

Distributed Probabilistic Model-Building Genetic Algorithm

Tomoyuki Hiroyasu¹, Mitsunori Miki¹, Masaki Sano¹, Hisashi Shimosaka¹,
Shigeyoshi Tsutsui², and Jack Dongarra³

¹ Doshisha University, Kyoto, Japan,
tomo@is.doshisha.ac.jp,

² Hannan University, Osaka, Japan,

³ University of Tennessee, TN, USA

Abstract. In this paper, a new model of Probabilistic Model-Building Genetic Algorithms (PMBGAs), Distributed PMBGA (DPMBGA), is proposed. In the DPMBGA, the correlation among the design variables is considered by Principal Component Analysis (PCA) when the offsprings are generated. The island model is also applied in the DPMBGA for maintaining the population diversity. Through the standard test functions, some models of DPMBGA are examined. The DPMBGA where PCA is executed in the half of the islands can find the good solutions in the problems whether or not the problems have the correlation among the design variables. At the same time, the search capability and some characteristics of the DPMBGA are also discussed.

1 Introduction

Genetic Algorithms (GAs) are stochastic search algorithms based on the mechanics of natural selection and natural genetics[1]. The GAs can be applied to several types of optimization problems by encoding design variables to individuals. Recently, a new type of GA called the Probabilistic Model-Building GA (PMBGA)[2] or Estimation of Distribution Algorithm (EDA)[3] have been the focus. In the canonical GA, children are generated from parents that are selected randomly. However, in the PMBGA and EDA, the good characteristics of parents are inherited by children using the statistical information. Since children must have the characteristics of parents, effective searching is expected. It is reported that the PMBGA and EDA have a better search ability than that of the canonical GA.

To make an effective search in continuous problems, the correlation among the design variables should be handled. Therefore, new searching points should be generated so that the correlation exists in the new points. Many real coded GAs where the real vectors are used as a genotype treat this correlation problem. One of the typical real coded GAs is Unimodal Normal Distribution Crossover (UNDX)[4]. The UNDX is good at finding the optimum in the functions where there is strong correlation between the design variables. Takahashi et al. introduced a new method[5]; they used the Principle Component Analysis (PCA) and

the Individual Component Analysis (ICA). In the Takahashi et al. algorithm, by using the PCA and the ICA, the individuals are transferred into the space. Then, Blend Crossover (BLX- α) [6] is performed to the transferred individuals. The correlation between the design variables is considered by using both ICA and PCA. At the same time, the diversity of the solutions is maintained by using BLX- α . Besides GAs, some of the evolutionary strategies[7] have the operation for the correlation relationship among the design variables. One of them is the correlated mutation method that is proposed by Schwefel[8]. In this method, there is a parameter that indicates the direction of the distribution of the individuals. By using this parameter, the mutation operation is performed by considering the correlation relationship among the design variables.

In this paper, a new PMBGA for continuous problems is proposed, which is called Distributed PMBGA (DPMBGA). This is one of the real coded GAs and the real vectors are treated as a genotype. In this algorithm, PCA is used for transforming the set of the solutions. This operation handles the correlation relationship among the design variables. The PMBGA sometimes lacks the diversity of the solutions during the search. To overcome this problem, a model of the distributed GA is performed.

In this paper, the basic algorithm of the DPMBGA is explained. In this paper, the DPMBGA is applied to solve test functions. Through these experiments, the following four topics of the DPMBGA are discussed. Firstly, the DPMBGA that is based on the distributed scheme is examined. Secondly, the search capability of the DPMBGA is compared with UNDX and Minimum Generation Gap. From the results, it is found that the PCA prevents the effective search in some functions. The discussion of the results is the third topic. Finally, the search capability of the DPMBGA for functions whose optimum is located near the boundary is discussed.

2 Distributed Probabilistic Model-Building Genetic Algorithm

2.1 Flow of DPMBGA

In this paper, a new PMBGA is proposed: that is Distributed Probabilistic Model-Building Genetic Algorithm (DPMBGA). The DPMBGA uses the distributed GA scheme[9,10]. Therefore, there are several subpopulations and the migration operation is performed. In the DPMBGA, the following procedure is performed for the migration operation. The topology of the migration is a ring. This ring topology is formed randomly when the migration is performed. In the ring, migrated individuals are moved from one subpopulation to the other in one direction. The migrated individuals are chosen randomly and they are substituted for the individuals whose fitness values are the worst in the subpopulation.

