Proceedings of the 2007 IEEE Workshop on
Evolvable and Adaptive Hardware (WEAH 2007)

Progresses and Challenges of Ant Colony Optimization-Based Evolvable Hardware

Haibin Duan, *Member, IEEE*
School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100083, China; hbduan@buaa.edu.cn

Xiufen Yu
Center for Space Science and Applied Research, Chinese Academy of Sciences Beijing, 100080, China; yxf@nmrs.ac.cn

***Abstract* - Evolvable hardware(EHW) refers to hardware that can change its architecture and behavior dynamically and autonomously by interacting with its environment, and ant colony optimization is a meta-heuristic algorithm for the approximate solution of combinatorial optimization problems that has been inspired by the foraging behavior of real ant colonies. In this paper, we take a broad view on the progresses of ant colony optimization-based EHW, which include ant colony optimization-based random number generator, FPGA implementation, digital circuits, digital IIR filters and so on. Some important issues of the challenges of ant colony optimization-based EHW are also presented, and online realization, robustness, generalization, disaster problem are four key challenging issues of ant colony optimization-based EHW in the future.**

I. INTRODUCTION

Evolvable hardware(EHW) refers to hardware that can change its architecture and behavior dynamically and autonomously by interacting with its environment. At present, almost all EHW use an evolutionary algorithm (EA) as their main adaptive mechanism. It was H. D. Garis who made the first move to investigate the design of evolving circuits. In his paper [1], H. D. Garis suggested the establishment of a new field of research called Evolvable Hardware (EHW). At about the same time, the first work in evolutionary design of digital circuits was carried out by S. J. Louis[2]. A complete review and taxonomy of the EHW field is described in [3] by X. Yao and T. Higuichi.

Ant colony optimization is a meta-heuristic algorithm for the approximate solution of combinatorial optimization problems that has been inspired by the foraging behavior of real ant colonies [4, 5]. In ant colony optimization, the computational resources are allocated to a set of relatively simple agents that exploit a form of indirect communication mediated by the environment to construct solutions to the finding the shortest path from ant nest to a considered problem. The natural metaphor on which ant colony optimization is based is that of ant colonies. Real ants are capable of finding the shortest path from a food source to their nest, without using visual cues by exploiting pheromone information. While walking, ants deposit pheromone on the ground, and follow, in probability, pheromone previously deposited by other ants [6]. A way ants exploit pheromone to find a shortest path between two points is shown in Fig. 1.

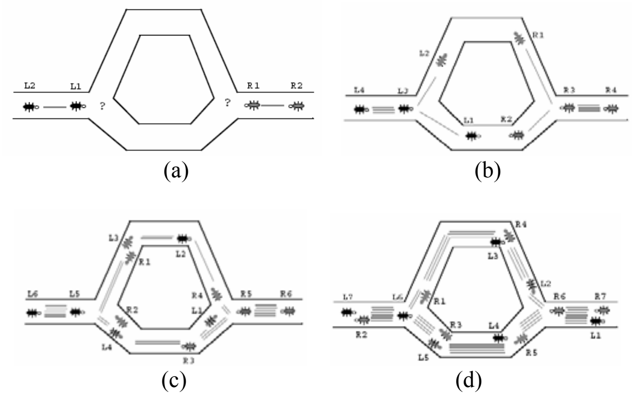

Fig. 1. How real ants find a shortest path

The above behavior of real ants has inspired ant colony optimization, an algorithm in which a set of artificial ants cooperate to the solution of a problem by exchanging information via pheromone deposited on graph edges. This process is thus characterized by a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path. With the above positive feedback mechanism, all ants will choose the shorter path in the end. By now, ant colony algorithm has been applied to combinatorial optimization problems such as the TSP(traveling salesman problem), JSP(job-shop problem), QAP(quadratic assignment problem), VRP(vehicle routing problem), GCP(graph coloring problem), and so on [7].

In this paper, we take a broad view on the progresses of ant colony optimization-based EHW and address some important issues of the challenges of ant colony optimization-based EHW in more detail in later sections. The remainder of this paper is organized as follows. Section II describes the overall new approaches in ant colony optimization-based EHW. Subsequently, a detailed analysis of challenges lying in ant colony optimization-based EHW are presented in Section III. Our concluding remarks and future work are contained in Section IV.

