ADAPTIVE PLAN SYSTEM WITH GENETIC ALGORITHM BASED ON SYNTHESIS OF LOCAL AND GLOBAL SEARCH METHOD FOR MULTI-PEAK OPTIMIZATION PROBLEMS

Hiroshi Hasegawa[1]

[1]Shibaura Institute of Technology, Faculty of Systems Engineering,
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama, 337-8570, Japan

h-hase@sic.shibaura-it.ac.jp (Hiroshi Hasegawa)

Abstract

The product design is becoming more and more complex for various requirements from customer needs and claims. As a consequence, its design problem seems to be a multi-peak problem with multiple dimensions. It is necessary to solve the multi-peak optimization problems. In this paper, a new evolutionary algorithm (EA) has been proposed to solve the multi-peak optimization problems with multiple dimensions. The proposed algorithm is a stochastic global search heuristics in which EAs-based approaches are combined with local search method. It has been named by author the Adaptive Plan system with Genetic Algorithm (APGA) in GA-based combination algorithms. APGA uses the Adaptive Plan (AP) to control its optimization process for the local search process. APGA differs in handling design variable vectors (DVs) from GAs. GAs encode DVs into genes, and handle them through GA operators. However, APGA encodes control variable vectors (CVs) of AP, which searches to local minima, into its genes. CVs decide on a global behavior of AP, and DVs are handled by AP in the optimization process of APGA. APGA is applied to many benchmark functions to evaluate its performance, and has been confirmed to improve the calculation cost and the stability of convergence towards the optimal solution.

Keywords: Memetic Algorithms, Evolutionary Computation, Genetic Algorithms, Local search method, Multi-peak problems.

Presenting Author's biography

Hiroshi Hasegawa. He received his BE and ME from Shibaura Institute of Technology, Japan, in 1992 and 1994, respectively. He received his PhD in Mechanical Engineering from Tokyo Institute of Technology, Japan, in 1998. He worked for CRC Research Institute Inc. from April 1994 to March 2004. From April 2004, he has been working at Shibaura Institute of Technology, and is currently an Associate Professor. He is member of Japan Society for Simulation Technology, Japan Society of Mechanical Engineers and American Society of Mechanical Engineers. His research interests include computer-aided exploration, especially multi-peak optimization, robust design, multi-disciplinary optimization, shape and topology optimization.

1 Introduction

The product design is becoming more and more complex for various requirements from customer needs and claims. As a consequence, its design problem seems to be multi-peak problem with multiple dimensions. It is necessary to solve the multi-peak optimization problems. Genetic Algorithm (GA) [1, 2] is the most prominent emergent computing method, which has been applied to various multi-peak optimization problems. Until now, the validity of the method has been reported by many researchers. Digalakis and Margaritis presented a review and experimental results on the major benchmark functions which are used for performance and control of GA [3]. In [3], it is suggested that an optimal population size is 200-250 individuals. Sakuma and Kobayashi have proposed a real-coded GA using the combination of two crossovers—UNDX-m and EDX [4]—where the effect are also reported through benchmark functions. Li and Kirley have introduced fine-grained parallel GA, which is derived from percolation theory [5]. Author group reported the efficiency of GA-based approaches through the studies—topology optimization of the mechanical structure [6, 7, 8], development of General-purposed Optimization Engine (GOE) and application of GOE to various optimal designs [9, 10].

However, Emergent computing methods require huge calculation costs for obtaining stability in the convergence to the optimal solution. To reduce the cost and improve stability, a strategy becomes necessary, which combines global and local search methods. As for this strategy, current research has proposed various methods. The popular approach is the combination of global search ability of GAs with the local search ability of Simulated Annealing (SA) [11]. As a pioneering research, Mahfoud and Goldberg proposed Parallel Recombinative Simulated Annealing (PRSA) applied SA to a selection strategy of GA [12]. Later, Uehara et al. have introduced metropolis loop process of SA to an elite strategy in GA process [13, 14]. Hiroyasu et al. proposed Parallel SA using Genetic Crossover (PSA/GAc) [15] and Miki et al. proposed parallel SA with adaptive neighborhood range determined by GA (PSA/ANGA) [16]. These hybrid methods have been applied to major benchmark functions and have reported to be valid. It is believed to be both locally and globally efficient. However, major multi-peak benchmark functions of multiple dimensions, i.e., Rastrigin (RA) and Griewank (GR) functions, require about 10^6 function calls for arriving at the optimal solution. Furthermore, GR function with multiple dimensions cannot achieve the optimal solution. Generally, when the optimal problem exhibits a dependence on design variable vectors (DVs) and the steepness of the objective function is small in the feasible space of DVs, it is difficult to obtain an optimal solution.

To overcome these difficulties, Ong and Keane have proposed meta-Lamarckian learning [17] in Memetic Algorithms (MA) [18]. This method improves search ability for multi-peak problems with multiple dimensions by introducing a human expert judgment where local search methods are to be used. However, this method has needed a manual operation in switching and choosing of global and local search method. Hasegawa et al. have proposed a hybrid meta-heuristic method by reflecting recognition of dependence relation among design variables automatically and have reported the efficiency of it [19, 20]. However, it is difficult to control the optimization process in switching and choosing of global and local search method.