In the DPMBGA, the following procedures are performed at the generation t .(Fig. 1).

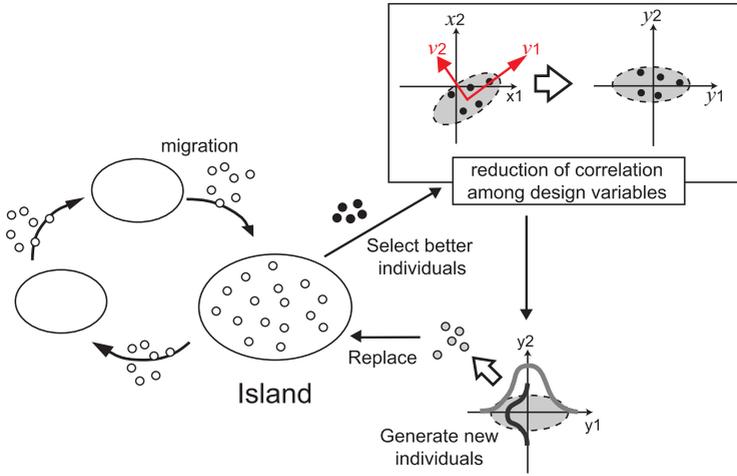


Fig. 1. DPMBGA

1. The elite individual is reserved.
2. The individuals who have the good values of the fitness are sampled.
3. The above individuals are transferred by the PCA into the new space.
4. The new individuals are generated in the new space.
5. The new individuals are transferred into the original space.
6. The new individuals are substituted for the old individuals.
7. The mutation is operated.
8. When the reserved elite individuals are eliminated, they are recovered.
9. The new individuals are evaluated.

In the following sections, each operation is explained precisely.

2.2 Sampling Individuals for Probabilistic Model

The following operation is performed in each island. The individuals who have the good evaluation values are chosen from each island $P_{sub}(t)$. The number of these individuals is determined with the sampling rate R_s . These individuals become sample individuals $S(t)$. The new individuals are generated from the information of these sampling individuals. These sampled individuals are chosen according to the higher fitness values. However, the same individual is not chosen repeatedly, hence the total number of $S(t)$ is fixed. When the number of individuals is low, the individuals are generated randomly and added to $S(t)$ and $S(t)$ exists in each subpopulation.

2.3 Sampling Individuals for PCA

This operation is also performed in each island. $S(t)$ is transferred by the PCA operation. The PCA is determined using the information of individual set $T(t)$.

$T(t)$ is different from $S(t)$ and $T(t)$ is formed in the following way. $T(t)$ consists of the individuals who are the best in each generation. Even when the number of $T(t)$ is less than the particular size, the new individuals are not added. When the size of $T(t)$ is exceeded, the worst individual is eliminated one by one. By this operation, the arbitrary number of the individuals can be used for the information of PCA. This is independent from the number of the subpopulation. $T(t)$ also exists in each island.

2.4 PCA Transformation

The average of $T(t)$ is subtracted from $T(t)$ and $T(t)$ becomes matrix \mathbf{T} ($nT(t)column \times Dline$). The average of $T(t)$ is also subtracted from $S(t)$ and $S(t)$ becomes \mathbf{X} ($nS(t)column \times Dline$). Then, the covariance matrix \mathbf{S} of \mathbf{T} is derived and the eigen values and vectors are obtained. \mathbf{S} is a real symmetric matrix and derived as follows,

$$\mathbf{S} = \frac{1}{nS(t) - 1} \mathbf{T}^T \mathbf{T}. \quad (1)$$

The eigen vector indicates the axis of the new space.

Using the derived eigen vectors, the design variables \mathbf{X} of the solution set $S(t)$ are transferred. After the transfer into the new space, there is no correlation among the design variables. The coordinate transfer matrix consists of the vectors $\mathbf{V} = [v_1, v_2, \dots, v_D]$. After multiplying \mathbf{V} , the vector \mathbf{X} becomes \mathbf{Y} . The coordinate of \mathbf{Y} corresponds to the eigen vectors.