II. PROGRESSES OF ANT COLONY OPTIMIZATION-BASED EHW

A. Ant Colony Optimization-Based Random Number Generator

A random number generator is a computational or physical device designed to generate a sequence of numbers or symbol. J. C. Isaacs and his colleagues proposed a new random number generator scheme[8], which employs simulated ant colonies evolved by a genetic algorithm to produce very high-quality random bit generators in evolvable hardware(see Fig. 2).

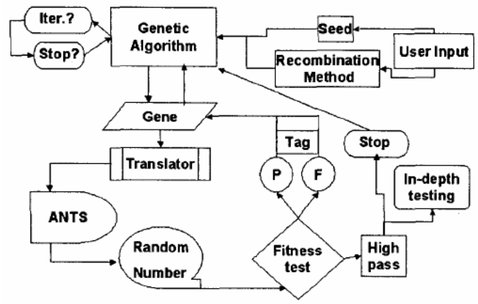

Fig. 2 Ant colony genetic algorithm schematic

Their off-line testing indicates that the evolved generators of schemata-based family routinely pass all tests in the Diehard battery, and those are superior to many known, non-evolvable gengerators. However, these are not implementable in hardware at that time, and there is a great need to develop the proposed method-with an eye towards optimizing the compactness of the hardware implementation, specifically, minimizing gate count and maximizing switching frequency.

B. Ant Colony Optimization-Based FPGA Implementation

B. Scheuermann and his colleagues took the single machine total tardiness problem (SMTTP) as a typical example, presented a hardware implementation of population-based ant colony optimization (P-ACO) on field-programmable gate arrays (FPGAs) in 2004[9]. In that paper, they describe the P-ACO algorithm and present a circuit architecture that facilitates efficient FPGA implementations.

FPGAs are used for a wide range of applications, e.g. network communication [10], video communication and processing [11] and cryptographic applications [12]. It has been shown that FPGAs are suitable for the implementation of soft computing techniques like Neural Networks [13, 14] and Genetic Algorithms [15-17]. They show that ant colony optimization can also be implemented on FPGAs, leading to significant speedups in runtime compared to implementations in software on sequential machines.

When mapping basic ant colony optimization to hardware, there are several restrictions set by the target architecture. These restrictions concern the number, the type, and the distribution of available computational, memory and I/O resources [9, 18]. These restrictions can be listed as follows:

(1) Pheromone values and the random numbers used require a floating point representation. Such a representation does not lend itself to a fine-grained programmable logic implementation.

(2) Evaporation, and the integration of heuristic information, requires multiplication operations. However, the computational resources available on an FPGA do not efficiently support the implementation of multiplication circuits. Only a few FPGAs have dedicated multiplication blocks and these are restricted in size.

(3) To make a selection according to the probability distribution, It is necessary to calculate prefix sums of the products in the numerator over the as yet unchosen items in the selection set. Therefore, the required space and time complexity is prohibitive even for the comparatively large present day programmable gate arrays.

For the above reasons, an alternative P-ACO approach was introduced in the paper [9, 19, 20]. At high level, the mapping of the P-ACO algorithm into the corresponding FPGA design is straightforward (see Fig. 3) and consists of three main modules: Population, Generator and Evaluation modules. The Population module contains the population matrix $Q=[q_{ij}]_{n\times k}$ comprising the elitist solution in column $j=0$ and the FIFO-queue in columns $j\in\{1, . . . , k\text{-}1\}$. For SMTTP, each item q_{ij} is the number of a job to be scheduled. The Population module is responsible for broadcasting items q_{ih} $(h\in\{0, . . . , k\text{-}1\})$ in the ith row of the population matrix to the Generator module. Furthermore, at the end of the current iteration it receives the best solution from the Evaluation module, which is then inserted into the queue. The Generator module holds m Solution Generators working concurrently, one Solution Generator per ant. The solutions are transferred from there to m parallel Evaluation Blocks in the Evaluation module. It is also possible to have less than m Solution Generators and Evaluation circuits. The evaluation results of these m solutions are collected in a Comparison block, which determines the best solution of the current iteration. This best solution also becomes the new elitist solution, if it is better than the current elitist solution.

