Transactions of the Institute of Measurement and Control 34, 2/3 (2012) pp. 318–333

Ant colony optimization-based bio-inspired hardware: survey and prospect

Haibin Duan[1], Yaxiang Yu[1], Jie Zou[2] and Xing Feng[2]

[1]National Key Laboratory of Science and Technology on Holistic Control, Department of Automatic Control, Beihang University, Beijing 100191, China
[2]Key Laboratory of National Defense Science and Technology on Fire Control Technology, Luoyang Institute of Electro-Optical Equipment, Luoyang 471009, China

Bio-inspired hardware (BHW) refers to hardware that can change its architecture and behaviour 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 behaviour of real ant colonies. In this paper, we take a broad survey on the recent progresses of ant colony optimization-based BHW, which includes ant colony optimization-based fuzzy controller, ant colony optimization-based hardware for the Travelling Salesman Problem (TSP), digital circuits, digital infinite impulse-response (IIR) filters, hardware-oriented ant colony optimization with look-up table and hardware/software partition. Some important issues of the challenges of ant colony optimization-based BHW are also presented. Online realization, robustness, generalization, disaster problems, theoretical analysis, implementation, swarm robotics, applications and hybrid approaches are eight key challenging issues for the ant colony optimization-based BHW.

Key words: ant colony optimization; bio-inspired hardware (BHW); positive feedback; prospect; robustness; swarm robotics.

1. Introduction

Bio-inspired hardware (BHW) is also named 'evolvable hardware' (EHW), which refers to hardware that can change its architecture and behaviour dynamically and

Address for correspondence: Haibin Duan, National Key Laboratory of Science and Technology on Holistic Control, Department of Automatic Control, Beihang University, Beijing 100191, China.
E-mail: hbduan@buaa.edu.cn
Figures 8–11 appear in colour online: http://tim.sagepub.com

 10.1177/0142331210366689

autonomously by interacting with its environment, and BHW has been proposed as a new method for designing systems for complex real-world applications. At present, almost all BHW use an evolutionary algorithm (EA) as their main adaptive mechanism. It was Garis who made the first move to investigate the design of evolving circuits. In his paper, Garis (1993) suggested the establishment of a new field of research called EHW (BHW). At about the same time, the first work in evolutionary design of digital circuits was carried out by Louis (1993). A complete review and taxonomy of the BHW field is described by Yao and Higuichi (1999).

Ant colony optimization is a meta-heuristic algorithm for the approximate solution of combinatorial optimization problems that has been inspired by the foraging behaviour of real ant colonies (Colorni *et al.*, 1991). 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 principle of this phenomenon is that ants deposit a chemical substance (called pheromone) on the ground, thus marking a path by the pheromone trail. In this process, a kind of positive feedback mechanism is adopted. An ant encountering a previously laid trail can detect the density of the pheromone trail (Bonabeau *et al.*, 2000). It decides with high probability to follow the shortest path, and reinforces the trail with its own pheromone. The larger the amount of pheromone on a particular path, the larger probability is that an ant selects that path and the path's pheromone trial will become denser. At last, the ant colony collectively marks the shortest path, which has the largest pheromone amount. Such a simple indirect communication among ants actually embodies a kind of collective learning mechanism. Figure 1 shows the principle that ants exploit pheromone to establish the shortest path from a nest to a food source and back.

The basic mathematical model of ant colony optimization has first been applied to the TSP (Travelling Salesman Problem). By now, ant colony optimization has been applied to combinatorial optimization problems such as the TSP, JSP (job-shop problem), QAP (quadratic assignment problem), VRP (vehicle routing problem), GCP (graph colouring problem), and so on (Duan, 2005).

