HYBRID NEIGHBORHOOD CONTROL METHOD OF ADAPTIVE PLAN SYSTEM WITH GENETIC ALGORITHM

Pham Ngoc Hieu[1], **Hiroshi Hasegawa**[2]

[1]Graduated School of Engineering, Shibaura Institute of Technology, Japan
[2]College of Systems Engineering and Science, Shibaura Institute of Technology, Japan

m409421@shibaura-it.ac.jp(Pham Ngoc Hieu)

Abstract

With the aim to reduce a large amount of calculation cost and to improve the convergence to the optimal solution for multi-peak optimization problems with multi-dimensions, we propose a new method of Adaptive Plan System with Genetic Algorithm (APGA). This is an approach that combines the global search ability of Genetic Algorithm (GA) and Adaptive Plan (AP) for the local search ability. The APGA differs from GAs in handling design variable vectors (DVs). GAs generally encode DVs into genes and handle them through GA operators. However, the APGA encodes the control variable vectors (CVs) of AP, which searches for local optimum, into its genes. CVs determine the global behavior of AP, and DVs are handled by AP in the optimization process of APGA. The proposed strategy using Hybrid neighborhood control method is introduced into the APGA (H-APGA) to improve the convergence towards the optimal solution. The H-APGA is applied to some benchmark functions to evaluate its performance.

Keywords: Memetic Algorithms, Evolutionary Computation, Genetic Algorithms, Adaptive system, Global - Local search, Multi-peak problems.

Presenting Author's Biography

Pham Ngoc Hieu. He received the BE (2007) in Mechatronics from Hanoi University of Technology, Vietnam. Currently, he is a second year master student at Graduate School of Engineering, Shibaura Institute of Technology, Japan. He is PhD candidate of Shibaura Institute of Technology. His research interests include optimization design, intelligent and expert systems and bio-engineering.

1 Introduction

The product design is becoming more and more complex for various requirements from customers and claims. As a consequence, its design problem seems to be multi-peak problem with multi-dimensions. The Genetic Algorithm (GA) [1, 2] is the most emergent computing method has been applied to various multi-peak optimization problems. The validity of this method has been reported by many researchers [3, 4, 5]. However, it requires a huge computational cost to obtain stability in the convergence to an optimal solution. To reduce the cost and to improve stability, a strategy that combines global and local search methods becomes necessary. As for this strategy, current research has proposed various methods [6, 7, 8, 9]. For instance, Meta-Lamarckian learning [10] in Memetic Algorithms (MAs) [10, 11, 12, 13, 14] improves the search ability for multi-peak functions with multiple dimensions by introducing a human expert judgment, where local search methods are used. Additionally, in Fast Adaptive Memetic Algorithm (FAMA) [12], the coordination and choosing of the local search method is controlled by means of a measurement of fitness diversity over the individuals of the population dynamically.

On the other hand, Hasegawa et al. proposed a hybrid meta-heuristic method (HMH) by reflecting the recognition of dependence relations among design variables automatically, and reported the effectiveness of this method [15]. The HMH needs to switch from the Simulated Annealing (SA) [16] to the intuitive method - direct search using the learning result of the dependency of a DV - just before convergence to improve the local search ability of the optimal solution environs. To sum up, these methodologies need to choose suitably a best local search method from various local search methods for combining with a global search method within the optimization process. Furthermore, since genetic operators are employed for a global search method within these algorithms, DVs which are renewed via a local search are encoded into its genes many times at its GA process. These certainly have the potential to break its improved chromosomes via gene manipulation by GA operators, even if these approaches choose a proper survival strategy.

To solve these problems and maintain the stability of the convergence to an optimal solution for multi-peak optimization problems with multiple dimensions, Hasegawa proposed a new evolutionary algorithm (EA) called an Adaptive Plan system with Genetic Algorithm (APGA) [17].

In this paper, we purposed a Hybrid neighborhood control method of Adaptive Plan system with Genetic Algorithm (H-APGA) to converge to the optimal solution. This paper is organized in the following manner. Section 2 and 3 present the algorithm of the proposed strategy. Section 4 discussed about the convergence to the optimal solutions of multi-peak benchmark functions. Section 5 provides some brief conclusions.