2.5 Generation of New Individuals

The new individuals are generated using the normal distribution of the information of \mathbf{Y} . Each value of the design variable in a new individual is also determined one by one independently. Therefore, when there are n design variables in an individual, there should be n different normal distributions. The normal distribution is formed as follows; the average is the same as the average value of the target design value of \mathbf{Y} . The distribution is derived by multiplying the distribution of \mathbf{Y} by the parameter *Amp*. The values of design variables are determined randomly, but the total distribution of the new individuals should be the same as the formed normal distribution. The number of the created new individuals is the same as the number of individuals in an island ($nP(t)$) and the generated individuals are stored in \mathbf{Y}_{offs} .

2.6 Restoring Correlation and Substitution of Old Individuals with New Individuals

\mathbf{Y} is the transferred set of \mathbf{X} into the new space. In this step, the derived \mathbf{Y}_{offs} is then substituted into the original space. \mathbf{Y}_{offs} is multiplied by the inverse of \mathbf{V} . After this operation, the set of \mathbf{Y}_{offs} is in the original space.

$$\mathbf{X}_{offs} = \mathbf{Y}_{offs} \cdot \mathbf{V}^{-1} \quad (2)$$

The average of \mathbf{X}_{offs} is added to the new individuals. These new individuals are substitute for the old ones $P(t)$ and those become $P(t + 1)$.

2.7 Mutation

The values of the design variables are changed randomly within the constraints using the mutation ratio R_{mu} .

2.8 Preservation and Recovering of Elite Individuals

The elite individuals are preserved as $E(t)$. The number of the preserved individuals is $nE(t)$. After the substitution of the new individuals generated from the probabilistic model, the elite individuals are recovered in the total population. In this case, the elite individuals $E(t)$ are substituted with the individuals $P(t + 1)$ whose evaluation values are not good.

3 Test Functions and Used Parameters for Numerical Experiments

In the following sections, the search capability and the characteristics of the DPMBGA are discussed. These discussions are illustrated through numerical experiments. In this section, the test functions and parameters for these experiments are explained.

The DPMBGA is used to find optimum solutions of the following five test functions: the Rastrigin function, Schwefel function, Rosenbrock function, Ridge function, and Griewank function. All of test functions are minimization problems. The global optimums are located at 0. There are 10 dimensions of the design variables in the Schwefel function and 20 dimensions of the design variables in the rest of the functions.

There is no correlation between the design variables in the Rastrigin function and the Schwefel function. There are many sub-peaks in the landscape of these functions. On the other hand, there is a correlation between the design variables in the Rosenbrock function and the Ridge function. In these test functions, there is only a peak in a landscape. In the Griewank function, there is a correlation between the design variables and many peaks in the landscape.

$$F_{Rastrigin} = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (3)$$

$$(-5.12 \leq x_i < 5.12)$$

$$F_{Schwefel} = \sum_{i=1}^n -x_i \sin\left(\sqrt{|x_i|}\right) - C \quad (4)$$

(C : optimum.)
 $(-512 \leq x_i < 512)$

$$F_{Rosenbrock} = \sum_{i=2}^n (100(x_1 - x_i^2)^2 + (1 - x_i)^2) \quad (5)$$

$(-2.048 \leq x_i < 2.048)$

$$F_{Ridge} = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \quad (6)$$

$(-64 \leq x_i < 64)$

$$F_{Griewank} = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left(\cos\left(\frac{x_i}{\sqrt{i}}\right) \right) \quad (7)$$

$(-512 \leq x_i < 512)$

The parameters used in these experiments are summarized in Table 1.

Table 1. Parameters

Population size	512
Number of elites	1
Number of islands	32
Migration rate	0.0625
Migration interval	5
Archive size for PCA	100
Sampling rate	0.25
Amp. of Variance	2
Mutation rate	0.1/ (Dim. of function)

4 Discussion on Effectiveness of PCA and Distributed Environment Scheme

In the DPMBGA, the new individuals are generated using the PCA. By this operation, the information from the correlation among the design variables that is found during the search is reflected to the new individuals. In the problems where there is a correlation among the design variables, the distribution of the value of each design variable is not independent from each other. Therefore,

the distribution of each design variable is affected by the distribution of the other design variables. In the new model, at first, the set of the individuals is transferred into the space where there is no correlation among the design variables. After this transformation, it is easy to generate new individuals using their information of the distribution. Therefore, it is expected that the DPMBGA can perform an effective search in the problems where there is a strong correlation among the design variables. In the numerical experiments, the following three models are discussed.