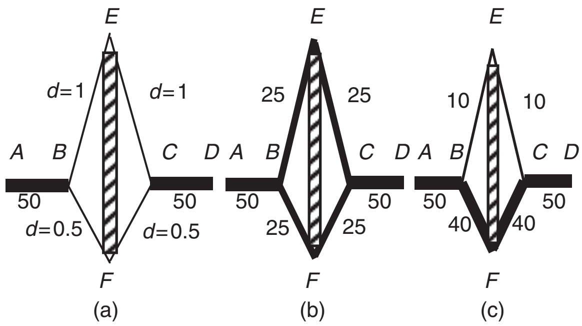

Figure 1 Diagram of ant colony optimization principle

Isaacs *et al.* first proposed a new random number generator scheme, which employs simulated ant colonies evolved by a genetic algorithm to produce very high-quality random bit generators in EHW (Isaacs *et al.*, 2002). However, a random number generator was not implementable in hardware at that time. Scheuermann and his colleagues took the single machine total tardiness problem (SMTTP) as a typical example, presenting a hardware implementation of population-based ant colony optimization (P-ACO) on field-programmable gate arrays (FPGAs) (Scheuermann *et al.*, 2004, 2007). In that work, they described the P-ACO algorithm and presented a circuit architecture that facilitates efficient FPGA implementations.

In this paper, we take a broad survey on the progresses of ant colony optimization-based BHW and address some important issues of the challenges of ant colony optimization-based BHW in more detail in later sections. The remainder of this paper is organized as follows. Section 2 describes the overall newest approaches in ant colony optimization-based BHW, which include an ant colony optimization-based fuzzy controller, ant colony optimization-based hardware for the TSP, digital circuits, digital infinite impulse-response (IIR) filters, hardware-oriented ant colony optimization with look-up table, hardware/software partition, and so on. Subsequently, a detailed analysis of challenges lying in ant colony optimization-based BHW is presented in Section 3. Our concluding remarks are contained in Section 4.

2. Progresses of ant colony optimization-based BHW

2.1 FPGA implementation of ant colony optimization for fuzzy controller

An ant colony optimization application to a fuzzy controller (FC) design (called the ACO-FC) was first proposed in Juang *et al.* (2008) for improving design efficiency and control performance (as shown in Figure 2), as well as ant colony optimization hardware implementation.

An FC's antecedent part, ie, the 'if' part of its composing fuzzy if–then rules, is partitioned in grid-type, and all candidate rule consequent values are then listed. An ant trip is regarded as a combination of consequent values selected from every rule. A pheromone matrix among all candidate consequent values is constructed. Searching for the best one among all combinations of rule consequent values is based mainly on the pheromone matrix. The objective of the best ant selection module is to find the global best ant so far over all iterations and to record the selected consequent combination of this ant once it is found in an iteration. The best ant selection module architecture is shown in Figure 3. The inputs of this module are the performance measure per ant sent from the PC and consequent combination from the consequent selection module. Instead of storing the selected consequent combinations of all the ants, only two registers Reg1 and Reg2 are used to store the best and current ant consequent combinations, respectively. The selected consequent combination for each ant is stored

