the case is as mentioned, the cell is zero and it is transferred to the next row in order not to be selected by ant in the next row. For each ant a design maker memory is selected in a form of distribution in fpga, selected items of each ant are saved. The previous ant should wait for the termination of solution selection process of the preceding ant. Then, it should begin to orbit pheromone matrix. This also consumes time and heuristic factor is not interfered in the next city selection. It is theoretically expressed. The third deficiency is because of the distributed memory utilization in cells. Space limitation and problem in items and bit quantity will be encountered (B. Scheuerman, K. So, M. Guntsch, M. Middendorf and H. Schmeck, 2004).

2.2.6. Method Based on CMOS Technology

In this method, the technology of serial transistor cmos has been used (K. Gheysari, A. Khoei, B. Mashroufi, 2011). Certainly, the diverse distance between cities has been influential, as well. The designation consists of five sections. The structure of each cell is illustrated in figure 9. Similar to the previous methods, ants start searching from the initial line of pheromone matrix. Their alternative city declines the amount from one to zero. In the memory set in each cell saves it in 6 bits and transfers to the next row.

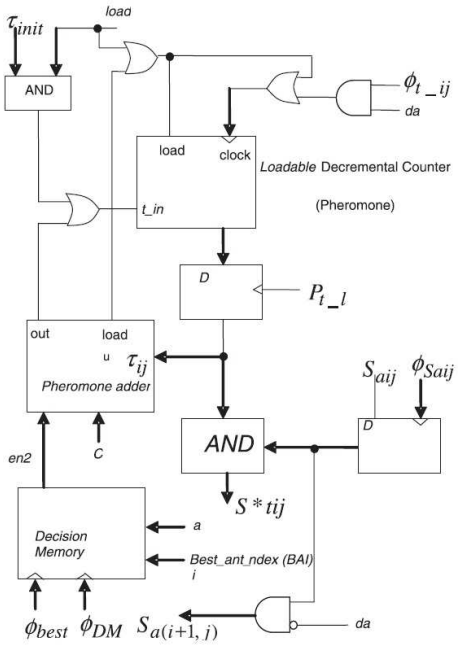

Figure 9. *Internal structure of a cell*

It is obvious which cell has been chosen by which ant. Decision memory, loadable decremented counter, and pheromone adder are considered as the main components of cells.

2. Choosing the next city:

This circuit determines the next city based on pheromone and heuristic amounts present in cells. A cell with high pheromone and heuristic amount will be chosen. As is shown in figure 10, this block includes 4 units: the first unit includes a digital multiplier which multiplies each value. The second unit containing DAC receives an output of multiplier and converts it to analog. For each row, there is a producer of random numbers which is an indirect application of roulette wheel can be seen. For an instance, if $I_{in1}<I_{ri}<I_{in2}$., then, the second node will deserve optional condition.

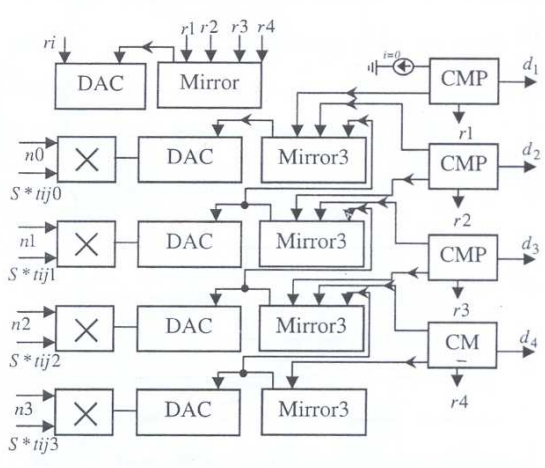

Figure 10. *Next city selection circuit*

The results of additions and multiplications of pheromone in deselected heuristic items are used in choosing the next city. The next city selected by random production of numbers existing in every row of pheromone matrix and by roulette wheel. This method is a function based on the integrated circuits. It has lower consumption of power considering cmos technology and has high speed in comparison with the presented models. However, in this method, each ant has to wait until the preceding process of route choice is finished. This is caused by the present algorithm and the zero-one position of transistors. Then it starts passing pheromone matrix.