2 The proposed strategy, APGA

2.1 Formulation of the optimization problem

The optimization problem is formulated in this section. Design variable, objective function and constrain condition are defined as follows:

$$Design\ variable : X = [x_1, \dots, x_n] \tag{1}$$

$$Objective\ function : -f(X) \rightarrow Max \tag{2}$$

$$Constrain\ condition : X^{LB} \leq X \leq X^{UB} \tag{3}$$

where $X^{LB} = \left[x_1^{LB}, \dots, x_n^{LB}\right]$, $X^{UB} = \left[x_1^{UB}, \dots, x_n^{UB}\right]$ and *n* denote the lower boundary condition vectors, the upper boundary condition vectors and the number of design variable vectors (DVs) respectively. A number of DV's significant figure is defined, and DV is rounded off its places within optimization process.

2.2 APGA

The APGA concept was introduced as a new EA strategy for multi-peak optimization problems. Its concept differs in handling DVs from general EAs based on GAs. EAs generally encode DVs into the genes of a chromosome, and handle them through GA operators. However, APGA completely separates DVs of global search and local search methods. It encodes Control variable vectors (CVs) of AP into its genes on Adaptive system (AS). Moreover, this separation strategy for DVs and chromosomes can solve MA problem of breaking chromosomes. The conceptual process of APGA is shown in Fig. 1. The control variable vectors (CVs) steer the behavior of adaptive plan (AP) for a global search, and are renewed via genetic operations by estimating fitness value. For a local search, AP generates next values of DVs by using CVs, response value vectors (RVs) and current values of DVs according to the formula:

$$X_{t+1} = X_t + NR_t \cdot AP(C_t, R_t) \tag{4}$$

where *NR, AP(), X, C, R, t* denote neighborhood ratio, a function of AP, DVs, CVs, RVs and generation, respectively. The APGA's algorithm is described by the pseudocode given in Fig. 2. In addition, for a verification of APGA search process, refer to ref. [17].

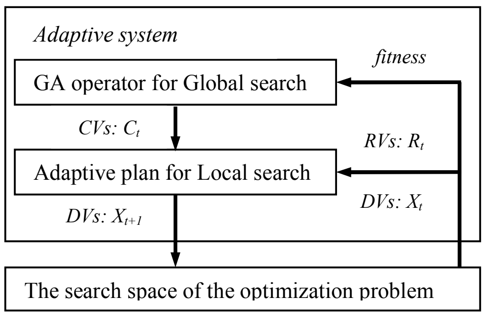

Fig. 1 Conceptual process of APGA

```
BEGIN
INITIALIZE population;
EVALUATE each individual;
REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
GENERATE DVs via AP for each individual;
EVALUATE individuals with DVs
SELECT parents;
RECOMBINE to produce offspring for CVs;
MUTATE offspring for CVs;
IF ( RESTRUCTURING CONDITION is satisfied )
RESTRUCTURE chromosome of offspring for CVs;
END DO
END
```

Fig. 2 Algorithm of APGA

2.3 Adaptive Plan (AP)

It is necessary that the AP realizes a local search process by applying various heuristics rules. In this paper, the plan introduces a DV generation formula using a sensitivity analysis that is effective in the convex function problem as a heuristic rule, because a multi-peak problem is combined of convex functions. This plan uses the following equation:

$$AP(C_t, R_t) = -Scale \cdot SP \cdot sign(\nabla R_t) \quad (5)$$

$$SP = 2C_t - 1 \quad (6)$$

where *Scale*, ∇R denote the scale factor and sensitivity of RVs, respectively.

A step size SP is defined by CVs for controlling a global behavior to prevent it falling into the local optimum. $C = [c_{i,j}, \ldots, c_{i,p}]$, $(0.0 \leq c_{i,j} \leq 1.0)$ is used by Eq. (6) so that it can change the direction to improve or worsen the objective function, and *C* is encoded into a chromosome by 10 bit strings (shown in Fig. 3). In addition, *i*, *j* and *p* are the individual number, design variable number and its size, respectively.