In this paper, a new evolutionary algorithm (EA)—Adaptive Plan system with Genetic Algorithm (APGA)—is proposed to solve the multi-peak optimization problems with multiple dimensions. The paper is organized in the following manner. Section 2 presents the algorithm of the proposed strategy. Section 3 discusses the convergence to optimal solutions of multi-peak benchmark functions. Section 4 provides some brief conclusions.

2 The proposed algorithm, APGA

2.1 Formulation of the optimization problem

The optimization problem is formulated in this section. DVs, objective function and range of DVs are defined as follows:

$$\text{DVs:} \qquad X = \left[x_1, \ldots, x_n\right], \tag{1}$$

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

$$\text{Range of DVs:} \qquad X^{LB} \leq X \leq X^{UB}\,, \tag{3}$$

where $X^{LB} = \left[x_1^{LB}, \ldots, x_n^{LB}\right]$, $X^{UB} = \left[x_1^{UB}, \ldots, x_n^{UB}\right]$ and n denote the lower boundary condition vector, the upper boundary condition vector and number of DVs respectively. DVs are defined by x_i and are represented by the vector X.

2.2 APGA

APGA is developed for overcoming difficulty in controlling—switching, choosing and steering—a combination of global and local search method.

In MA which fits best with Dawkin's concept of meme, a more general form can be described by the pseudocode given in Fig. 1 [18]. The local search method is applied to one or more phases of improvement to individuals of the population within each generation of EA. These improvement phases intend to search at neighborhood area of DVs are coded into chromosome of offspring, and improved DVs are encoded into its genes again. This combination approach has the potential to break improved its chromosome via manipulating genes by

GA operators. As a matter of course, EAs of Ref. [17, 19] have this risk within each generation. Moreover, it is difficult to decide what method to use in which timing within a global search method, because DVs of global and local search methods affect one another through the chromosome.

```
Begin
  INITIALIZE population;
  EVALUATE each individual;
  Repeat Until ( TERMINATION CONDITION is satisfied ) Do
    SELECT parents;
    RECOMBINE to produce offspring;
    EVALUATE offspring;
    IMPROVE offspring via Local Search;
    MUTATE offspring;
    IMPROVE offspring via Local Search;
    SELECT individuals for next generation;
  end DO
End
```

Fig. 1 Pseudocode for a simple MA [18]

On the other hand, Natural and artificial systems adapt the behavior of themselves to the changing global and social environments over generations. These systems have been defined Adaptive System (AS) by Holland in [1], and AS has an adaptive plan (AP) which decides on its behavior through response to environments (ENV).

APGA is introduced its concept to a new EA strategy for multi-peak optimization problems. Hence, this study considers a global search method as AS, a local search method as AP and an optimization space of DVs as ENV, respectively. The conceptual strategy of APGA is shown in Fig. 2. APGA differs in handling DVs from general EAs based on GAs to overcome theses difficult problems. EAs generally encode DVs into genes of chromosome, and handle them through GA operators. However, APGA separates DVs of global search and local search methods completely. It encodes control variable vectors (CVs) of AP, which searches to local minima, into its genes on AS. CVs decide on a global behavior of AP, and DVs are handled by AP in the optimization process of APGA. The generation process of DVs is shown in Fig. 3. This process generates a new DVs X_{t+1} from current search point X_t according to the following formula:

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

where $AP(\ )$, C, R and t denote a function of AP, CVs, response value vectors (RVs) and generation, respectively. AS is controlled by the behavior of $AP(\ )$ via feedback loop of fitness value f of R from a function of problem (ENV) during the global search process. Moreover, C can be renewed by estimating f by using the GA operators within this process, and their trends are believed to make optimal behaviors like a cooling temperature of SA.

The APGA's algorithm can be described by the pseudocode given in Fig. 4. It can be understood that this is a stochastic global search heuristics in which EAs-based approaches are combined with local search method.

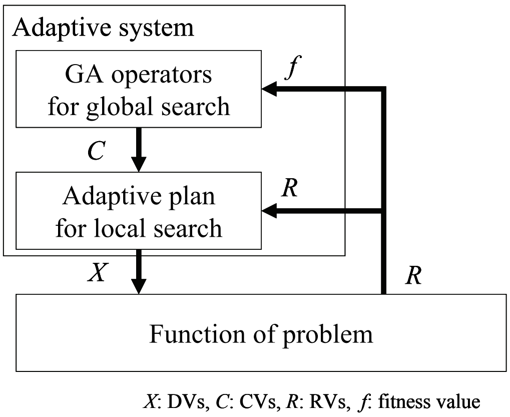

Adaptive system
GA operators for global search
C
Adaptive plan for local search
f
R
X
R
Function of problem
X: DVs, C: CVs, R: RVs, f: fitness value

Fig. 2 Conceptual strategy of APGA

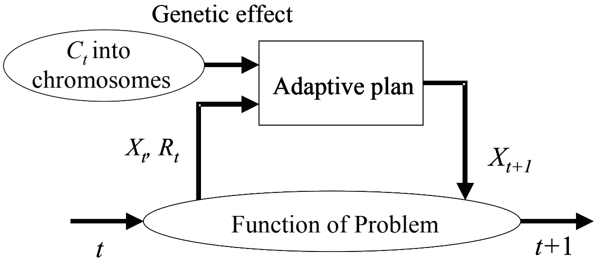


Fig. 3 Generation process of DVs

```
Begin
  INITIALIZE population;
  EVALUATE individuals with initial DVs;
  Repeat Until ( TERMINATION CONDITION is satisfied ) Do
    GENERATE DVs via AP for each individual;
    EVALUATE individuals with DVs;
    SELECT parents;
    RECOMBINE to produce offspring;
    MUTATE offspring;
    IF ( RESTRUCTURING CONDITION is satisfied)
      RESTRUCTE chromosome of offspring;
  end DO
End
```