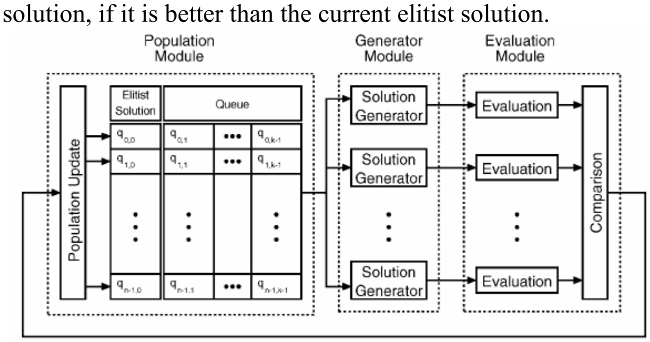

Fig. 3. P-ACO design with population, generator and evaluation modules.

C. Ant Colony Optimization-Based Digital Circuits

Design of digital circuits is a process to assemble a collection of components to realize a specified function using a target technology. Typically, the behavior of each component of the designed circuit is well known. The difficulty lies in predicting how an assembly of such components will behave. Unfortunately, current design systems tend to depend on domain-specific knowledge, which is somewhat constrained both by the training and experience of the designer. On the other hand, nondeterministic iterative heuristics, with little domain knowledge, may allow us to define a search space, make some assumptions and use domain-independent operators for generating candidate solutions in the design space. Iterative heuristics have tendency to search for solutions in a much larger, and often richer, design space beyond the realms of the conventional techniques. It may therefore be possible to use them to obtain novel designs that are difficult to find using conventional methods.

In a recent development, much attention is given to the evolutionary design of digital circuits. Such effort has resulted in the development of digital circuits that range from a simple sequential adder structure to the more complex3-bit multiplier. Some of the recent work can be found in [23-25]. Unfortunately, majority of the published work attempts to obtain optimized circuits in terms of gate count only, and overlook other major issues such as delay and power consumption. M. Abd-El-Barr presented a multi objective evolutionary logic design based on ant colony optimization for digital circuits, and the goal is to find optimized circuits in terms of area, delay and power. They have shown that the proposed scheme is feasible and efficient.

D. Ant Colony Optimization-Based Digital IIR Filters

In order to transform and analyze signals that have been sampled from analogue sources, digital signal processing (DSP) algorithms are employed. After the cheap and powerful general-purpose computers and custom-designed DSP chips have been developed, DSP has found very significant applications in several engineering areas from communication, biomedical, and control to meteorology [26]. The advantages of DSP are based on the fact that the performance of the applied algorithm is always predictable. There is no dependence on the tolerances of electrical components as in analogue systems.

Any DSP algorithm or processor can be reasonably described as a digital filter. Digital filters can be broadly classified into two groups: recursive and non-recursive filters. The response of non-recursive, or finite impulse response (FIR) filters is dependent only upon present and previous values of the input signal. Recursive, or infinite impulse-response (IIR) filters, however, depend not only upon the input data but also upon one or more previous output values. The main advantage of a digital IIR filter is that it can provide a much better performance than the FIR filter having the same number of coefficients. However, there are some problems with the design of IIR filters. The fundamental problem is that they might have a multimodal error surface. A further problem is the possibility of the filter becoming unstable during the adaptation process. Although this second problem can be easily handled by limiting the parameter space, in order to avoid the first problem, a design method which can achieve the global minima in a multi-modal error surface is required. However, the conventional design methods based on gradient search can easily be stuck at local minima of error surface. Therefore, some researchers have attempted to develop the design methods based on modern global optimisation algorithms such as the simulated annealing (SA) algorithm[27] and genetic algorithm (GA) [28, 29].