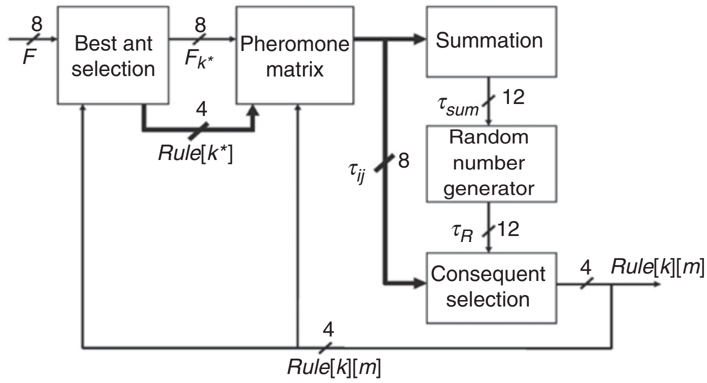

Figure 2 Architecture of hardware-implemented ant colony optimization

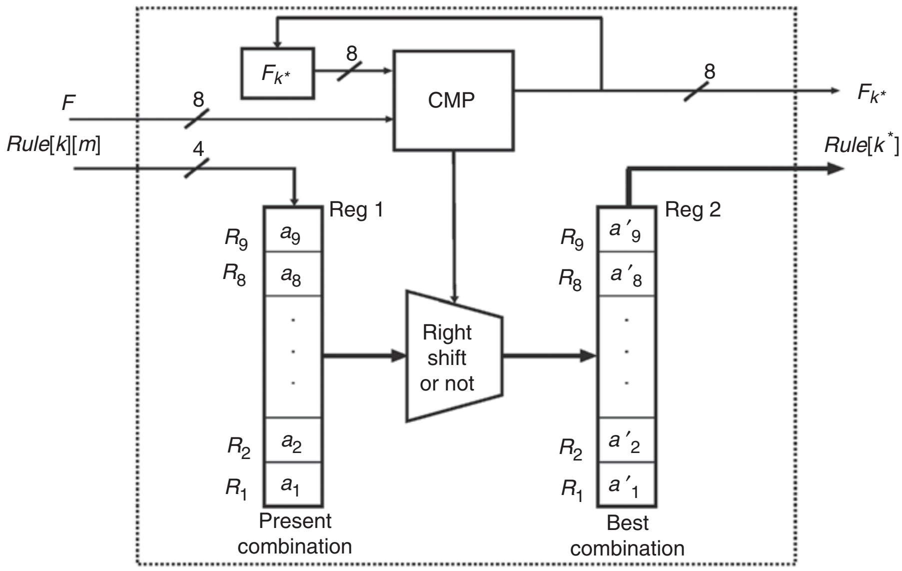

Figure 3 Best ant selection module

in Reg1. Then, the performance measure of this ant is compared with the global best performance (set to zero initially) through a comparator (the CMP circuit).

The proposed ACO-FC performance is shown to be better than other meta-heuristic design methods on simulation examples. The ant colony optimization used in ACO-FC is based on the known ant colony system and is hardware implemented on a FPGA chip. The ant colony optimization chip application to fuzzy control of a simulated water bath temperature control problem has verified the designed chip effectiveness.

2.2 Ant colony optimization hardware for TSP

The TSP is one of the most important problems in combinational optimization. Much work has been done on this problem using ant colony optimization. However, there are some disadvantages, such as the long interval in operation and stagnation in evolution. In order to overcome these disadvantages, a new architecture for high speed ant colony optimization has been proposed in Yoshikawa and Terai (2007). The proposed architecture introduces parallel processing for city selection processing, dedicated hardware for floating point arithmetic and a new identification code (ID) structure to manage the pheromone with one SRAM (Static Random Access Memory).

Figure 4 shows part of the parallel processing. As shown in Figure 4, the city selection block consists of an ID generating circuit, memory controller, city selection circuit, judgment circuit, visit register, non-visit circuit, flag circuit selector and selection controller. The dedicated hardware for floating point arithmetic is introduced in order to calculate the pheromone effectively. The ID structure consists of four parts, which are read–write ID, memory ID, data ID and data. Three IDs are not stored in SRAM. Experimental results to evaluate the proposed algorithm showed improvement in comparison with software processing.

2.3 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 behaviour of each component of the designed circuit is well known (Abd-El-Barr *et al.*, 2003a, 2003b). 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. 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 complex 3-bit multiplier. Unfortunately, the 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. Abd-El-Barr *et al.* presented a multi objective evolutionary logic design based on ant colony optimization for digital circuits (2003b), 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.

2.4 Hardware-oriented ant colony optimization with look-up table

A novel hardware-oriented ant colony optimization (H-ACO) is proposed to achieve high-speed optimization based on mechanism of ant colony optimization algorithm by Yoshikawa (2008). The characteristics of the proposed 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 ant colony optimization; 2) a new technique using a 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 the effectiveness of the proposed algorithm.