Fig. 4 Algorithm of APGA

2.3 Adaptive plan (AP): sensitivity plan

It is necessary that AP realizes a local search process by applying various heuristics rules. In this paper, this plan introduces a generation formula of DVs of steepest descent method, which is effective in the convex function problem as a heuristics rule, because a multi-peak problem is combined by convex functions. This plan uses the following equation.

$$AP(C_t, R_t) = -NR_0 \cdot Scale \cdot SP \cdot sign(\nabla R_t), \qquad (5)$$

$$SP = 2C_t - 1,\ 0.0 \le c_{i,j} \le 1.0\,, \qquad (6)$$

where NR_0, *Scale* and ∇R denote initial neighbourhood ratio, scale factor and sensitivity of RVs, respectively. A step size *SP* is defined by CVs for controlling a global behavior to prevent from falling into the local optima. $C=[c_{i,j}, \ldots, c_{i,p}]$ is used by Eq. (6) so that it can change the direction to improve or worsen the objective function, and *C* is encoded into chromosome by 10 bit strings (Fig. 5). In addition, *i*, *j* and *p* are individual number, population number and population size.

2.4 Ingenuities for DVs and CVs

2.4.1 Handling of significant figures

In the optimal design of the product design, dimensions of products can be mainly dealt with as DVs. These are always assigned dimensional accuracies on a mechanical drawing. Therefore, a value of DVs is done well to use a number of significant figures of assigned dimensional accuracy in its drawing in the optimal process. In APGA, a number of significant figures of DVs are defined, and DVs are truncated to it within optimal process.

2.4.2 Handling of DVs's out of range

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

$$\left.\begin{array}{l} \textit{if}\ \ X_t < X^{LB}\ \ \textit{then}\ \ X_t = X^{LB} \\ \textit{if}\ \ X_t > X^{UB}\ \ \textit{then}\ \ X_t = X^{UB} \end{array}\right\}. \qquad (7)$$

2.4.3 Coding into chromosome for CVs

CVs are individually coded into a string as chromosome. This 10 bit strings with two values—0 and 1—represents a real value of CVs by using procedure of Fig. 5. In addition, Fig. 5 shows DVs and CVs of two dimensional cases.

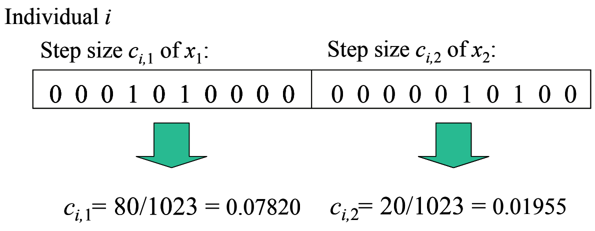

Individual *i*
Step size $c_{i,1}$ of x_1:
Step size $c_{i,2}$ of x_2:
0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 1 0 1 0 0
$c_{i,1}$= 80/1023 = 0.07820
$c_{i,2}$= 20/1023 = 0.01955

Fig. 5 Encoding into genes of chromosome

2.5 GA operators

2.5.1 Selection

Selection is performed using tournament strategy to keep a diverseness of individuals at early generations. The tournament size of 2 is used.

2.5.2 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 global behaviors of AP by updating DVs with AP, not GA. Therefore, its strategy replicates the best individual to two elite individuals, and keeps them to the next generation. As shown in Fig. 6, DVs of one of them (▲ symbol) is renewed by AP, and its CVs which are coded into chromosome are not changed by GA operators. Another one (● symbol) means that both DVs and CVs are not renewed, and are kept to next generation as an elite individual at the same search point.

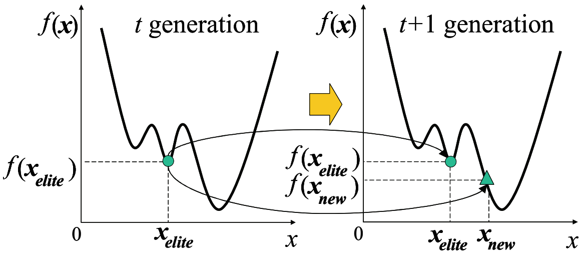


Fig. 6 Elite strategy

2.5.3 Crossover and mutation

In order to pick up the best values of each CV, a single-point crossover is used for string of each CV. This can be considered to be a uniform crossover for the string of chromosome as shown in Fig. 7. The mutation is adapted for string of each CV, and its method reverses the 1 bit in its string.

