

Hardware Architecture for Hybrid Genetic Algorithm

Masaya Yoshikawa, Hironori Yamauchi, and Hidekazu Terai

Abstract—This paper discusses a new dedicated hardware architecture for hybrid optimization based on Genetic algorithm (GA) and Simulated Annealing (SA). The hybrid optimization of GA and SA in the proposed architecture realized searching not only globally but also locally. To keep general purpose, the proposed architecture is flexible for many genetic operations on GA and achieves high speed processing by adopting dedicated hardware. Simulation result evaluating the proposed architecture is shown to the effectiveness.

Index Terms—Genetic Algorithm, Simulated Annealing, Dedicated Hardware, TSP

I. INTRODUCTION

Genetic Algorithm (GA)[1] was proposed by Holland as an algorithm for probabilistic search, learning, and optimization, and is based in part on the mechanism of biological evolution and Darwin's theory of evolution. This algorithm is a powerful search tool, particularly when applied for combinatorial optimization problems and has been investigated for use in scheduling and combinatorial problems, as well as applications in various fields[2]-[7]. However, the implementation of an efficient GA often faces two major problems, on one side, the premature convergence to local optima and on the other the requirements for the GA search of long times in order to reach an optimal or a good suboptimal solution. In order to prevent the premature convergence, the coupling of GA and local search methods, such as Simulated Annealing (SA)[8]-[11], to form hybrid GA can be advantageous. SA repeatedly generates succeeding solutions using the local search procedure. Some of them are accepted and some will be rejected, according to a predefined acceptance rule. The acceptance rule is motivated by an analogy with annealing processes in metallurgy as shown in Fig.1 (a). GA converges on globally competitive solutions irrespective of local optima in the search space. SA can potentially improve on the solutions discovered by genetic algorithm by climbing to the optima of their corresponding attraction basins.

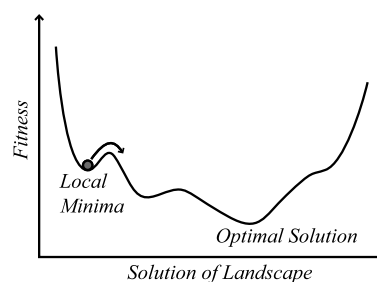
On the other hand, GA repeatedly propagates generations for a large population by applying three operators, which consist selection, crossover and mutation as shown in Fig.1 (b). Therefore, GA requires the long computational time. The

dedicated hardware for genetic algorithm processing is an important in order to reduce processing time.

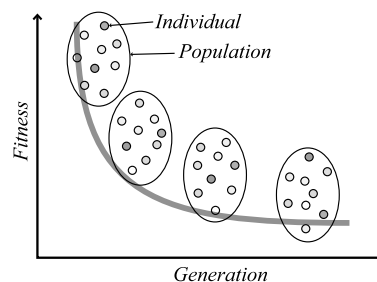
In this paper, we propose the new dedicated hardware architecture for hybrid optimization based on GA and SA. Examples of GA hardware have been proposed by Scott et al.[12], Graham and Nelson[13], and Yoshida et al.[14]. Scott et al. developed a hardware-based GA and demonstrated its superiority to software in speed and solution quality. Yoshida et al. proposed a genetic algorithms processor based on a steady-state GA, in which generations do not change. Most of these previous works deal with small-scaled problems and limit to some fixed genetic operations. However, no studies have ever seen the effect of the dedicated hardware for GA and SA.

II. ARCHITECTURE METHODOLOGY

Since the GA, or multiple point search algorithm, generates a new search point focusing on crossover calculation, a systematic search near a good solution is poor. While the local search (one point search algorithm) is excellent in a local search, its global search is poor. Hybrid search combines algorithms having such complementary relationships, so the ability to search for a solution is improved.



(a) Search of Simulated Annealing



(b) Search of Genetic Algorithm

Fig.1 One point search and population based search

Masaya Yoshikawa is with the Department of Information engineering, Meijo University (corresponding author to provide e-mail: yosikawa@ritusmei.ac.jp).

Hironori Yamauchi and Hidekazu Terai are with the Department of System LSI, Ritsumeikan University.