2.2.7. Hardware-Software Compound Method

The designation is done on the arrays in a way that C software and hardware arrays have been planned with repartition in Verilog language. The designed framework of algorithm, as shown in figures 11 and12, is composed of two major sections. Hence, choosing the best route is done by software and is applied in C language on the hidden processors NIOS II. Other algorithm calculations are applied by hardware on array logic which can be reprogrammed. The proposed solution brings a kind of negotiation between hardware speed designation and the flexibility of software planning (S-Anli, M.Hao, C-Wei, Y-Hong, 2012).

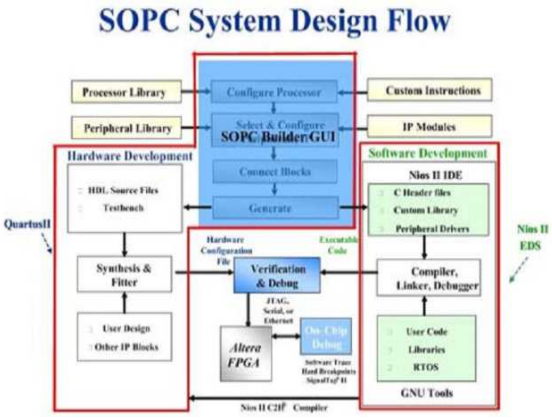

Figure 11. *Software/hardware interaction on array*

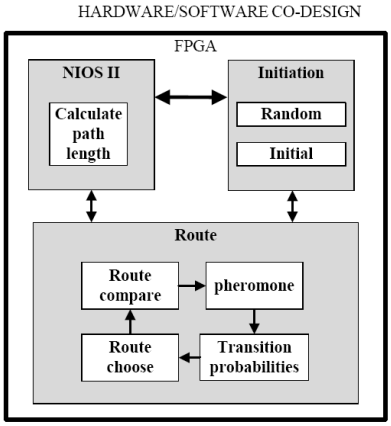

Figure 12. *Ant colony circuit chart*

2.2.8. Hardware Design Based on Deliberate Pheromone Temperation (M. Yoshikawa, H. Terari, 2008)

In this method, the management of ant colony algorithm varieties is important. As the major idea of the method, the variables are focused on the produced pheromone. The pheromone amount is defined according to the three limits; upper, benchmark and lower. According to formulas 2 and 3 for updating pheromone, some decimal calculations should be done.

The calculations for hardware pieces as fpga are not compatible because of their structures. The statement in formula 5 suggested for local update and not using decimals. General update is done like the defined variables in statement 6.

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

$$\beta = \begin{cases} 8 & if \;\; 8 < \tau < std \\ 4 & if \;\; std < \tau < max \\ 0 & if \;\; 8 = \tau \end{cases} \qquad \text{Formula 5}$$

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

$$\varepsilon = \begin{cases} 8 & if \;\; \tau < std \\ 0 & otherwise \end{cases} \qquad \text{Formula 6}$$

2.2.8.1. Search and Node Selection

Like update method, search method is set up in a way that it does not need decimal calculation. The chart information indicate that for pheromone variable 100 distance is dedicated for heuristic factors per 8 block variations. All the application is done on LUT. The quantity of increase, pheromone, heuristic factors, and LUT amount will rise.

Table 1.

$\tau(i,j)$ \ $\eta(i,j)$	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	…
.	.	.	.	.	…
.	.	.	.	.	

2.2.8.2. Design Methodology

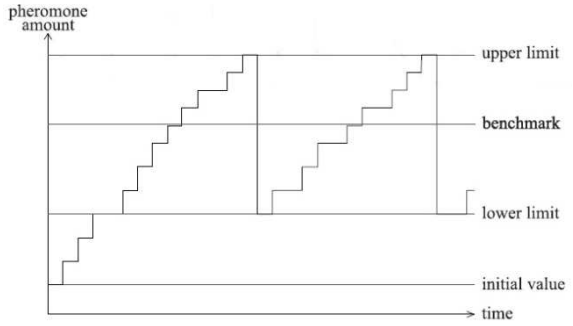

Figure 13. *An example of pheromone amount*

Every ant makes a different route like the standard algorithm of ant colony. Each search is done parallel to other processes (Figure 13). Several process blocks including management, general update memory composed of composite 6units and processors unit containing processor elements for parallel search have been illustrated in 14.