Handling of DV's out of range

DVs are renewed by AP, and when their values exceed their range, the APGA returns by Eq. (7) to their range.

$$\begin{cases} X_t = 2X^{LB} - X_t, \; X_t < X^{LB} \\ X_t = 2X^{UB} - X_t, \; X_t > X^{UB} \end{cases} \quad (7)$$

Coding into chromosome for CVs

CVs are individually coded into a string to form a chromosome. This 10 bits string with two values (0 and 1) represents a real value of CVs by using the procedure shown in Fig. 3. In addition, this figure shows both DVs and CVs of 2 dimensions cases.

2.4 GA operators

Selection

Selection is performed using a tournament strategy to maintain the diverseness of individuals with a goal of keeping off an early convergence. A tournament size of 2 is used.

Elite strategy

An elite strategy, where the best individual survives in the next generation, is adopted during each generation process.

It is necessary to assume that the best individual, i.e., as for the elite individual, generates two behaviors of AP by updating DVs with AP, not GA operators. Therefore, its strategy replicates the best individual to two elite individuals, and keeps them to next generation. As shown in Fig. 4, DVs of one of them (Δ symbol) is renewed by AP, and its CVs which are coded into chromosome are not changed by GA operators. Another one (◯ symbol) is that both DVs and CVs are not renewed, and are kept to next generation as an elite individual at the same search point.

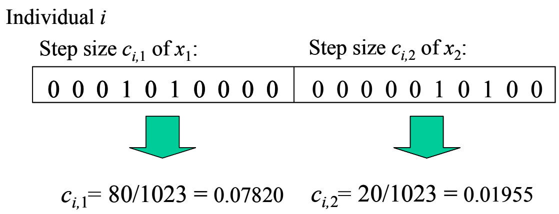

Fig. 3 Encoding into genes of a chromosome

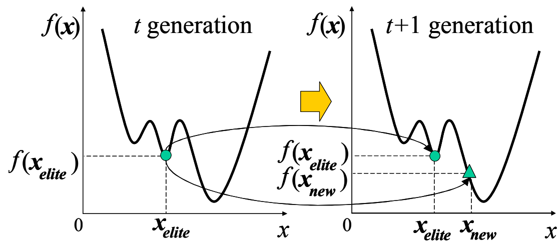

Fig. 4 Elite strategy

Crossover and mutation

In order to pick up the best values of each CV, a single point crossover is used for the string of each CV. This can be considered to be a uniform crossover for the string of the chromosome as shown in Fig. 5(a).

Mutation are performed for each string at mutation ratio on each generation, and set to maintain the strings diverse as shown in Fig. 5(b).

Recombination of genes

At following conditions, the genetic information on chromosome of individual is recombined by uniform random function.

(1) One fitness value occupies 80% of the fitness of all individuals.

(2) One chromosome occupies 80% of the population.

If this manipulation is applied to general GAs, an improved chromosome into which DVs have been encoded is broken down. However, in the APGA, the

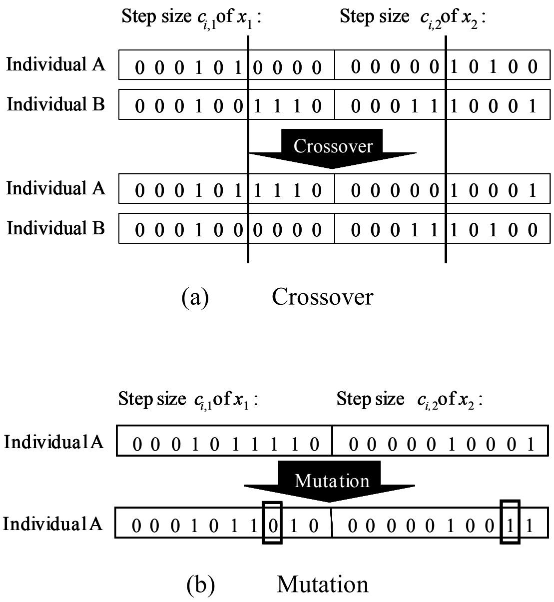

Fig. 5 Example of crossover and mutation

genetic information is only CVs used to make a decision for the AP behavior. Therefore, to prevent from falling into a local optimum, and to get out from the condition of being converged with a local optimum, a new AP behavior is provided by recombining the genes of the CVs into a chromosome. And the optimal search process starts to re-explore by a new one. This strategy is believed to make behavior like the re-annealing of an SA.