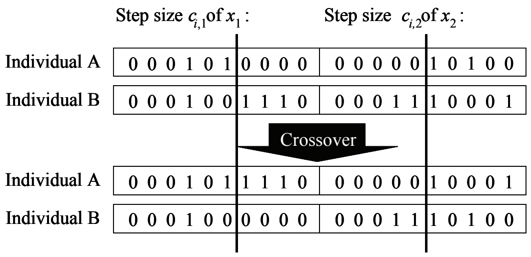


Fig. 7 Example of crossover

2.5.4 Recombination of gene

At following conditions, the genetic information on chromosome of individual is recombined by a 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 for general GAs, improved chromosome which DVs have been encoded into is broken down. However, in APGA, genetic information is only CVs to make a decision of a behavior of AP. Therefore, to prevent from falling into a local optima, and to get out from the condition converged with a local optima, the optimal search process can recalculate by recombination genes of

CVs into chromosome. This strategy is believed to make behavior like a reannealing of SA.

3 Numerical experiments

3.1 Benchmark functions

APGA estimates the stability of convergence to the optimal solution by using eight benchmark functions—RA, Ridge (RI), GR, Ackley (AC), Schwefel's Problem2.22 (SC), Sphere (SP), Schaffer F6 (SF) and Rosenbrock (RO) functions—in the numerical experiments. These functions are given as follows:

$$\text{RA: } F_0 = 10n + \sum_{i=1}^{n} \left\{ x_i^2 - 10\cos(2\pi x_i) \right\}, \tag{8}$$

$$\text{RI: } F_1 = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2, \tag{9}$$

$$\text{GR: } F_2 = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right), \tag{10}$$

$$\begin{aligned} \text{AC: } F_3 = -20\exp&\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \right) \\ &- \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos 2\pi x_i \right) \\ &+ 20 + e, \end{aligned} \tag{11}$$

$$\text{SC: } F_4 = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|, \tag{12}$$

$$\text{SP: } F_5 = \sum_{i=1}^{n} x_i^2, \tag{13}$$

$$\text{SF: } F_6 = 0.5 + \frac{\left(\sin\sqrt{\sum_{i=1}^{n} x_i^2} \right)^2 - 0.5}{\left\{ 1.0 + 0.001\left(\sum_{i=1}^{n} x_i^2 \right) \right\}^2}, \tag{14}$$

$$\text{RO: } F_7 = \sum_{i=1}^{n-1} 100\left(x_{i+1} + 1 - (x_i + 1)^2 \right)^2 + x_i^2. \tag{15}$$

Tab. 1 shows characteristics of them, and epistasis, multi-peak and steep denote a dependence relation of DVs, a presence of multi-peak and a level of steepness, respectively. All functions are minimized to $F_k = 0$ ($k = 0, \ldots, 7$), when optimal DVs $X = 0$ are obtained. Moreover, to search for their optimal solutions is difficult by applying only one optimization strategy, because each function has a different complicated characteristic. In Tab 2, their range and digit of DVs are summarized. If the search point attains an optimal solution or a current generation process reaches the terminal generation, the search process is terminated.

Tab. 1 Characteristics of benchmark functions

	Characteristics		
Function	Epistasis	Multi-peak	Steep
0	No	Yes	Average
1	Yes	No	Average
2	Yes	Yes	Small
3	No	Yes	Average
4	No	No	Average
5	No	No	Average
6	No	Yes	Small
7	Yes	No	Big

Tab. 2 Range and digit of DVs

Function	Design range	Number of digits
0	$-5.12 \le X \le 5.12$	2
1	$-51.2 \le X \le 51.2$	1
2	$-51.2 \le X \le 51.2$	1
3	$-5.12 \le X \le 5.12$	2
4	$-51.2 \le X \le 51.2$	1
5	$-5.12 \le X \le 5.12$	2
6	$-51.2 \le X \le 51.2$	1
7	$-2.048 \le X \le 2.048$	3

3.2 Discussion of results

In this section, the numerical experiment is performed by benchmark functions. These experiments are performed 5 times for every function. The initial seed number which is used by APGA is randomly varied during every trial. In each experiment, the GA parameters used in solving the benchmark functions are set as follows: the population size is 50 individuals, selection ratio is 1.0, crossover ratio is 0.8, mutation ratio is 0.01 and terminal generation is the 1,000th generation. Sensitivity plan parameters in Eq. (5) are listed in Tab. 3.

Tab. 3 Neighbourhood ratio *NR* and scale factor

Function	NR$_0$	Scale
0	1.0	1.0
1	1.0	10.0
2	1.0	10.0
3	1.0	1.0
4	1.0	1.0
5	1.0	1.0
6	1.0	10.0
7	1.0	0.1

3.2.1 Verification of search process

To handle the AP through CVs by GA operators of AS, is verified in this section. This verification uses function 0 and 2—RA and GR functions. These functions have multiple peaks, and it is known that their optimization is difficult for GA. This experiment uses one dimensional DV, to assess the detail of their search process behavior. Moreover, scale factors of AP are set to 10.0 and 102.4 to function 0 and 2 so that each individual can be moved in all search area. In addition, population size is made the minimum number, 10 individuals. The optimal results of function 0 and 2 are shown in Fig. 8 and 9. A landscape of function, elite individual and individuals of each generation are shown on their figures. These results are the optimal solution.

In the case of function 0 (Fig. 8), individuals are spread out all over the search area through the early generations (1st – 30th). At the late generations (40th – 53rd), they are concentrated on the peak which the optimal point belongs to, and an individual finds the optimal point at 53rd generation. As shown in Fig. 9 (function 2), individuals are spread out all over the search area at all generations. It is thought this cause is that the steepness of the function 2 is small. However, elite individual certainly searches the design space, and reaches the optimal point at 35th generation. Next, behaviors of these elite individuals are shown in Fig. 10 and 11. Function 0 and 2 are obtained the optimal solution by the elite individual through searching a multiple peak in the order of number from 1 to 6 and from 1 to 7, respectively.