N. Karaboga applied a touring ant colony optimization (TACO) algorithm to digital IIR filter design(see Fig. 4)[26].

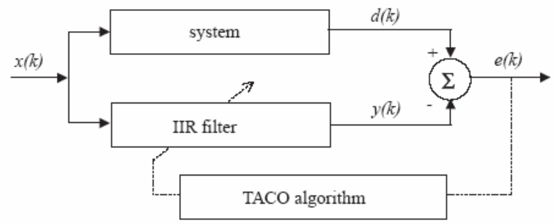

Fig. 4. Block diagram of system identification process using IIR filter

In Fig. 4, the parameters of the IIR filter are successively adjusted by the modified TACO algorithm until the error between the outputs of the filter and the system is minimized. Simulation studies show that the proposed method is accurate and has a fast convergence rate, and the results obtained demonstrate that the new method based on TACO can be efficiently used for digital IIR filter design.

In addition, there are many great progresses in other ant colony optimization-based EHWs: D. Merkle proposed a novel type of ant colony optimization, called HIGHLOW-P-Ant, which can be implemented in quasi-linear time (with respect to the total number of ants that search for a solution and the problem size) on a reconfigurable mesh with processors; B. A. B. Sarif presented a fuzzified and colony algorithm for the multi-objective optimization of logic circuits.

III. CHALLENGES OF ANT COLONY OPTIMIZATION-BASED EHW

Although adaptive ant colony optimization-based EHW has made great progresses in the last several years, there are some fundamental and interesting issues in ant colony optimization-based EHW that are worth probing further. Several of the typical challenges are listed as follows.

A. Online Realization

Online realization requires EHW learning to be incremental and responsive [3], and online realization of ant colony optimization-based EHW is the key issue of this research area.

In spite of the great progress of ant colony optimization-based EHW, almost no satisfied work has been reported on online realization of ant colony optimization-based EHW. Most of the achieved results are got offline, in which the proposed ant colony optimization scheme happens during the learning phase of EHW. A prominent disadvantage of offline ant colony optimization-based EHW is that the evolutionary speed is slow, and this limit the wide application of ant colony optimization in industry.

On the other hand, evaluating an ant colony optimization-based EHW in a real physical environment could cause severe damages to the ant colony optimization-based EHW or the physical environment. This potential risk restricts possible applications of ant colony optimization-based EHW in domains in which evaluating ant colony optimization-based EHW in a real physical environment is impractical and an accurate simulation model of the physical environment is difficult to obtain.

A possible way to solve this problem is to develop a knowledge-based adaptive ant colony optimization-based EHW, where constraints and knowledge about the uncertain environment in which the ant colony optimization-based EHW will be evaluated are incorporated into the iteration process as its front end, such that only most of the elitist "ants" could be passed to the real physical environment.

B. Robustness

Robustness is a key issue for the reliable ant colony optimization-based EHW. When there is failure, the failure detection scheme of ant colony optimization-based EHW should immediately detect the error and initiates a reprogramming that correctly programs in the hardware. But whether the failure detection scheme succeeds in each failure is an open problem by now. In fact, many reported ant colony optimization-based EHW applications are not satisfied in detecting the faults correctly each time.

Therefore, the robustness of ant colony optimization-based EHW may be achieved through various means of fault detection and repair or through fault tolerance, and the robustness of ant colony optimization-based EHW quality should be further investigated in the future.

C. Generalization

Generalization is also a key issue for any learning or adaptive systems [3], including ant colony optimization-based EHW. Evaluating the ant colony optimization-based EHW's generalization can be a difficult task due to different implementations. This difficulty is closely related to that of evaluating the generalization ability of ant colony optimization system in general. It is not uncommon to read papers that only report a good system evolved at a certain number of generations. It is unclear, however, whether such a good system is the result of one particular run or the average of multiple runs.

However, studies on this topic are relatively few in the area of ant colony optimization-based EHW. Some experiments on ant colony optimization-based EHW did not address the issue since the same training and testing data set was used, and statistical analysis of the experimental results seems to be missing. In addition, it is unclear how to decide when to stop to get the good system. It is also unclear how well the ant colony optimization-based EHW could generalize to different situations in these cases.