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),

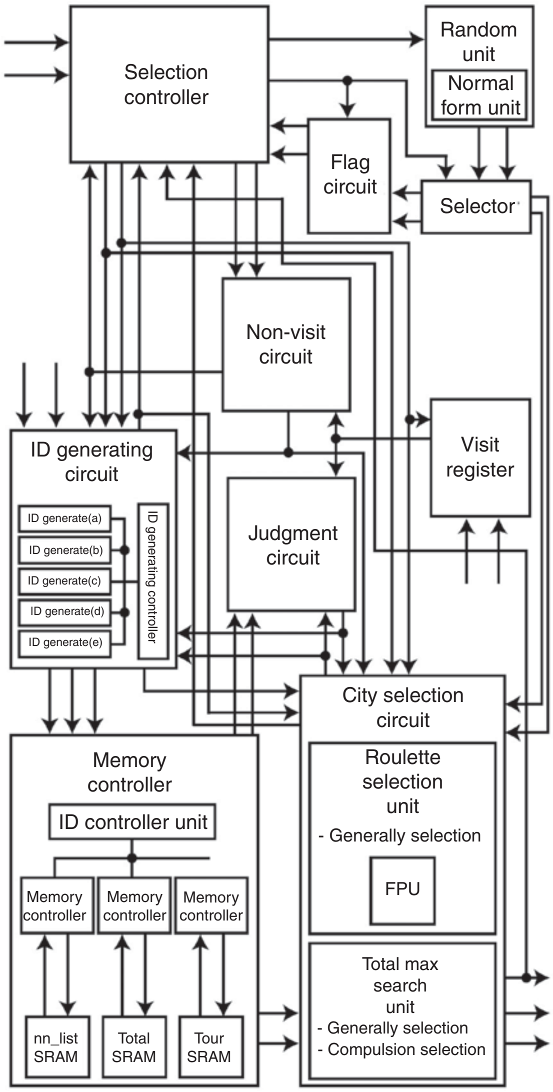

Figure 4 The city selection block

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 Figure 5.

Figure 6 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.

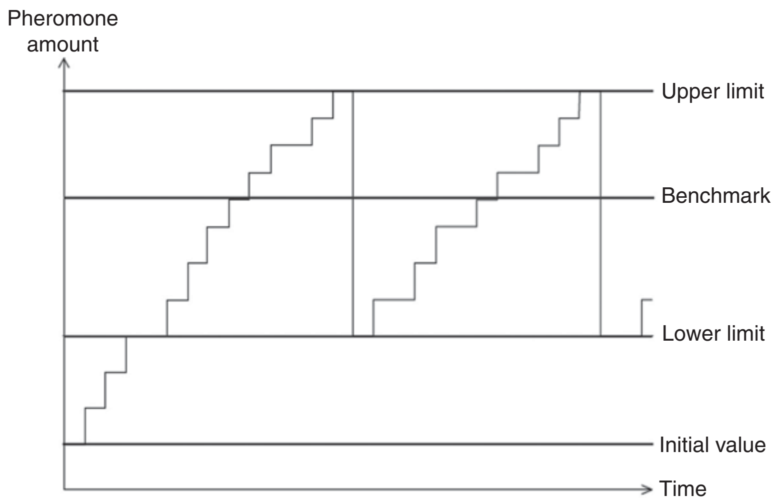

Figure 5 Example of transition of pheromone

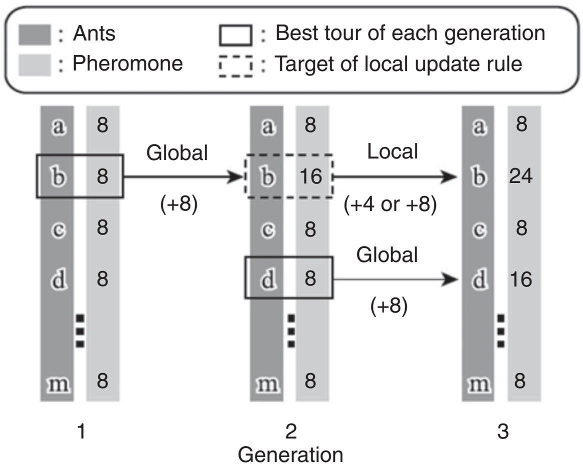

Figure 6 Example of local and global update rule

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, this is only the values that are multiples of 4. Then, the LUT, into which both the heuristics and pheromone values are input, is created. As a result, the proposed algorithm achieved not only high speed processing, but also maintenance of the quality of solutions. Experiments using benchmark data also proved the effectiveness of the proposed algorithm.