Several proposals on hybridization have been made [15]-[24] in which the local search is made for a systematic search near a good solution and the GA conducts the global search. Therefore, the proposed architecture realizes hybrid optimization using GA and SA. Fig.2 shows the processing flow of the proposed architecture. For GA, it is important to set suitable evaluation parameters for controlling the selection of individuals. The selection operations are implemented as roulette wheel selection and ranking selection.

The roulette wheel selection method uses the percentage represented by the evaluation value for each individual with respect to the sum total of evaluation values for all individuals as the selection probability for each individual. As a result, if the evaluation value for the i -th individual is $f(i)$, then a roulette wheel showing this percentage as a wedge can be used, as shown in Fig.3.

To put this concretely, let us consider the case of the evaluation values shown in Fig.4 (a). First, the product of the individual evaluation values is calculated in alphabetical order, as shown in Fig.4 (b). Next, a random number taking a value in the range of "0" to "product of the evaluation values" is generated. For Fig.4 (b), if the generated random number is from "0" to "99", individual A will be selected. And then, if it is from "100" to "399", individual B will be selected. The Roulette wheel selection runs in this fashion.

In relation to the Ranking selection, this method attempts to preserve descendants using a probability determined for each initial rank assigned to individuals by using the evaluation value.

To put this in concrete terms, let us consider a case in which there are four individuals with the evaluation values shown in Fig.4 (a), and with the selection probabilities shown in Fig.4 (c).

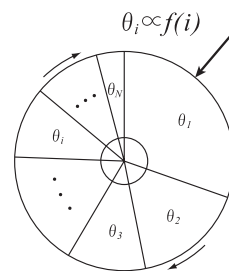


Fig.3 Example of roulette wheel

individual	A	B	C	D
Evaluation Value	100	300	150	50

(a) Evaluation values for individuals

Product	100	400	550	600
	A	B	C	D

(b) Product of evaluation values

Ranking	1	2	3	4
Selection probability	40%	30%	20%	10%

(c) Selection probabilities for individuals

Product	40	70	90	100

(d) Product of selection probabilities

Product	40	70	90	100
	B	C	A	D

(e) Correspondence between selection probability and individual

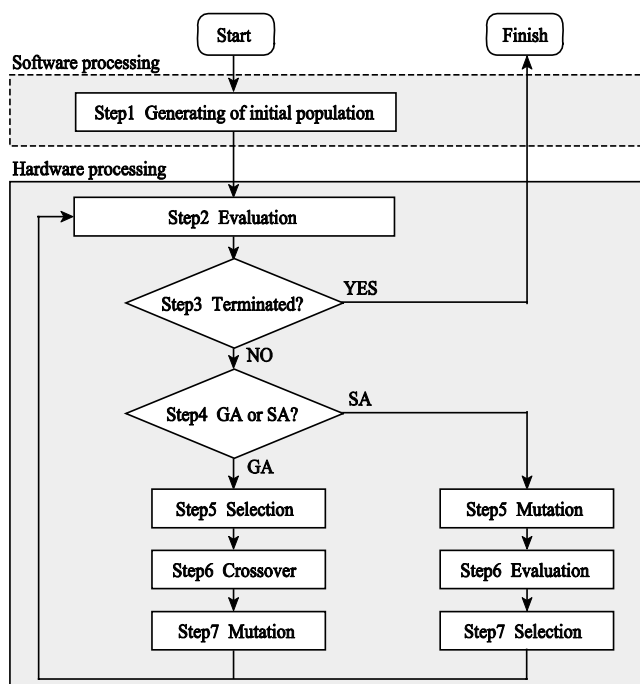


Fig.2 Processing flow of the proposed architecture

Fig.4 Processing flow of the proposed architecture

First, the product of the selection probability is calculated in descending rank as shown in Fig.4 (d) as a form of preprocessing (performed only once for the first generation).

Next, individuals are sorted in descending order for the evaluation value. As is illustrated in Fig.4 (e), the sorted individuals correspond to the products of the selection probability in Fig.4 (d). Then random numbers are generated in the same fashion as seen in the Roulette wheel selection, and the ranking selection is implemented by selecting individuals.