Model 1 : In every island, the PCA is performed.

Model 2 : In every island, the PCA is not performed.

Model 3 : In half of the islands, the PCA is performed.

In these models, the number of the islands where the PCA is performed is different. The model 1 is the same model explained in section 2. In model 2, the PCA is not performed at all. In model 3, the distributed environment scheme is applied for using the PCA.

The Distributed Environment GA (DEGA) is one of the distributed GA schemes that Miki et al. proposed[11]. In the DEGA, the different parameters or the different operations are performed in each island. It is well known that the search capability of GA depends on the value of the parameters. The optimum values of these parameters also depend on the targeted problems. Therefore, preliminary experiments are necessary in order to derive the optimum values of the parameters. In the DEGA, the values of the parameters and the operations are different in each island. These parameter values are not the best but can derive adequate solutions. In this paper, the DEGA scheme is applied; the CPA is performed in some islands and not in other islands.

In Table 2, the number of the trials where the optimum value is derived is summarized in using 20 trials. The higher number means that the robust search has been performed. In Figure 2, the average number of the evaluations is illustrated when the simulation derives the optimum values. The model that has the smaller number of this value may be the better model.

Table 2. Number of times that the threshold is reached

	model 1	model 2	model 3
Rastrigin	0	20	20
Schwefel	20	20	20
Rosenbrock	20	0	20
Ridge	20	20	20
Griewank	19	17	20

In the Schwefel function, all the models derive the optimum solutions in every trial with the small number of the evaluations. In the Rastrigin function, the search ability of model 1 is worse than the other models. That means the

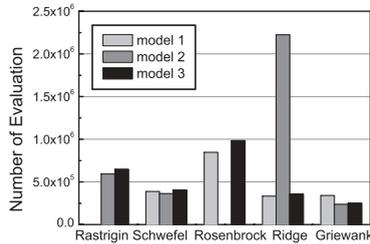


Fig. 2. Comparison of search capability between models

PCA does not help for achieving a good solution. In the Rosenbrock function that has the correlation relationship among the design variables, model 2 (where the PCA is not performed) cannot derive the optimum solutions. In the Ridge function that also has the correlation, model 2 can derive the optimum solutions but needs many evaluations compared to the other models. These results suggest that the PCA should have a positive effect on the search in these problems.

From these results, the following point is made clearer; the PCA is useful for the problems that have the correlation among the design variables but not useful for the problems that do not have the correlation. Therefore, the effect of the PCA depends on the type of the problems. On the other hand, model 3 where the PCA operations are performed in some islands but not in others is good at finding optimum solutions in every function. The Griewank function has the correlation among the design variables and has also many peaks in the landscape. Therefore, it is a difficult problem to find the optimum. Model 1 and 2 did not find the optimum solutions in some trials. On the other hand, model 3 derived the optimum in all the trials. From these results, there is a possibility that model 3 can find the optimum solutions not only in the problems whether the correlation of the design variables exists or not but also in the problems where there are many peaks in the landscape. Because of the result in this section, in the following discussions, model 3 is used.

5 Comparison of DPMBGA with UNDX + MGG

In this section, the search capability of the DPMBGA is compared with the conventional real-coded GA. The comparison real-coded GA is UNDX[4] with Minimal Generation Gap (MGG)[12,13]. In the UNDX, two new individuals are generated from three individuals. Two of the parent individuals form the main axis with the normal distribution generated on this main axis. The third parent determines the variance of the normal distribution. The child individuals are generated in accordance with this normal distribution. Using the UNDX, an effective search can be performed with the consideration of the correlation between the design variables.

MGG is one of generation alternation models. When the generation alternation occurs in the MGG, the following procedure is performed. Two of the parent individuals are chosen from the population randomly from which child individuals are generated by the crossover with the parent one time or many times. From the set of the child and parent individuals, the individuals that remain as the next generation are selected. These individuals are then substituted for the parent individuals and are backed to the total population. The MGG has the characteristics to maintain the diversity of the solutions during the search since the selection is limited to the small number of solutions.