Selected city unit is constructed of sub-units and float point unit is used for processing decimals in a way that the function quality is the approximation of the real ant colony function (figure 14). The suggested evaluation method with its software equivalence is shown in table 2.

Table 2

	ACO	HAAC
Solution	Opt.	Opt.
Time to 100 tours	.68(s)	0.30(s)
Time to Opt.	6.9(s)	8.3(s)

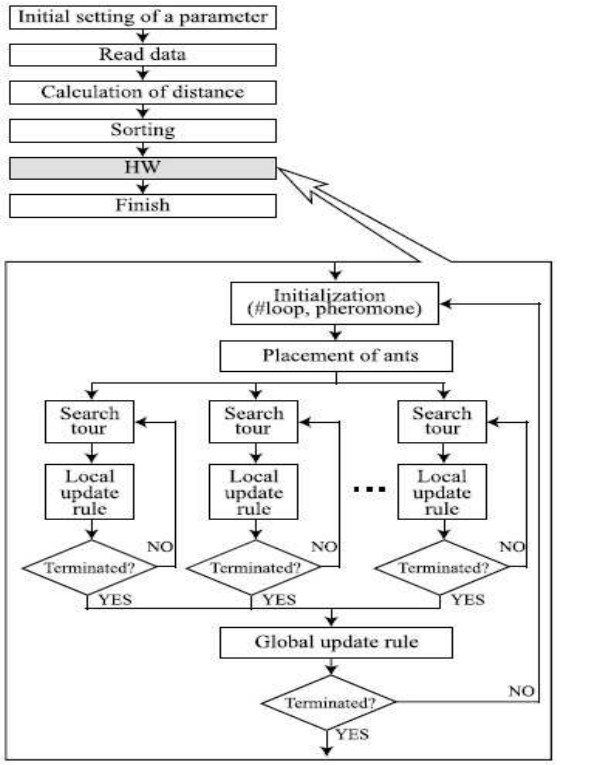

Figure 14. *Suggested flowchart*

2.2.9. Another Design for ACO Speed Elevation (Masaya Yoshikawa, Hidekazu Terai, 2007)

Ant colony algorithm has some shortcomings:

1. Processing occurs repeatedly and sequentially
2. Pheromone variable value is very little.
3. All the ants should share pheromone variables for transition and update. Due to the repeated processing difficulty, parallel method is perfect. block diagram of this method is shown in figure 16.

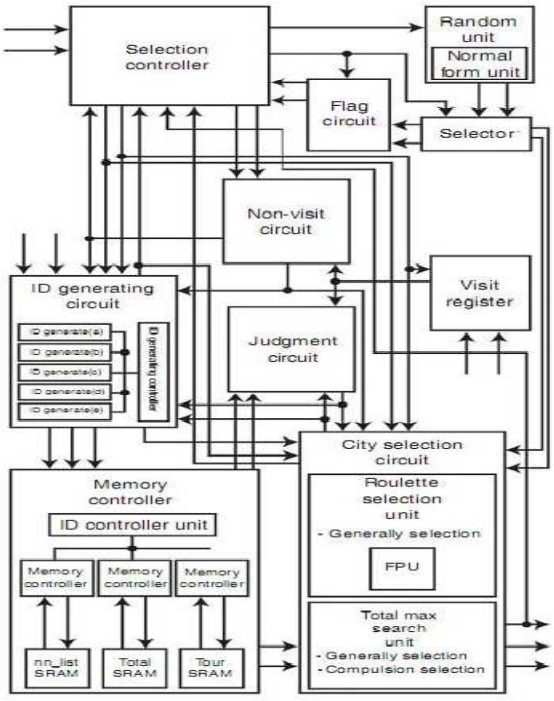

Figure 15. *City Selection Diagram*

Series of ids saved in SRAM memory are used for managing pheromone. They have the similar structure as the below.