Based on the experimental results, It could be confirmed that APGA can optimize a multi-peak problem by handling AP which takes on a local search.

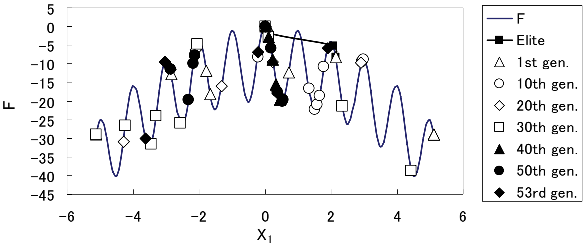


Fig. 8 Search process to the function 0

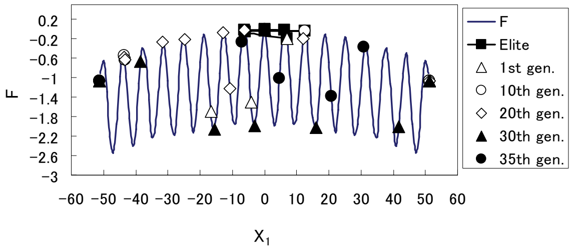


Fig. 9 Search process to function 1

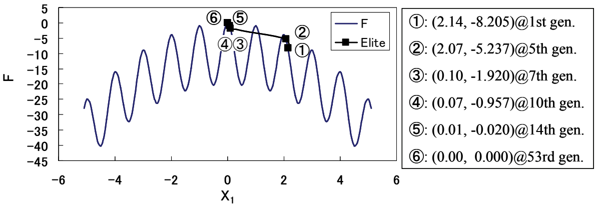


Fig. 10 Search process of elite individual to the function 0

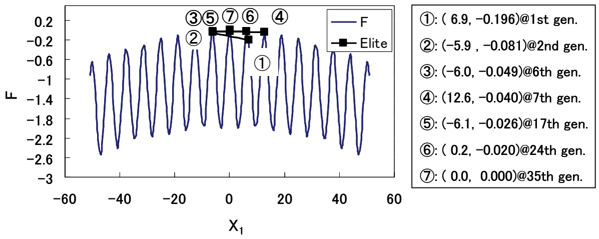


Fig. 11 Search process of elite individual to the function 2

3.2.2 Multiple dimensional DVs

Now, this experiment uses two and multiple dimensional DVs to the benchmark functions to assess the validity of search ability of APGA.

1) Two dimensional DVs: Tab. 4 and 5 list the results from eight benchmark functions. The solutions of all the benchmark functions reach their optima during every trial. Therefore, the success ratio of APGA becomes 100%. This can optimize the complex or multi-peak optimization problems. However, some trials are gotten an optimum point at hundreds place generation in the experiment of function 2, 6 and 7 as shown in Tab. 5. By means of this experiment, it could be confirmed that their calculation cost is necessary more than other functions. Because, their steepness levels are not an average.

2) Ten dimensional DVs: Next, Number of DVs is increased to fivefold its amount—ten dimensional DVs. Therefore, the difficulty of optimization is increased. As this experiment, function 0, 1, 2, 3 and 7—RA, RI, GR, AC and RO functions—are used. These have characteristic features—epistasis, multi-peak and steep (non average), and function 7 is introduced to this experiment because its calculation cost is biggest. GA parameter is changed only terminal generation from 1,000th to 5,000th generation. Fig. 16-20 show historical tendency of elite individual in optimization process for each function. Tab. 6 and Tab. 7 denote success ratio and found generation to optimal solution, respectively. APGA could search optimal solution at 100% of success ratio for the case of ten dimensions. In addition, as for function 2 and 7, the average of found generation is one digit larger than other functions.

3) Recombination of gene: The efficiency of recombination of gene is validated by using 1st trial of ten dimensional function 0. Fig. 21 shows its historical tendency of fitness. Its average fitness is moved closer to maximum fitness at the late generations as shown in Fig. 21(a). As shown in Fig. 21(b) which zooms in this range, a local convergence condition is kept from 94th until 102nd generations, and is gotten out by the strategy of recombination of gene at the 103rd generation. Moreover, maximum fitness is improved

by the strategy of two times which occurred at the 119th and 134th generations. As for this strategy, it is understood that it contributes to the process to search optimal solution. Therefore, it could be confirmed that this strategy is effective for APGA.