The block diagram of the proposed architecture is shown in Fig.5. The proposed architecture consists of selection circuit, mutation circuit, crossover circuit, memory controller, generation management circuit, serial communication circuit, evaluation circuit, FPGA board interface circuit, and SRAM.

Moreover, the proposed architecture introduces the self-control circuit in each sub circuit as shown in Fig.5. The self-control circuit is required for the shake-hand type of data communication in each circuit and improves independence of each sub circuit.

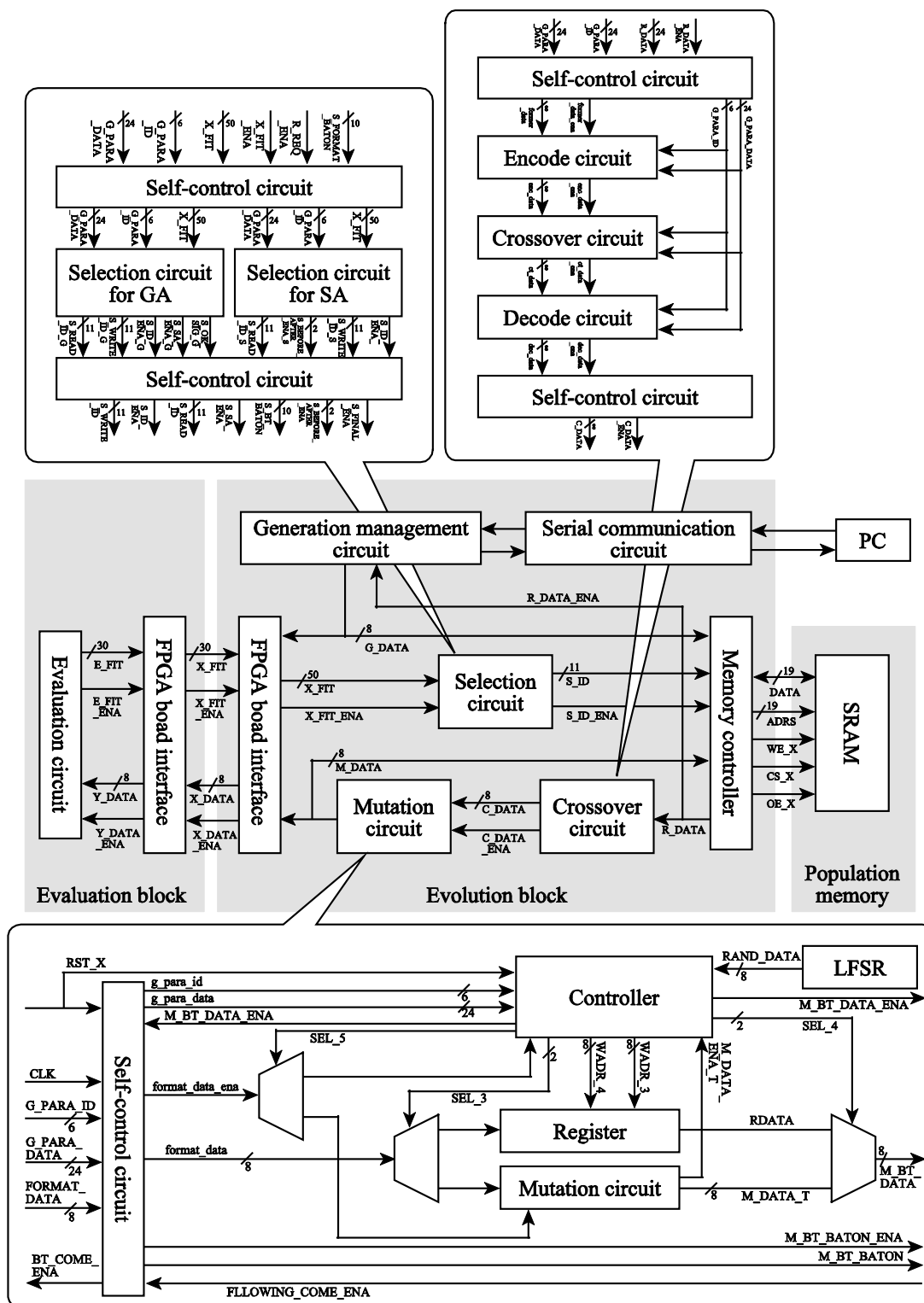


Fig.4 Block diagram

III. EXPERIMENT AND DISCUSSION

In order to evaluate the proposed architecture, we applied it to TSP problem. The traveling salesman problem (TSP) is one of the most important problems in combinational optimization. The proposed architecture has been designed by Verilog-HDL and synthesized by the Synplify Synplify.