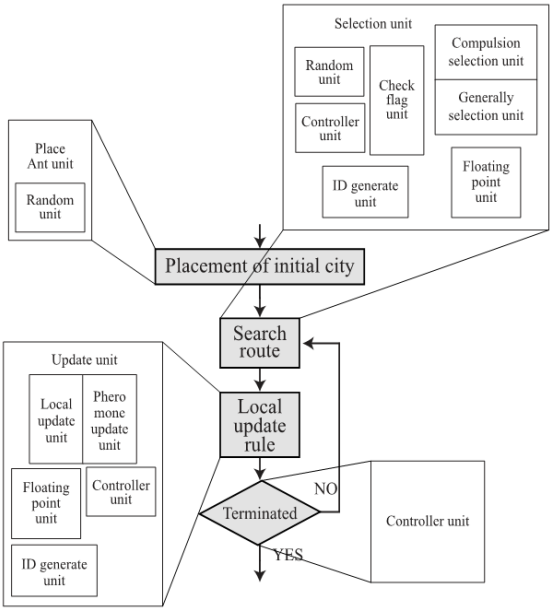

Figure 16. *The relation of processing flow chart with the circuit*

As figure 16, the function of an ant using units is shown in the beginning; each ant selects randomly a node or station. After choosing the next station, each ant for each city drags with 24 bit pheromone from memory and the remaining bits which control selection and deselect of the target city uses it. Immediately after choosing city starts local updating and it does it for all the cities so that one repetition suggests one solution. In chart 3, methods of hardware are compared in a summarized form.

Table 3. *Hardware Methods Compared*

function	Weak point	Strength	Design	Method
Continuous Function in Limited Time	Communication Expenses	Capable of Scaling	A PE System	Mesh
Route Finding	Maximum Comparison- not Capable of Scaling	Limited Space Exploration	FPGA Array	Based on Population
Route Finding	Ant Limitation Caused by Memory of Cells	Pheromone Matrix Exploration- Design Simplicity	FPGA Array	Duplicator
For Noiseless Places	Serial Ant Functioning- not Capable of Scaling – Noise Absorption	Lower Power Limit- Design Simplicity	Combination of Transistor and Comparer	CMOS
Programs Needing Exact Process	Lower Speed Compared to Hardware Type-Expensive	Flexibility-Algorithm Match	Fpga Array and NIOS II Processor	Software-Hardware
Improper Processing	Complex Design	Simple Operations	Combination of fpga and Memory	HAAC
Proper Calculations	Complexity and Difficult Management ID	Fpu usage and Proper Results	Combination of SRAM-FPGA	Hardware Design

3. Challenges and Problems

The reviewed works hardware designs are incomparable with hardware application. Because of the speed running continuous programs they will be deficient. In hardware design we encounter problems. Hardware designation can be divided in two sections: 1. Real designation. 2. Flexible systems.

1. Meaning the approximation of the planned method and the real ant colony is apparent, flexible systems react to the errors and change, and it is reliable to carry errors.

2. Online radiance: One of the challenges in hardware design is that despite the progress of ant colony algorithm optimization, no satisfying report pertinent to the online radiance is given. Usually, it has been offline; the speed process is very low. This is regarded as the biggest problem in functional programs running by ant colony in industry. If the evaluation of best ants can be collected in the repeated forms and best ants can enter real world, the result of application related to time will be better.

3. Power and Strength: while an error occurs, it is essential to discover the error and resolve it by restarting the program so that no time inconsistency shall happen. But is the error discovering estimated and applied properly? Many reports on discovering error designation of hardware have not been delivered. This is the biggest challenge.

4. Critical Issues: It is important to generalize designation to any environment; whether the designation operates in efficient systems or can optimize every system under possible conditions. The more complex issues in optimization, the more critical issues are generated. In some cases algorithm does not answer properly to these issues. There are 2 possible solutions to these problems:

1. Optimizing hardware designations for speed increasing
2. Optimizing the ant colony algorithm

5. Theoretical Analysis

Theoretical analysis yield a comprehensive function of optimization algorithm but for those who want hardware application it is very proper. A basic theory is that before applying in larger scale, a theory is proposed. Since unlimited time and space is needed is needed for converging algorithm optimization with satisfying results.