2.5 Ant colony optimization-based digital IIR filters

In order to transform and analyse signals that have been sampled from analogue sources, digital signal processing (DSP) algorithms are employed.

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 the 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 fundamental problem is that they might have a multi-modal 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 that 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. Karaboga *et al.* (2004) applied a touring ant colony optimization (TACO) algorithm to digital IIR filter design (as shown in Figure 7).

In Figure 7, 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.

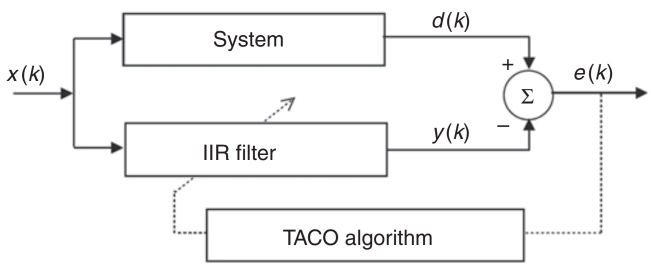

Figure 7 System identification process using IIR filter

2.6 Hardware/software partition using ant colony optimization

The flexibility, performance and cost effectiveness of reconfigurable architectures have lead to its widespread use for embedded applications. Coarse-grained reconfigurable system design is very complex for multi-fields experts to collaborate on application algorithm design, hardware/software co-design and system decision. However, the existing reconfigurable system design methods and environments can only support hardware/software co-design, ignoring the collaboration between multi-field experts. Wang *et al.* (2008) presented a collaborative partition approach of coarse-grained reconfigurable system design using evolutionary ant colony optimization. They created a distributed collaborative design environment for system decision engineers, software designers, hardware designers and application algorithm developers. The method not only utilizes the advantages of ant colony optimization for searching global optimal solutions, but also provides a framework for multi-field experts to work collaboratively. Experimental results also given to show that the proposed method can improve the quality and speed of hardware/software partition for coarse-grained reconfigurable system design efficiently.

In addition, there has been much great progresses in other ant colony optimization-based BHWs: Merkle and Middendorf (2001) 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; Sarif *et al.* (2004) presented a fuzzified and colony algorithm for the multi-objective optimization of logic circuits.

3. Challenges of ant colony optimization-based BHW

Although adaptive ant colony optimization-based BHW has made great progress in the past several years, there are some fundamental and interesting issues in ant colony optimization-based BHW that are worth probing further. Generally, these problems fall into two categories: original design and adaptive systems. The original design uses ant colony optimization algorithms to design a system that meets a predefined specification. Adaptive systems reconfigure an existing design to counteract faults or a changed operational environment. The typical challenges for the ant colony optimization-based BHW can be generalized as eight aspects: online realization, robustness, generalization, disaster problems, theoretical analysis, implementation, application and hybrid approaches (as shown in Figure 8).

3.1 Online realization

Online realization requires BHW learning to be incremental and responsive Yao and Higuichi (1999), and online realization of ant colony optimization-based BHW is the key issue of this research area. In spite of the great progress in ant colony