Tab. 4 Success ratios (2 dimensions)

Function	Success(%)
0	100
1	100
2	100
3	100
4	100
5	100
6	100
7	100

Tab. 5 Found generations of optimal solution (2 dimensions)

Function	Found generations		
	Trial 1	Trial 2	Trial 3
0	36	38	25
1	41	31	6
2	232	45	25
3	22	47	19
4	9	28	20
5	46	49	61
6	313	19	164
7	48	261	247

Function	Found generations		
	Trial 4	Trial 5	Average
0	27	31	31
1	48	96	44
2	30	67	80
3	37	18	29
4	29	13	20
5	44	64	53
6	38	243	155
7	40	191	157

Tab. 6 Success ratios (10 dimensions)

Function	Success(%)
0	100
1	100
2	100
3	100
7	100

Tab. 7 Found generations of optimal solution (10 dimensions)

Function	Found generations		
	Trial 1	Trial 2	Trial 3
0	138	147	169
1	540	486	575
2	1518	1143	1226
3	164	246	274
7	1128	1506	1445

Function	Found generations		
	Trial 4	Trial 5	Average
0	196	148	160
1	685	186	494
2	1326	1028	1248
3	127	230	208
7	1500	392	1194

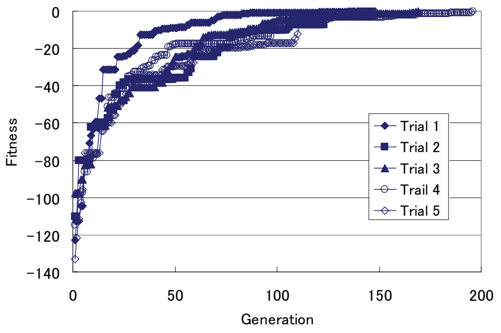


Fig. 16 Elite individual's fitness of Function 0 (10 dimensions)

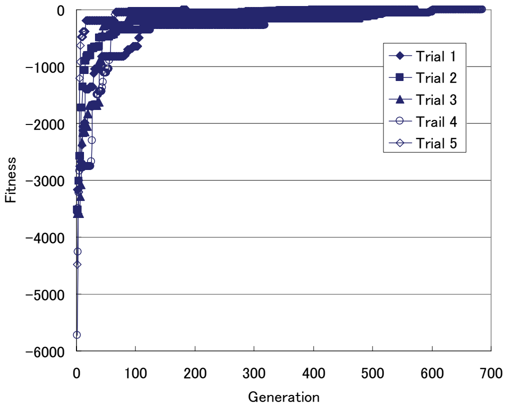


Fig. 17 Elite individual's fitness of Function 1 (10 dimensions)

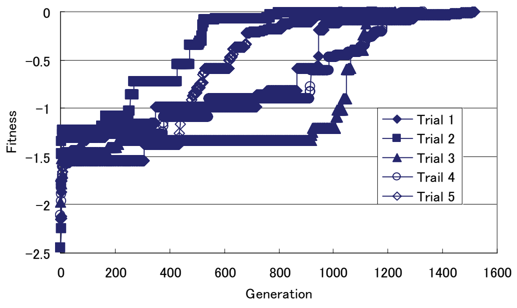


Fig. 18 Elite individual's fitness of Function 2 (10 dimensions)

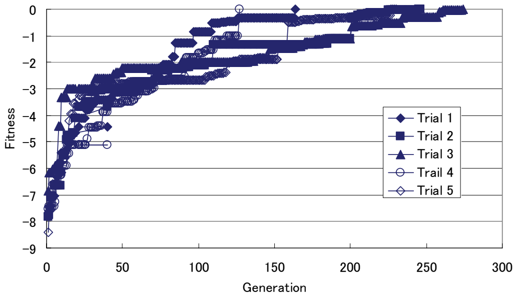


Fig. 19 Elite individual's fitness of Function 3 (10 dimensions)

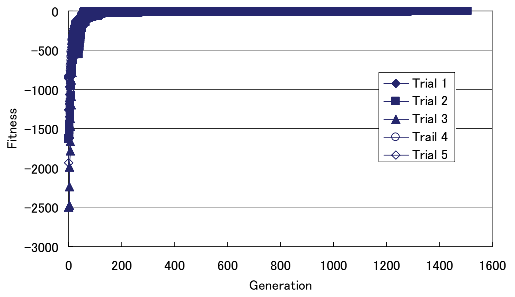
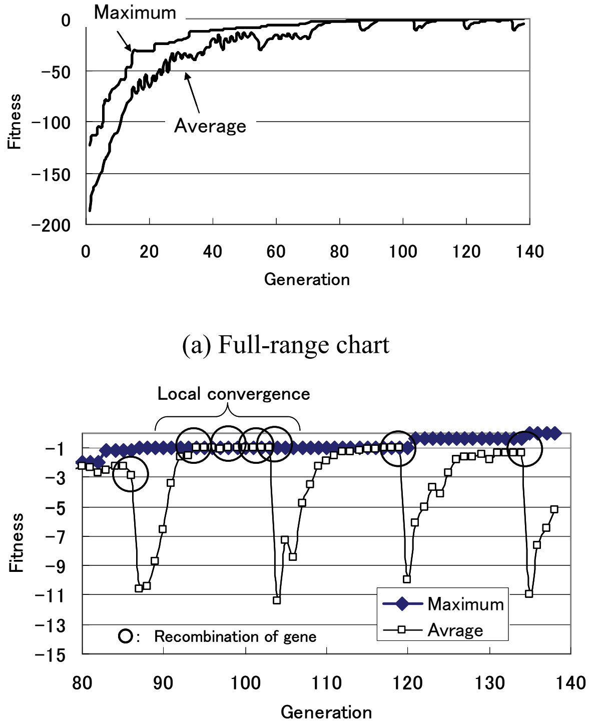


Fig. 20 Elite individual's fitness of Function 7 (10 dimensions)



(a) Full-range chart

(b) Close up

Fig. 21 Historical tendency of fitness for 1st trial of function 0

4) Control of neighbourhood range: The dimension of DVs is set at 20 dimensions to compare APGA with other methodologies such as GA and MA. The terminal generation of this experiment is 5,000th generation. Function 0-3 are used as for benchmark functions of this. Function 7 is not used because it is not a multi-peak problem, and its average of found generation is smaller than the value of Function 2.