In essence, such experiments demonstrated the effectiveness of ant colony optimization-based EHW as an alternative to circuit design, but not necessarily as an adaptive or learning system. A more detailed and disciplined approach to experimental studies of generalization in ant colony optimization system will greatly help ant colony optimization-based EHW's study.

D. Disaster Problem

With the increase of complexity of optimization problem, there will occur disaster problem. It appears that the presented ant colony optimization-based EHW would not be sufficient to solve the disaster problem.

There are two possible ways to get around this problem: one is to develop more new ant colony optimization models to supplement it, the other is to improve the hardware architecture to get more high-speed processing EHW schemes.

IV. CONCLUDING REMARKS

This paper reviews the current research on ant colony optimization-based EHW, and a number of issues on the challenges are raised and discussed. In particular, ant colony optimization-based EHW research needs to address issues, such as scalability, online adaptation, generalization, circuit correctness, and potential risk of evolving hardware in a real physical environment. It is argued that a theoretical foundation of EHW should be established before rushing to large-scale EHW implementations.

AKNOWLEDGEMENT

The authors are grateful to the anonymous referees for their valuable comments and suggestions that have led to the better presentation of this paper. This work was supported by the Natural Science Foundation of China under grant #60604009, Aeronautical Science Foundation of China under grant #2006ZC51039 and "New Talent of Blue Sky" Foundation of Beihang University.

REFERENCES

[1] H. D. Garis, "Evolvable Hardware: Genetic Programming of a Darwin Machine". *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, San Mateo, 441-449, 1993.