3 Hybrid Neighborhood Control method

In the multi-point search of APGA, individuals move from their various points to the new ones in the design space of DVs. For example, as shown in Fig. 6, individual A requires a slight change to the value of the DVs to obtain the global optimum solution. On the other hand, individual B cannot reach a global optimum solution without a significant change. In addition, individual C has landed in a local optimum solution. Such a situation, in which the individual are intermingled, can generally occur at any time in search process. Therefore, it is necessary to find a suitable DV generation process for the situation of each individual in the design space.

In this paper, to improve the multi-point search capability of APGA, we propose a hybrid neighborhood control method by mixing between linear function and exponential function. Additionally, applying a neighborhood control is a common approach for an SA, and the search point can be changed by controlling the neighborhood range. Therefore, the purposed method is introduced to the AS. The method automatically adapts the neighborhood range to obtain DV generation accuracy for the situation of each individual. As a result, we believe that it will steadily provide a global optimum solution and reduce the calculation cost.

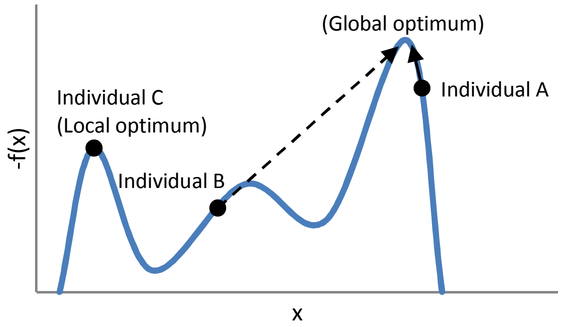

Fig. 6 Example of individual situations

3.1 Linear method (method 1)

The neighborhood range is determined by the *NR*. The assignation step of method 1 is shown as Fig. 7. The formula for *NR* by linear function is as follows:

$$NR_t = 1 - \frac{1}{2}\left(\frac{inv}{\frac{individual}{2}}\right) \tag{8}$$

where *inv*, *individual* denote the current individual and number of individual.

To calculate *NR*, this method uses individual number only. Therefore, *NR* is distributed to an individual at random. It is adjusted to wire range with the first half of individual number and it is modified to small range with the last haft of one.

3.2 Mixed methods (method 2 and 3)

Method 2 and 3 sort all of individuals by estimating their fitness then ranking them by results. The rank is assigned *NR* that corresponds with this number. In these methods, with one haft of individual number, *NR* is adjusted by following sigmoid function. Another half one is determined by linear function.

$$NR_t = \frac{1}{\left\{1 + exp\left(\beta . \frac{rank - \frac{individual}{2}}{individual}\right)\right\}} \tag{9}$$

where β, *rank* denote the gain of the sigmoid function and rank number. The search process is varied according to these methods toward the best search direction for all of individuals.

In method 2, as shown in Fig. 8, the good individuals (high fitness values) are allocated large *NR* values to search global area in the design space of the DVs following Eq. (8). On the other hand, the low individuals (low fitness values) are allocated small *NR* values to perform a local search efficiently following Eq. (9).

Method 3 employs the inverse method of method 2 (as shown in Fig. 8). Method 3 assigns a large *NR* to the good individuals following Eq. (9). Moreover, as for the low individuals, this method assigns a small *NR* following Eq. (8). From these handlings, to assign larger *NR* than method 2, individuals can widely move to new points in the design space of DVs.