The clock frequency of the proposed architecture is set up with 20 MHz. Table.1 shows the gate size. ESBs indicate memory blocks of FPGA in Table.1. We simulate the performance of the proposed architecture with the number of steps and compare it with software processing using TSP benchmark data (eil51).

Table.1 Gate size

Circuit name	Logic resource (%)	EBS (%)
Generating management	1	1
Memory controller	8	15
Selection	50	28
Crossover	7	5
Mutation	8	2
Evaluation	24	92

The software processing was implemented in the C language and run on a Linux PC. The clock frequency of the Linux PC is 2.4GHz. The proposed architecture improved the speed of 22% in comparison with software processing. The clock frequency is 20MHz for a limit of FPGA in this research. However, the proposed architecture is speeded up more if it is realized by LSI.

IV. CONCLUSION

In this paper, we proposed the new hardware architecture for hybrid optimization based on Genetic Algorithm and Simulated Annealing. The hybrid optimization of GA and SA in the proposed architecture realized searching not only globally but also locally. To keep general purpose, the proposed architecture is flexible for many genetic operations on GA and achieves high speed processing by adopting dedicated hardware. Simulation result evaluating the proposed architecture is shown to the effectiveness. Moreover, the gate size of the proposed architecture is reduced by adopting the combine architecture of GA and as SA. In relation to future works, we will expand the architecture to the Artificial intelligence chip.

REFERENCES

[1] J.Holland, *Adaptation in Natural Artificial Systems*, the University of Michigan Press (Second edition ; MIT Press), (1992)

[2] M.Yoshikawa, T. Fujino, and H.Terai, "A Novel Genetic Algorithm Routing Technique in 3-Dimentional Space", *Proceedings of the 10th World Multiconference on Systemics, Cybernetics and Informatics*, Vol.1, pp.70-75, 2006.

[3] S.Nakaya, Koide and T. Wakabayashi, S, "An adaptive genetic algorithm for VLSI floor planning based on sequence-pair", *Proc. IEEE ISCAS*, Vol.3, pp.65-68, 2000.

[4] M.Yoshikawa and H.Terai, "Asynchronous Parallel Genetic Algorithm for Congestion-Driven Placement Technique", *Proceedings of 3rd International Conference on Software Engineering Research, Management and Applications*, pp.130-136, 2005.

[5] K.Y.Lee and P.S.Mohamed, "A real-coded genetic algorithm involving a hybrid crossover method for power plant control system design", *Proceedings of the 2002 Congress on Evolutionary Computation*, Volume 2, pp.1069-1074, 2002.

[6] L.Zhang, Raut, R., Ling Wang, and Yingtao Jiang, "Analog Module Placement Realizing Symmetry Constraints Based on a Radiation Decoder", *Proceedings of IEEE 47th Midwest Symposium on Circuits and Systems*, pp. I-481-484, 2004.

[7] H.Kanoh and T.Nakamura, "Knowledge based genetic algorithm for dynamic route selection", *Proceedings of Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, Vol.2, pp.616-619, 2000.

[8] N.Metropolis, and et al., "Equation of state calculations by fast computing machines", *Journal of Chemistry and physics*, pp.1087-1092, 1953.

[9] V.Ravi, B.S.N.Murty, and J.Reddy, "Nonequilibrium simulated-annealing algorithm applied to reliability optimization of complex systems", *IEEE Transactions on Reliability*, Volume 46, Issue 2, pp.233-239, 1997.