[2] S. J. Louis. "*Genetic Algorithms* as *a Computational Tool& Design*". PhD thesis, Department of Computer Science, Indiana University, 1993.
[3] X. Yao, T. Higuichi, "Promises and challenges of evolvable hardware". *IEEE Transactions on Systems, Man, and Cybernetics-Part C*,1999, 29(1):87-97.
[4] A. Colorni, M. Dorigo, V. Maniezzo, "Distributed optimization by ant colonies", *Proceedings of European Conference on Artificial Life*, Paris, France, pp. 134-142, 1991.
[5] M. Dorigo, V. Maniezzo, A. Colorni, "Ant system: optimization by a colony of cooperating agents", *IEEE Transaction on Systems, Man, and Cybernetics-Part B*. 26(1), pp.29-41, 1996.
[6] E. Bonabeau, M. Dorigo, G. Theraulaz, "Inspiration for optimization from social insect behavior". *Nature*, 406(6), pp. 39-42, 2000.
[7] H. B. Duan, *Ant Colony Algorithms*: *Theory and Applications*. Beijing: Science Press, 2005.
[8] J. C. Isaacs, R. K. Watkins, S. Y. Foo, "Evolving ant colony systems in hardware for random number generation". *Proceedings of the 2002 Congress on Evolutionary Computation, Piscataway*, vol. 2, pp. 1450-1455, 2002.
[9] B. Scheuermann, K. So, M. Guntsch. "FPGA implementation of population-based ant colony optimization". *Applied Soft Computing*, 4(4), pp. 303-322, 2004.
[10] O. Cheung, P. Leong, Implementation of an FPGA based accelerator for virtual private networks, in: IEEE International
Conference on Field Programmable Technology (FPT), Hong Kong, pp. 34-43, 2002.
[11] J. Villasenor, C. Jones, B. Schoner, Video communications using rapidly reconfigurable hardware, IEEE Transactions on Circuits and Systems for Video Technology, 5(6), pp. 565-567, 1995.
[12] A. Hämäläinen, M. Tommiska, J. Skyttä, 6.78 gigabits per second implementation of the IDEA cryptographic algorithm, in: M. Glesner, P. Zipf, M. Renovell (Eds.), Field-Programmable Logic and Applications. Reconfigurable Computing Is Going Mainstream, LNCS 2438, Springer-Verlag, pp. 760-769, 2002.
[13] S. Bade, B. Hutchings, Fpga based stochastic neural network implementation, in: Proceedings of the IEEE Workshop on
FPGAs for Custom Computing Machines, pp. 189-198, 1994.
[14] R. Gadea, J. Cerd, F. Ballester, A. Mochol, Artificial neural network implementation on a single fpga of a pipelined on-line backpropagation, Proceedings of the 13th Conference on International Symposium on System Synthesis, pp. 225-230, 2000.
[15] P. Graham, B. Nelson, Genetic algorithms in software and in hardware—a performance analysis of workstation and custom computing machine implementations, IEEE Symposium on FPGAs for Custom Computing Machines, pp. 216-225, 1996.
[16] I.M. Bland, G.M. Megson, The systolic array genetic algorithm, an example of systolic arrays as a reconfigurable
design methodology, in: IEEE Symposium on FPGAs for Custom Computing Machines, pp. 260-261, 1998.
[17] C. Aporntewan, P. Chongstitvatana, Hardware implementation of the compact genetic algorithm, in: Proceedings of the 2001 IEEE Congress on Evolutionary Computation, Seoul, pp. 624-629, 2001.
[18] B. Scheuermann, M. Guntsch, M. Middendorf, "Time-scattered heuristic for the hardware implementation of population-based ACO". *Lecture Notes in Computer Science*, vol. 3172, pp. 250-261, 2004.
[19] M. Guntsch, M. Middendorf, "Applying population based ACO to dynamic optimization problems". *Lecture Notes in Computer Science*, vol. 2463, pp. 111-122, 2002.
[20] M. Guntsch, M. Middendorf, "A population based approach for ACO". *Lecture Notes in Computer Science*, vol. 2279, pp. 72-81, 2002.
[21] M. Abd-El-Barr, S. M. Sait, B. A. B. Sarif, "A modified ant colony algorithm for evolutionary design of digital circuits". *Proceedings of the IEEE Congress on Evolutionary Computation*, Piscataway, pp. 708-715, 2003.
[22] M. Abd-El-Barr, S.M. Sait, B.A.B. Sarif, "Ant Colony Algorithm for Evolutionary Design of Arithmetic Circuits". Proceedings of the 15th International Conference on Microelectronics, Cairo, pp. 198-201, 2003.
[23] J. F. Millcr, D. Job, V. K. Vassilev, "Principles in the evolutionary design of digital circuits-part I", *Journal* of *Genetic Programming and Evolvable Machines,* 1(1):8-35, 2000.
[24] C. A. Coello, A . Hemandez, R. L. Zavala, "Ant colony system for the design of combinational logic circuits". *Proceedings of the International Conference on Evolvable Systems*, Edinburg, pp. 21-30, 2000.
[25] C. A. Cocllo, A. D. Christianscn, A. H. Aguirrc, "Towards automated evolutionary design of combinational circuits". *Computers and Electrical Engineering,* 27(l): l-28, 2001.
[26] N. Karaboga, A. Kalinli, D. Karaboga, "Designing digital IIR filters using ant colony. optimization algorithm". *Engineering Application of Artificial Intelligence*, 2(17), pp. 301-309, 2004.
[27] R. Nambiar, P. Mars, Genetic and annealing approaches to adaptive digital filtering. *Proceedings of the 26th IEEEAsilomar Conference on Signals, Systems and Computers*, pp. 871-875, 1992.
[28] D. M. Etter, M. J. Hicks, K. H. Cho, "Recursive adaptive filter design using an adaptive genetic algorithm". *IEEE International Conference on ASSP*, pp. 635-638, 1982.
[29] K. S. Tang, K. F. Man, S. Kwong, Q. He, "Genetic algorithms and their applications". *IEEE Signal Processing Magazine*, 13(6), pp. 22-37, 1996.
[30] D. Merkle, M. Middendorf, "Fast ant colony optimization on reconfigurable processor arrays". *Proceedings of the 15th International Parallel & Distributed Processing Symposium*, San Francisco, pp. 1465-1472, 2001.
[31] B. A. B. Sarif, M. Abd-Ei-Barr, S. M. Sait, U. Al-Saiari, "Fuzzified ant colony. optimization algorithm for efficient combinational circuits synthesis". *Proceedings of the 2004 Congress on Evolutionary Computation*, Oregon, vol.2, pp. 1317-1324, 2004.