In Figure 3, the transitions of the DPMBGA search and the UNDX+MGG model are shown. The horizontal axis shows the number of evaluations and the vertical axis shows the average fitness values in 20 trials. In the UNDX+MGG model, some of 20 trials did not get the optimum solution. The average values of the trials where the optimum solutions are derived are illustrated. These are minimization problems with smaller fitness values indicating the better solutions.

For the parameters of UNDX+MGG, there are 300 individuals for the functions with many peaks and there are 50 individuals for the functions with only one peak. The number of crossover is 100 and $\alpha = 0.5$. $\beta = 0.35$ is used.

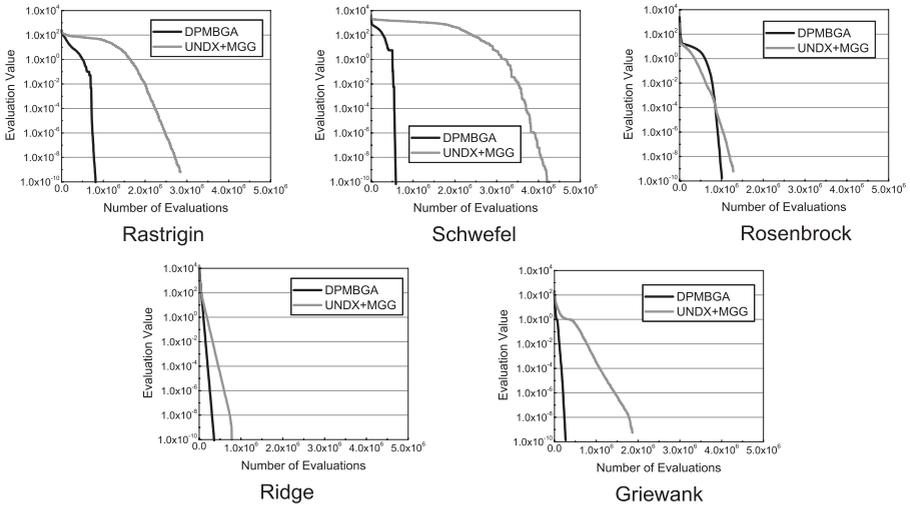


Fig. 3. History of average of evaluation values

Figure 3 indicates that the results of the DPMBGA are better than the other models in the problems whether a correlation relationship between the design variables exists or not. Therefore, it can be concluded that the DPMBGA is a useful GA for these continuous functions.

6 Discussion on Case Where PCA Does Not Work Effectively

In the former section, it is found that the PCA operation sometimes prevents the effective finding of the optimum. In the Rastrigin function, the model without the PCA operation derived better solutions than the model with the PCA. In this section, the reasoning for why the PCA does not work effectively is discussed.

One reason may be the early convergence of the solutions in the archive. The PCA uses the information from the solutions in the archive and it is important for the effective search that this archived information should reflect the information of the real landscape of the problem. If these solutions in the archive are not renewed, the proper transfer by the PCA cannot be expected to occur. Since the Rastrigin function has many sub-peaks on the landscape, it may be that many solutions are stuck in the sub-peaks. Then, the solutions of the archive are not renewed possibly preventing an effective search.

This assumption is illustrated with numerical experiments. In these experiments, the model where the PCA operation is performed in all the sub-populations is used. The target test functions are the Rastrigin function and the Rosenbrock function.

In Figure 4, the history for the renewal of the archive is shown. The horizontal axis shows the number of the evaluations and the vertical axis shows the number of the renewed individuals. From this figure, it is obvious that the number of the renewed individuals becomes small, especially in the latter part of the search in the Rastrigin function.

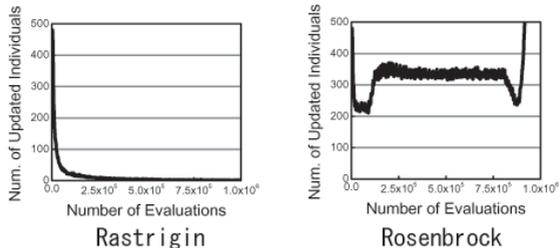


Fig. 4. History of number of updated individuals in archive of the best individuals

Conversely, in the Rosenbrock function, the number of the renewed individuals does not decrease and most of the individuals of the archive are always renewed.