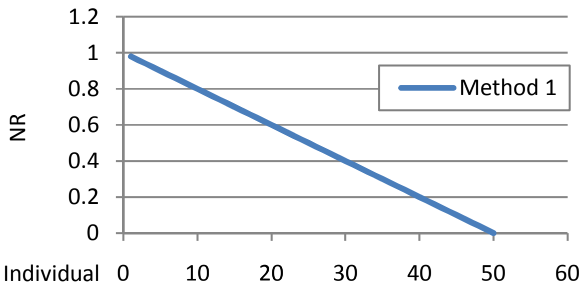

Fig. 7 The relationship of NR and individual

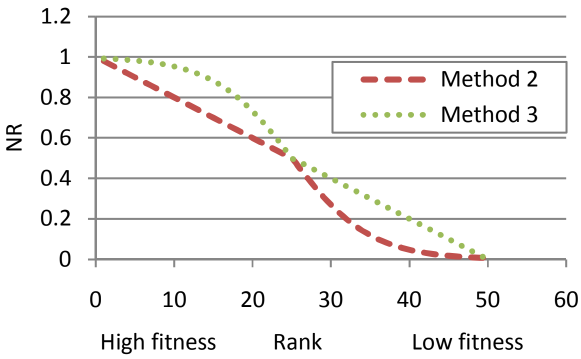

Fig. 8 The relationship of NR and rank, when $\beta = 10$

4 Numerical experiments

In this section, the numerical experiments are first performed to assign the gain β of the sigmoid function (method 2 and 3) and to compare among methods. Next, these methods are compared with other methodologies for the robustness of the optimization process. These experiments are performed 20 trials for every function. The initial seed number is randomly varied during every trial. In each experiment, the GA parameters used in solving benchmark functions are set as follows: selection ratio, crossover ratio and mutation ratio are 1.0, 0.8 and 0.01 respectively. The population size is 50 individuals and the terminal generation is 5000th generation. The sensitivity plan parameters in Eq. (5) are listed in Tab. 3.

4.1 Benchmark functions

For the H-APGA, we estimate the stability of the convergence to the optimal solution by using three benchmark functions with 20 dimensions Rastrigin (RA), Griewank (GR) and Rosenbrock (RO) function. These functions are given as follows:

$$RA = 10n + \sum_{i=1}^{n} \{x_i^2 - 10cos(2\pi x_i)\} \tag{10}$$

$$GR = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) \tag{11}$$

$$RO = \sum_{i=1}^{n} 100(x_{i+1} + 1 - (x_i + 1)^2)^2 + x_i^2 \tag{12}$$

Tab. 1 shows their characteristics, and the terms epistasis, multi-peak, steep denote the dependence relation of the DVs, presence of multi-peak and level of steepness, respectively. All functions are minimized to zero, when optimal DVs $X = 0$ are obtained. Moreover, it is difficult to search for their optimal solutions by applying one optimization strategy only, because each function has a different complicated characteristic. In Tab. 2, their design range, the digits of DVs are summarized. If the search point attains an optimal solution or a current generation process reaches the termination generation, the search process is terminated.

Tab. 1 Characteristics of benchmark functions

Function	Epistasis	Multi-peak	Steep
RA	No	Yes	Average
GR	Yes	Yes	Small
RO	Yes	No	Big

Tab. 2 Design range, digits of DVs

Function	Design range	Number of digits
RA	No	2
GR	Yes	1
RO	Yes	3

Tab. 3 Scale factor for normalizing the benchmark functions

Function	Scale factor
RA	10.0
GR	100.0
RO	4.0

4.2 Experiment results

The experiment results are shown in Tab. 4, Tab. 5 and Tab. 6. The success ratio of all benchmark functions is 100% with small computation cost. And the improvement rate (Fig. 12) is average value of improvement rate of generation number compared with the average generation of Simple APGA [17].

The solutions of all benchmark functions reach their global optimum solutions. However, there are some differences among methods. Method 1 converged faster than method 2, and this is really good with RA function. Method 3 could arrive at a global optimum at a high probability with every function.

As a result, we assign the best trial that is found by the maximum improvement rate. Its best gain value is trial 20 with method 1, trial 19 with method 2 and trial 17 with method 3. Moreover, the results using the best trial of these methods are compared. From this comparison, we can confirm that method 3 is the best solution.

Next, Fig. 9, Fig. 10 and Fig. 11 shows diagrams for the average fitness of individual until these methods reach global optimum solutions, in the numerical experiment again to confirm above mentioned result.