[10] K.Kurbel, B.Schneider, and K.Singh, "Solving optimization problems by parallel recombinative simulated annealing on a parallel computer-an application to standard cell placement in VLSI design", *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Volume 28, Issue 3, pp.454 - 461, 1998.

[11] Lipo Wang, Sa Li, F.Tian, and Xiuju Fu, "A noisy chaotic neural network for solving combinatorial optimization problems: stochastic chaotic simulated annealing", *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Volume 34, Issue 5, pp.2119 - 2125, 2004.

[12] S.D.Scott, A.Samal and S.Seth, "HGA:A Hardware-Based Genetic Algorithm", *Proceedings of International Symposium on Field-Programmable Gate Array*, pp.53-59, 1995.

[13] P.Graham and B.Nelson, "Genetic Algorithm in Software and in Hardware A performance Analysis of Workstation and custom Computing Machine Implementations", *FPGAs for Custom Computing Machines*, pp.216-225, 1996.

[14] N.Yoshida, T.Moriki, T.Yasuoka, "GAP:Genetic VLSI Processor for Genetic Algorithms", *Proceedings of Second International ICSC Symposium on Soft Computing*, pp.341-345, 1997.

[15] M.Alabau, L.Idoumghar, R.Schott, "New hybrid genetic algorithms for the frequency assignment problem", *IEEE Transactions on Broadcasting*, Vol.48, No.1, pp.27-34, 2002.

[16] Wang Bo, Wu Jie, Yang Jinming, and Zhao Shiwei, "A distributed hybrid wind-solar power control system based on genetic algorithm and hierarchical fuzzy control", *Proceedings of Fifth World Congress on Intelligent Control and Automation*, Volume 6, pp.5185-5188, 2004.

[17] Sang Yong Yang, Lac-Jeong Park, Cheol Hoon Park, and Jung Woong Ra, "A hybrid algorithm using genetic algorithm and gradient-based algorithm for iterative microwave inverse scattering", *Proceedings of IEEE International Conference on Evolutionary Computation*, Volume 1, p.450, 1995.

[18] M.Celino, P.Palazzari, N.Pucello, M.Rosati, and V.Rosato, "New parallel hybrid genetic algorithm based on molecular dynamics approach for energy minimization of atomistic systems", *Proceedings of IEEE International Conference on Evolutionary Computation*, pp.115-119, 1997.

[19] Cheon-Seok Park and Bong-Sik Jeong, "Reconstruction of a high contrast and large object by using the hybrid algorithm combining a Levenberg-Marquardt algorithm and a genetic algorithm", *IEEE Transactions on Magnetics*, Volume 35, Issue 3, Part 1, pp.1582-1585, 1999.

[20] Liu Jinlan, and Jian Ni, "A Hybrid Immune Genetic Algorithm Approach to Optimize the Integrated Forward/Reverse Logistics Network for 3PLs", *Proceedings of Third International Conference on Natural Computation*, Volume 3, pp.609-613, 2007.

[21] Jingjing Wu, Kelin Xu, Qinghua Kong, and Wenxian Jiang, "Application of the Hybrid Genetic Algorithm to Combinatorial Optimization Problems in Flow-shop Scheduling", *Proceedings of International Conference on Mechatronics and Automation*, pp. 1272-1277, 2007.

[22] Aihua Wang, and Weizi Yang, "Design of Energy Management Strategy in Hybrid Electric Vehicles by Evolutionary Fuzzy System Part II: Tuning Fuzzy Controller by Genetic Algorithms", *Proceedings of The Sixth World Congress on Intelligent Control and Automation*, Volume 2, pp.8329-8333, 2006.

[23] Decai Huang, Haidong Guo, and Neng Qian, "Hybrid Genetic Algorithm for Minimizing the Range of Lateness and Make-span on Non-identical Parallel Machines", *Proceedings of International Conference on Neural Networks and Brain*, Volume 1, pp.150-154, 2005.

[24] K.P.Wong, and Y.W.Wong, "Hybrid genetic/simulated annealing approach to short-term multiple-fuel-constrained generation scheduling", *IEEE Transactions on Power Systems*, Volume 12, Issue 2, pp.776-784, 1997.