4. Conclusion

This paper had investigated time problem in applying the ant colony algorithm. Parallel application of hardware methods together with the advantages and disadvantages were discussed. As a result, the compound method was introduced as the beneficial one according to flexibility and speed. Finally, the changes for hardware designers were pointed out. The investigator hopes that in the future, the hardware flexibility method as a novel and functional method will solve speed challenging problems regarding the application of ant colony algorithm.

References

[1] B. Scheuermann, S. Janson, M. Middendorf, Hardware-oriented ant colony optimization, Journal of Systems Architecture 53 (7) (2007) 386–402.

[2] C. Blum, A. Roli, Metaheuristics in combinatorial optimization: overview and conceptual comparison, ACM Computing Surveys 35 (3) (2003) 268–308

[3] Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science 1498 (1998) 692–701

[4] D. Merkle, M. Middendorf, Fast ant colony optimization on runtime reconfigurable processor arrays, Genetic Programming and Evolvable Machines 3 (4) (2002) 345–361

[5] E. Talbi, O. Roux, C. Fonlupt, D. Robillard, Parallel ant colonies for the quadratic assignement problem, Future Generation Computer Systems 17 (4) (2001) 441–449

[6] F. Glover, G. Kochenberger (Eds.), Handbook of Metaheuristics, International Series in Operations Research & Management Science, 57, Springer, 2003.

[7] H. Bai, D. OuYang, X. Li, L. He, H. Yu, Max-min ant system on gpu with cuda, in:Proceedings of the 2009 Fourth International Conference on Innovative Computing,Information and Control, IEEE Computer Society, 2009, pp. 801–804 .

[8] H.Duan ,Yaxiang.Yu , JieZou and Xing Feng .Ant colony optimization-based bio-inspired hardware.

[9] K.Gheysari,A.Khoei,B.Mashoufi High speed ant colony optimization CMOS Chip .

[10] M. Bolondi, M. Bondaza, Parallelizzazione di unalgoritmo per la risoluzione del problema del commessoviaggiatore, Master's thesis, Politecnico di Milano, Italy, 1993

[11] M. Dorigo, G. Di Caro, L. Gambardella, Ant algorithms for discrete optimization, Artificial Life 5 (2) (1999) 137–172

[12] M. Dorigo, Parallel ant system: an experimental study, unpublished manuscript, Cited by [41], 1993

[13] M. Pedemonte, H. Cancela, A cellular ant colony optimisation for the generalized Steiner problem, International Journal of Innovative Computing and Applications 2 (3) (2010) 188–201.

[14] M. Pedemonte, S. Nesmachnow, H.Cancela Survey on parallel ant colony optimization .Applied Soft computing hournal.(2011.)

[15] P. Delisle, M. Gravel, M. Krajecki, C. Gagné, W. Price, Comparing parallelization of an ACO: message passing vs. shared memory, in: Proceedings of the 2nd International Workshop on Hybrid Metaheuristics, Lecture Notes in Computer Science vol. 3636 (2005) 1–11

[16] R. Michel, M. Middendorf, An island model based ant system with lookahead for the shortest supersequence problem, in: Proceedings of the 5th International.

[17] R.Vaidyanathan and J.L.Trahan :Dynamic Reconfiguration :Architectures and Algorithms .Kluwer,(2004)

[18] S.Li,M.Hao Yang ,Chung-Wei WENG,Yi –Hong Chen Ant Colony Optimization design and its FPGA implementation

[19] Yoshikawa ,M and Terai,H 2007 : Architecture for high – speed ant colony optimization .oroceedings of IEEE International Conference on information Reuse and integration ,lasVegas,NV, 1-5 .

[20] Yoshikawa,M and Terai,H 2008:Hardware-oriented ant colony optimization considering intensification and diversification in:Bednorz, W.editor. Advances in greedy algorithms,I-Tech,359-68.