The result of testing by the values of gain β with RO function is shown in Fig. 13. From this comparison, we can confirm that method 3 converged faster than method 2, and the best gain ranges from 6 to 14.

To sum up, its validity confirms that this strategy can reduce the computation cost and improve the stability of the convergence to the optimal solution.

Tab. 4 Experiment result by method 1

Trial	Function			Improvement Rate
	RA	GR	RO	(%)
1	178	355	1256	62
2	175	366	1216	62
3	177	321	1563	58
4	214	412	1564	72
5	209	321	1473	62
6	208	376	1382	68
7	218	374	1443	69
8	217	355	1201	67
9	206	385	1350	69
10	181	321	1057	58
11	211	297	1578	60
12	208	302	1387	60
13	197	336	1408	62
14	213	396	1597	71
15	183	413	1262	68
16	199	394	1503	69
17	201	302	1304	59
18	202	427	1476	72
19	194	410	1506	70
20	*160*	*314*	*1538*	*55*
Average	198	359	1404	65

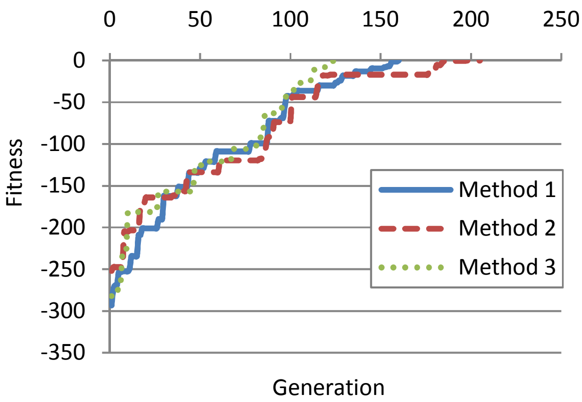

Fig. 9 Individual's fitness with RA function

4.3 Comparison

Method 1, 2 and 3 was compared with two other methods - Simple APGA [17] and APGA/VNC [18]. The comparison among methods is shown in Fig. 14.

These methods were better than Simple APGA in all of benchmark functions. Therefore, it is desirable to introduce these methods into APGA.

Tab. 5 Experiment result by method 2

Trial	Function			Improvement Rate
	RA	GR	RO	(%)
1	219	384	1792	70
2	220	429	1456	75
3	206	429	1779	73
4	166	377	1466	62
5	218	305	1863	62
6	216	387	1664	70
7	194	421	1693	71
8	217	448	1479	77
9	202	402	1800	70
10	205	471	1072	78
11	217	301	1825	61
12	193	472	1851	76
13	212	459	1118	77
14	213	360	1574	67
15	218	352	1873	67
16	195	409	1612	70
17	207	403	1810	71
18	210	403	1303	71
19	*205*	*232*	*1878*	*52*
20	217	337	1443	65
Average	208	390	1618	69

Tab. 6 Experiment result by method 3

Trial	Function			Improvement Rate
	RA	GR	RO	(%)
1	209	276	1049	57
2	172	307	1108	56
3	212	327	1162	63
4	207	274	835	57
5	180	305	1057	57
6	200	298	1184	58
7	214	299	1134	60
8	167	311	1171	55
9	211	319	1016	62
10	205	275	1212	57
11	190	323	1076	60
12	207	239	1061	53
13	178	305	1045	56
14	206	316	976	61
15	201	311	1022	60
16	205	327	1191	62
17	*125*	*306*	*1156*	*49*
18	210	294	1116	59
19	201	242	1190	53
20	208	293	999	59
Average	196	298	1088	58

In particular, it was confirmed that the calculation cost with these methods could be reduced for benchmark functions. And it showed that the convergence to the optimal solution could be improved more significantly.

Method 3 had better convergence than APGA/VNC method with RA function and GR function, however it did not gain a high probability with RO function.

In summary, from the result shown in Tab. 7, we employed method 3 for H-APGA model.

Overall, the H-APGA was capable of attaining robustness, high quality, low calculation cost and efficient performance on many benchmark problems.