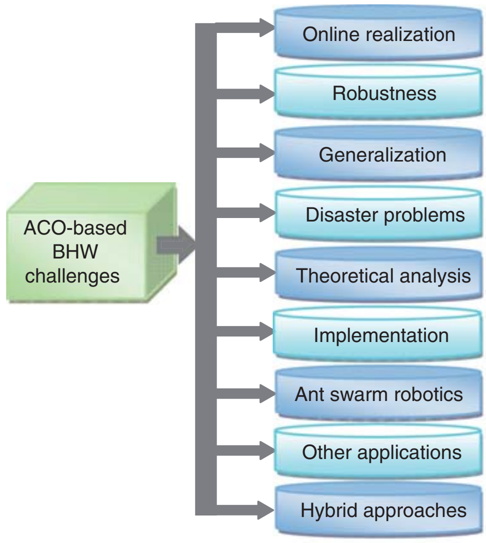

Figure 8 Typical challenges for the ant colony optimization-based BHW

optimization-based BHW, almost no satisfactory work has been reported on online realization of ant colony optimization-based BHW. Most of the achieved results are obtained offline, in which the proposed ant colony optimization scheme happens during the learning phase of the BHW. A prominent disadvantage of offline ant colony optimization-based BHW is that the evolutionary speed is slow, and this limits the wide application of ant colony optimization in industry. An effective way to solve this problem is to develop a knowledge-based adaptive ant colony optimization-based BHW, where constraints and knowledge about the uncertain environment in which the ant colony optimization-based BHW 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.

3.2 Robustness

Robustness is a key issue for the reliable ant colony optimization-based BHW. When there is failure, the failure detection scheme of ant colony optimization-based BHW should immediately detect the error and initiate a reprogramming that correctly programs in the hardware. However, whether the failure detection scheme succeeds in each failure is an open problem. In fact, many reported ant colony optimization-based BHW applications are not satisfactory in detecting the faults correctly each time. Hence, robustness of ant colony optimization-based BHW may be achieved through various means of fault detection and repair or through fault tolerance, and the robustness of the ant colony optimization-based BHW quality should be further investigated in the future.

3.3 Generalization

Generalization is also a key issue for any learning or adaptive systems (Yao and Higuichi, 1999), including ant colony optimization-based BHW. Evaluating the ant colony optimization-based BHW's generalization can be a difficult task because of different implementations. This difficulty is closely related to that of evaluating the generalization ability of the 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 BHW. Some experiments on ant colony optimization-based BHW 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 a good system. It is also unclear how well the ant colony optimization-based BHW could generalize to different situations in these cases.

3.4 Disaster problem

With the increase of complexity of optimization problem, a disaster problem will occur (Duan and Yu, 2007). It appears that the presented ant colony optimization-based BHW 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 and the other is to improve the hardware architecture to get more high-speed processing BHW schemes.

3.5 Theoretical analysis

Theoretical analysis cannot only provide insight into the working of the ant colony optimization, but also is very useful for the practitioner that wants to implement efficient algorithms with BHW. However, the existing theoretical analysis (especially convergence proof) is mainly for particular ant colony optimization models. A theoretical foundation of ant colony optimization-based BHW should be established before rushing into large-scale BHW implementations. This is because, generally, either infinite time or infinite space is required for a stochastic optimization algorithm to converge to an optimal solution.

3.6 Implementation

Currently, almost all the ant colony optimization-based BHWs are implemented with FPGA. FPGA is a semiconductor device that can be configured by the customer or designer after manufacturing – hence the name 'field-programmable' (as shown in

Figure 9 Typical FPGA chip

Figure 9). FPGAs are programmed using a logic circuit diagram or a source code in a hardware description language (HDL) to specify how the chip will work. They can be used to implement any logical function that an application-specific integrated circuit (ASIC) could perform, but the ability to update the functionality after shipping offers advantages for many applications. FPGAs contain programmable logic components called 'logic blocks', and a hierarchy of reconfigurable interconnects that allow the blocks to be 'wired together' – somewhat like a one-chip programmable breadboard. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

The adoption of FPGAs in the ant colony optimization-based BHW implementation is currently limited by the complexity of FPGA design compared with conventional software and the extremely long turn-around times of current design tools, where a long time wait is necessary after even minor changes to the source code.

A digital signal processor is a specialized microprocessor designed specifically for DSP, generally in real-time computing (as shown in Figure 10). 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. 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

Figure 10 Typical DSP chip

analogue systems. In this way, DSP is another implementation tool for the ant colony optimization-based BHW.