This result is shown in Tab. 8. The optimal solutions could be gotten as for all trials of these functions except for function 2. As for function 2, APGA cannot achieve the optimal point, and its average fitness becomes -2.993. Because the optimal problem of function 2 exhibits a dependency on DVs and the steepness of this function is small, to solve its multiple dimensional optimization problem is difficulty.

Tab. 8 Found generations of optimal solution (20 dimensions)

Function	Found generations					
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
0	288	341	324	316	228	299
1	3475	4899	3213	1394	1692	2935
2	–	–	–	–	–	–
3	525	462	414	645	400	489

For overcoming this difficulty, the control of neighbourhood range is proposed. To apply this control is a common approach for SA. Therefore, this is introduced into AP, and can change the distance for search point by controlling neighbourhood range. This neighbourhood range is decided by neighbourhood ratio, and the formula of neighbourhood ratio is as follows:

$$AP(C_t, R_t) = -NR_t \cdot Scale \cdot SP \cdot sign(\nabla R_t), \quad (16)$$

$$NR_t = (t_{\max} - t)/(t_{\max} - 1), \quad (17)$$

where, t_{max} is terminal generation. Eq. (17) is a linear formula, and the relationship of the neighbourhood ratio and the generation is shown in Fig. 22.

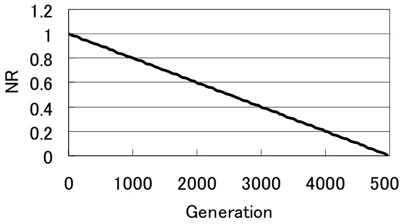


Fig. 22 Relationship of neighbourhood ratio and generation

The neighbourhood range of the early generations is adjusted to wide range by this control using this relationship, moreover is modified to small range at the late generations. As results, I believe that search process is varied according to this control schedule roughly or delicately toward the good search direction. To increase search distance of the early generations, scale factors are set as follows: scale factors of function 0, 1, 2 and 3 are 10.0, 100.0, 100.0 and 10.0, respectively.

This control method is introduced, and these optimizations are performed once again. As shown in

Tab. 9, from these results, the solutions of all the benchmark functions reach their optima during every trial, though calculation costs are doubled. Therefore, the success ratio of APGA becomes 100%. It could be confirmed that the control of neighbourhood range produces improvement to a convergence of optimal process.

Tab. 9 Found generations of optimal solution (20 dimensions)

Function	Found generations		
	Trial 1	Trial 2	Trial 3
0	657	747	658
1	3454	3877	2857
2	2274	1686	3053
3	2021	1323	1222

Function	Found generations		
	Trial 4	Trial 5	Average
0	568	523	631
1	4860	4115	3833
2	1589	1286	1978
3	715	1178	1292

3.2.3 Comparison for robustness

APGA is compared with other methodologies—GA in [3], Real-coded GA in [4], PSA/GAc in [15], MA in [17], Parallel SA with Selection (PSAS) in [14] and Hybrid meta-heuristics (HMH) in [19,20]. This comparison list is shown in Tab. 10. The calculation cost of APGA is the mean of 5 trials.

By means of this list, APGA could certainly achieve optimal solution with low calculation cost from results of comparison of GA and Real-coded GA. Next, APGA is compared with EAs—PSA/GAc, MA, PSAS and MHM. It is recognized that the convergence of the optimal solution could be improved more significantly in APGA than that in EAs for the same calculation cost.

Overall, APGA could be capable of attaining robustness, high quality, low calculation cost and efficient performance on many benchmark problems.

4 Conclusion

In this paper, the new EA—APGA—has been proposed to solve the multi-peak optimization problems with multiple dimensions. APGA is a stochastic global search heuristics in which EAs-based approaches are combined local search method. This uses AP to search local minima, and encodes CVs into genes to decide on a global behavior of AP. DVs are optimized by AP through this process. This newly proposed algorithm is applied to many benchmark functions to evaluate its performance, moreover is compared with GAs and EAs. As results, it has been confirmed that APGA dramatically improves calculation cost and convergence up to the optimal solution.

Finally, this study has the plan to apply APGA to optimization of product design to investigate this assistance capability.