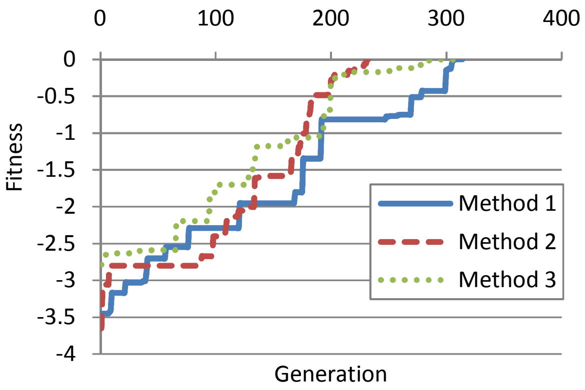

Fig. 10 Individual's fitness with GR function

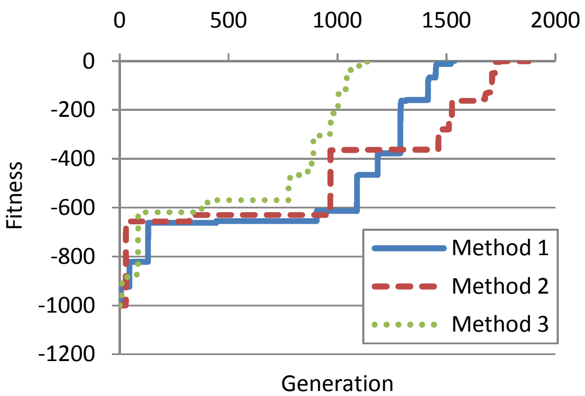

Fig. 11 Individual's fitness with RO function

5 Conclusion

In this paper, H-APGA method has been proposed to solve the multi-peak optimization problems with multi-dimensions.

The H-APGA was applied to three benchmark functions to evaluate its performance. Moreover it was compared with simple APGA. As a result, we can confirm that the H-APGA reduces the calculation cost and improves the convergence to the optimal solution.

Next, the H-APGA was compared with APGA/VNC method, and it was confirmed that it could be better with some benchmark functions.

About the optimal solution such as minimum time and maximum reliability, it is a future work.

Finally, this study plans to do a comparison with the sensitivity plan of the AP by applying other optimization methods to the AP and optimizing the benchmark functions.