3.7 Ant swarm robotics

A swarm robot is a new type of intelligent robot. A swarm of relatively simple physically embodied swarm robots can be designed and controlled to accomplish tasks collectively that are beyond the capabilities of a single robot (Hettiarachchi and Spears, 2009a, 2009b). The advantages of swarm robotics are as follows:

(1) Complex control is achievable through simple local interactions of the swarm members.
(2) The results scale well with larger numbers of robots.
(3) The swarm is robust to failure of individual members.

Algorithms, techniques and methods based on swarm robotics principles have been successfully applied to a wide range of complex problems, and a new special issue on 'Swarm Robotics' has been published in the *International Journal of Intelligent Computing and Cybernetics* (Volume 2, No. 4, 2009). How to apply ant colony optimization to swarm robotics is a new orientation, which can make the robotics more smart and adaptive.

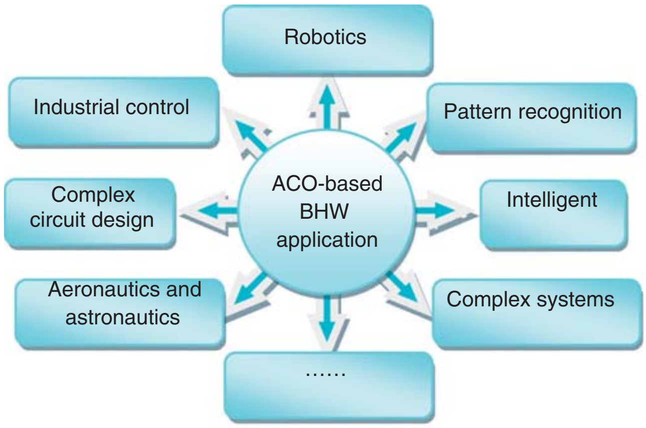

Figure 11 Ant colony optimization-based BHW promising application fields

3.8 Other applications

By now, the ant colony optimization-based BHW can only find its application in solving TSPs, JSPs and simple circuit design problems. In fact, the ant colony optimization-based BHW can also be applied to the design of complex circuits, intelligent control, industrial control, robotics, pattern recognition, aeronautics and astronautics, and complex systems (as shown in Figure 11).

3.9 Hybrid approaches

Ant colony optimization is just one type of bio-inspired algorithm for optimization problems; it has strong robustness and is easy to combine with other methods in optimization, but it has the shortcomings of stagnation that limit the wider application to the various areas. Currently, a series of schemes on improving the ant colony optimization model is proposed. How to combine the ant colony optimization with genetic algorithms, particle swarm optimization and artificial bee colony optimization in BHW is a challenging problem. How to combine ant colony optimization with chaos and quantum computing is also a significant research interest in the ant colony optimization-based BHW development.

4. Conclusion

This work reviews the current research on ant colony optimization-based BHW, and a number of issues on the typical challenges are raised and discussed in detail. In particular, ant colony optimization-based BHW research needs to address issues such as online realization, robustness, generalization, disaster problems, theoretical analysis, implementation, application and hybrid approaches. In conclusion, we hope this paper has achieved its goal: to convince the readers that ant colony optimization-based BHW, as a novel type of BHW, is very promising and worth further research.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Grant No. 60975072 and 60604009), the Aeronautical Science Foundation of China, the Program for New Century Excellent Talents in University of China (Grant No. NCET-10-0021), the 'Beijng NOVA Program' Foundation of China (Grant No. 2007A0017) and the Fundamental Research Funds for the Central Universities (Grant No. YWF-10-01-A18). The authors are grateful to the anonymous referees for their valuable comments and suggestions.

References

Abd-El-Barr, M., Sait, S.M. and **Sarif, B.A.B.** 2003a: A modified ant colony algorithm for evolutionary design of digital circuits. *Proceedings of the IEEE Congress on Evolutionary Computation*, Piscataway, 708–15.