Tab. 10 Comparison list

Function	Dimension		GA in Ref. [3]	Real-coded GA in Ref. [4]	APGA
0 (RA)	20	Success ratio	50-70%		100%
		calculation cost Num. of gen. Num. of func. call	-		299th gen. 14,950
0 (RA)	30	Success ratio		100%	100%
		calculation cost Num. of gen. Num. of func. call		- 4,010,000	486th gen. 24,290
2 (GR)	10	Success ratio	75-90%		100%
		calculation cost Num. of gen. Num. of func. call	-		1,248th gen. 62,400

Function	Dimension		PSA/GAc in Ref. [15]	MA in Ref. [17]	APGA
0 (RA)	10	Success ratio	100%		100%
		calculation cost Num. of gen. Num. of func. call	- 3,034,881		160th gen. 8,000
0 (RA)	20	Success ratio		0%	100%
		calculation cost Num. of gen. Num. of func. call		40,000th gen. 2,000,000	299th gen. 14,950
2 (GR)	10	Success ratio	100%	100%	100%
		calculation cost Num. of gen. Num. of func. call	- 3,008,201	21,250th gen. 1,062,500	1,248th gen. 62,400
2 (GR)	20	Success ratio			100%
		calculation cost Num. of gen. Num. of func. call			1,978rd gen. 98,900
7 (RO)	2	Success ratio		100%	100%
		calculation cost Num. of gen. Num. of func. call		4,177th gen. 208,850	157th gen. 7,850

Function	Dimension		PSAS in Ref. [14]	HMH in Ref. [19,20]	APGA
0 (RA)	20	Success ratio	100%	100%	100%
		calculation cost Num. of gen. Num. of func. call	1,500th gen. 150,000	3,630th gen. 145,184	299th gen. 14,950
1 (RI)	20	Success ratio	100%	100%	100%
		calculation cost Num. of gen. Num. of func. call	5,200th gen. 520,000	17,686th gen. 707,440	2,935th gen. 146,750
2 (GR)	20	Success ratio	100%	100%	100%
		calculation cost Num. of gen. Num. of func. call	4,100th gen. 410,000	9,003rd gen. 360,132	1,978rd gen. 98,900
3 (AC)	20	Success ratio	100%	100%	100%
		calculation cost Num. of gen. Num. of func. call	1,500th gen. 150,000	3,309th gen. 132,360	489th gen. 24,450

5 References

[1] J. Holland. Adaptation in Natural and Artificial Systems. *The University of Michigan* 1975, *MIT Press*, 1992.

[2] D. E. Goldberg. Genetic Algorithms in Search Optimization and Machine Learning. *Addison-Wesley*, 1989.

[3] J. D. Digalakis and K. G. Margaritis. An experimental 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)*, 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] H. Hasegawa and K. Kawamo. A Method for the Shape and Topology Optimization of Mechanical Structures by Using Genetic Algorithm (Layout Optimization Method by Adoption of Removal and Addition parameters of Elements as Chromosomes), *Transactions of the Japan Society of Mechanical Engineers, Series A*, 61(581):183-190, 1995 (in Japanese).

[7] Y. Tsuruta, H. Hasegawa and K. Kawamo. A Method for the Shape and Topology Optimization of Mechanical Structures Using Genetic Algorithm (2nd Report, On Convergence of Solutions of Our Method by Adoption of Removal and Addition parameters of Elements as Chromosomes), *Transactions of the Japan Society of Mechanical Engineers, Series A*, 63(605):170-177, 1997 (in Japanese).

[8] Y. Tsuruta, Y. Suzuki, H. Hasegawa and K. Kawamo. A Method for the Shape and Topology Optimization of Mechanical Structures Using Genetic Algorithm, *JSME Centenial Grand Congress by the JSME International Symposium on OPID97*, Paper #137, 1997.

[9] H. Nishi, T. Okamoto and K. Kawamo. Study on the Development of General-purpose Genetic Algorithm Engine. *Extend Abstracts of WCSMO-4*, 428-429, 2001.

[10]H. Kawada, H. Uehara and K. Kawamo. Development of General-purposed Optimization Engine and Test Run. *Journal of Japan Society for Simulation Technology*, 21(4):280-288, 2003 (in Japanese).

[11]B. E. Rosen and R. Nakano. Simulated annealing - basics and recent topics on simulated annealing -. *Journal of Japanese Society for Artificial Intelligence*, 9(3), 1994 (in Japanese).

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

[13]H. Uehara, H. Kawada. and K. Kawamo. Numerical Experiments on Optimal Points Searching Using Hybrid Method of Genetic Algorithm and Simulated Annealing, *52nd Japan National Congress for Theoretical and Applied Mechanics*, 117-118, 2003 (in Japanese).

[14]H. Uehara. Study on Development of General-purposed Optimization Engine and its Performance Evaluation - The proposal of Parallel Simulated Annealing with Selection -. *M of Eng thesis, Graduate School of Engineering*, Shibaura Institute of Technology, 2004 (in Japanese).

[15] T. Hiroyasu, M. Miki and M. Ogura. Parallel Simulated Annealing using Genetic Crossover. *Proceedings of the ISCA 13th International Conference on PDCS-2000*, 2000.

[16]M. Miki, T. Hiroyasu and T. Fushimi. Parallel Simulated Annealing with Adaptive Neighborhood Determined by GA. *IEEE International Conference on System, Man and Cybernetics*, 2003.

[17]Y. S. Ong and A. J. Keane. Meta-Lamarckian Learning in Memetic Algorithms. *IEEE transactions on evolutionary computation*, 8(2):99-110, 2004.

[18]W. E. Hart, N. Krasnogor and J. E.Smith. Recent Advances in Memetic Algorithms. *Springer*, 2005.

[19]H. Hasegawa, M. Yoshikawa, H. Uehara and K. Kawamo. The Hybrid Meta-heuristics by Reflecting Recognition of Dependence Relation among Design Variables for Integer Optimization of Multi-peak Problems. *Journal of Japan Society for Simulation Technology*, 25(2):144-155, 2006 (in Japanese).

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