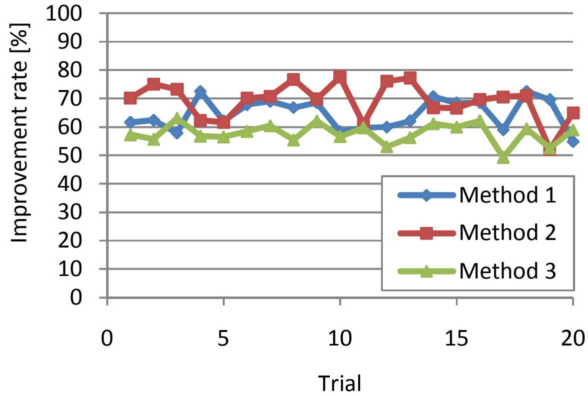

Fig. 12 The improvement rate though trials

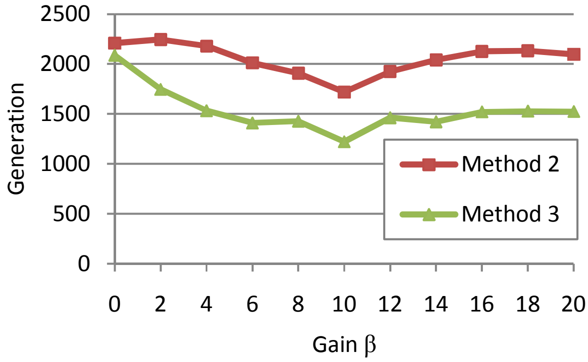

Fig. 13 Gain β with RO function

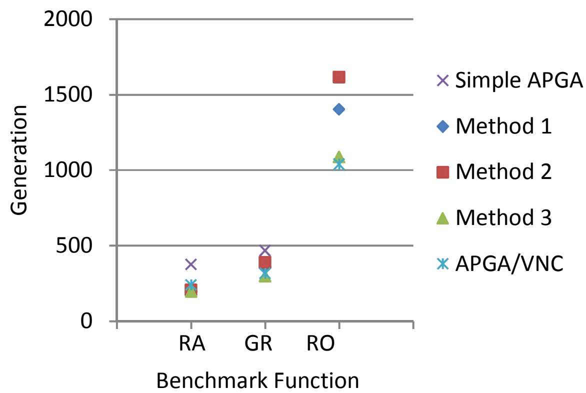

Fig. 14 Comparison among methods

Tab. 7 Comparison of Method 3 and APGA/VNC

Function	Method 3	APGA/VNC
RA	O	X
GR	O	X
RO	X	O
Sum of better	2	1
O - better result; X - worse result		

6 References

[1] David E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-

Wesley, 1989.

[2] Charles L. Karr and L. Michael Freeman. *Industrial applications of genetic algorithms*. CRC Press LLC, 1999.

[3] J. D. Digalakis and K. G. Margaritis. An experiment study of benchmarking functions for genetic algorithms. *Proceedings of IEEE Conference on Transactions*, 5:3810–3815, 2000.

[4] J. Sakuma and S. Kobayashi. Extrapolation-directed crossover for real-coded ga: Overcoming deceptive phenomena by extrapolative search. *Proceedings of Congress on Evolutionary Computation (CEC 2001)*, pages 685–662, 2001.

[5] X. Li and M. Kirley. The effects of varying population density in a fine-grained parallel genetic algorithm. *Proceedings of Congress on Evolutionary Computation (CEC 2002)*, 2:1709–1714, 2002.

[6] S. W. Mahfoud and D. E. Goldberg. A genetic algorithm for parallel simulated annealing. *Parallel Problem Solving from Nature*, 2:301–310, 1992.

[7] David E. Goldberg and Siegfried Voessner. Optimizing global local search hybrids. *IlliGAL report no. 99001*, January 1999.

[8] M. Miki T. Hiroyasu and M. Ogura. Parallel simulated annealing using genetic crossover. *Proceedings of the ISCA 13th International Conference on PDCS-2000*, 2000.

[9] T. Hiroyasu M. Miki and T. Fushimi. Parallel simulated annealing with adaptive neighborhood determined by ga. *IEEE International Conference on System, Man and Cybernetics*, 2003.

[10] Y. S. Ong and A. J. Keane. Meta-lamarckian learning in memetic algorithms. *IEEE transactions on evolutionary computation*, 8(2):99–110, 2004.

[11] J.E. Smith William E. Hart, N. Krasnogor. *Recent Advances in Memetic Algorithms*. Springer, 2005.

[12] F. Neri N. Salvatore A. Caponio, G. L. Cascella and M. Sumner. A fast adaptive memetic algorithm for online and offline control design of pmsm drives. *IEEE transactions on Systems, Man and Cybernetics Part B, Special Issue on Memetic Algorithms*, 37(1):28–41, 2007.

[13] T. Krkkinen F. Neri, V. Tirronen and T. Rossi. Fitness diversity based adaptation in multimeme algorithms: A comparative study. *IEEE transactions on Systems, Man and Cybernetics Part B*, 36(1):141–152, 2006.

[14] T. Krkkinen K. Majava V. Tirronen, F. Neri and T. Rossi. An enhanced memetic differential evolution in filter design for defect detection in paper production. *Evolutionary Computation Journal, MIT Press*, 16(4):529–555, 2008.

[15] H. Uehara H. Hasegawa, H. Sasaki and K. Kawamo. The optimization of spot-weld positions for vehicle design by using hybrid meta-heuristics. *International Journal of Vehicle Design*, 43(1-4):151–172, 2007.

[16] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical Science*, Vol.8, No.1:10–15, 1993.

[17] H. Hasegawa. Adaptive plan system with genetic algorithm based on synthesis of local and global search method for multi-peak optimization problems. *Proceedings of the 6th EUROSIM Congress on Modelling and Simulation*, 2007.

[18] H. Hasegawa Sousuke Tooyama. Adaptive plan system with genetic algorithm using the variable neighborhood range control. *IEEE Congress on Evolutionary Computation (CEC 2009)*, pages 846–853, 2009.