Abd-El-Barr, M., Sait, S.M. and **Sarif, B.A.B.** 2003b: Ant colony algorithm for evolutionary design of arithmetic circuits. *Proceedings of the 15th International Conference on Microelectronics*, Cairo, 198–201.

Bonabeau, E., **Dorigo, M.** and **Theraulaz, G.** 2000: Inspiration for optimization from social insect behavior. *Nature* 406, 39–42.

Colorni, A., Dorigo, M. and **Maniezzo, V.** 1991: Distributed optimization by ant colonies. *Proceedings of European Conference on Artificial Life*, Paris, 134–42.

Duan, H.B. 2005: *Ant colony algorithms: theory and applications*. Science Press.

Duan, H.B. and **Yu, X.F.** 2007: Progresses and challenges of ant colony optimization-based evolvable hardware. *Proceedings of the IEEE Workshop on Evolvable and Adaptive Hardware*, Honolulu, HI, 67–71.

Garis, H.D. 1993: Evolvable hardware: genetic programming of a Darwin machine. *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, San Mateo, CA, 441–9.

Hettiarachchi, S. and **Spears, W.M.** 2009a: Guest editorial: Special issue on 'swarm robotics'. *International Journal of Intelligent Computing and Cybernetics* 2, no. 4.

Hettiarachchi, S. and **Spears, W.M.** 2009b: Distributed adaptive swarm for obstacle avoidance. *International Journal of Intelligent Computing and Cybernetics* 2, 644–71.

Isaacs, J.C., Watkins, R.K. and **Foo, S.Y.** 2002: Evolving ant colony systems in hardware for random number generation. *Proceedings of the 2002 Congress on Evolutionary Computation*, Piscataway, NJ, Volume 2, 1450–5.

Juang, C.F., **Lu, C.M.**, **Lo, C.** and **Wang, C.Y.** 2008: Ant colony optimization algorithm for fuzzy controller design and its FPGA implementation. *IEEE Transactions on Industrial Electronics* 55, 1453–62.

Karaboga, N., **Kalinli, A.** and **Karaboga, D.** 2004: Designing digital IIR filters using ant colony optimization algorithm. *Engineering Application of Artificial Intelligence* 2, 301–9.

Louis, S.J. 1993: Genetic algorithms as a computational tool & design. PhD thesis, Department of Computer Science, Indiana University.

Merkle, D. and **Middendorf, M.** 2001: Fast ant colony optimization on reconfigurable processor arrays. *Proceedings of the 15th International Parallel & Distributed Processing Symposium*, San Francisco, CA, 1465–72.

Sarif, B.A.B., Abd-Ei-Barr, M., Sait, S.M. and **Al-Saiari, U.** 2004: Fuzzified ant colony optimization algorithm for efficient combinational circuits synthesis. *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, Portland, OR, vol. 2, 1317–24.

Scheuermann, B., **So, K.** and **Guntsch, M.** 2004: FPGA implementation of population-based ant colony optimization. *Applied Soft Computing* 4, 303–22.

Scheuermann, B., **Janson, S.** and **Middendorf, M.** 2007: Hardware-oriented ant colony optimization. *Journal of Systems Architecture* 53, 386–402.

Wang, D.W., Li, S.K. and **Dou, Y.** 2008: Collaborative hardware/software partition of coarse-grained reconfigurable system using evolutionary ant colony optimization. *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, 679–84.

Yao, X. and **Higuichi, T.** 1999: Promises and challenges of evolvable hardware. *IEEE Transactions on Systems, Man, and Cybernetics C* 29, 87–97.

Yoshikawa, M. and **Terai, H.** 2007: Architecture for high-speed ant colony optimization. *Proceedings of the IEEE International Conference on Information Reuse and Integration*, Las Vegas, NV, 1–5.

Yoshikawa, M. 2008: Hardware-oriented ant colony optimization considering intensification and diversification. In: Bednorz, W. editor. *Advances in greedy algorithms*. I-Tech, 359–68.