In Figure 5, the history of the search is illustrated. The horizontal axis shows the number of evaluations and the vertical axis shows the average value of the evaluation in 20 trials. This is the minimizing problem; the smaller evaluation value indicates the better solution. The term "normal" in this figure indicates

the result of the normal model. "erase/10" indicates the result of the model where the archive of the individuals is eliminated every 10 generations.

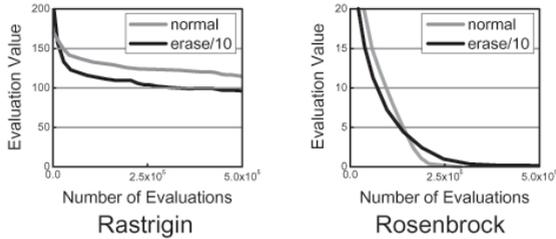


Fig. 5. History of average of evaluation values in the model in which archive is erased each 10 generation

From this figure, the result of the "erase/10" model is better than the normal model in the Rastrigin function. On the other hand, the result of the "normal" model is better than the "erase/10" in the Rosenbrock function. These results point out that the archive effect is worse in the Rastrigin function.

In conclusion, one of the reasons that the PCA operation prevents an effective search in the Rastrigin function is stagnation of the renewal of the individuals in the archive.

7 Discussion on Search Capability of DPMBGA for Functions Whose Optimum Is Located Near the Boundary

When a normal distribution is used in the crossover operation, it is often said that the real-coded GA is good at finding the solution in the problem where the optimum is located at the center of the search area but is not good at finding the solution in the problem where the optimum is located at the boundaries [14].

One solution to this problem is Boundary Extension by Mirroring (BEM) [15]. In the BEM, the solutions that violate the constraints can exist when these solutions are within certain distance. The distance is determined by the extension rate r_e ($0.0 < r_e < 1.0$).

The DPMBGA is one of the real-coded GAs that may be weak at finding the solutions in the problems where the optimum is located in the boundary. At the same time, when the optimum is located on the boundary, the probabilistic model may be different from the real distribution of the individuals. This situation may prevent an effective search. The search capability of the DPMBGA for problems where the optimum solutions is located near the boundary is discussed in this

section. The search capability of the DPMBGA is compared with the model using the BEM.

The test functions are modified to have their optimum solutions near their boundaries. The ranges of the functions are summarized in Table 3.

Table 3. Domain of objective functions

Function	Optimal solution	Domain
Rastrigin	0.0	[0 , 5.12]
Schwefel	420.968746	[-512, 421]
Rosenbrock	1.0	[-2.048, 1]
Ridge	0.0	[0, 64]

In Figure 6, the transition of the search is expressed. The horizontal axis shows the number evaluations and the vertical axis shows the average of the fitness values for 20 trials.

These figures illustrate that the proposed model where BEM is not used derives better solutions. In the proposed model, the individuals who are out of the feasible region are pulled back on the closest boundary of the feasible region. The search concentrates on the individual with a good evaluation value by the DPMBGA. When the optimum solution is on or near the boundary, the search is concentrated near the boundary. This may be a reason why the proposed model is better than the model using the BEM. Thus, the DPMBGA derives the good solution for the type of problems with constraints that are the upper and lower boundaries of the design variables.

8 Conclusions

In the DPMBGA, the correlation among the design variables is analyzed by PCA. The new individuals are generated from the probabilistic model that is formed with the individuals that have good fitness values. However, before generation, the individuals that are used for forming the probabilistic model are transferred by PCA into the space where there is no correlation among the design variables. Then the new individuals are generated and these are placed into the original space. From this operation, the generated new individuals may have the correlation among the design variables. Therefore, an effective search may be expected. At the same time, the island model is utilized for maintaining the diversity of the solutions during the search.

The DPMBGA is applied to find the optimum solutions of the test functions. Through these numerical experiments, the following four topics are made clarified.

Firstly, the DPMBGA with PCA operations is useful for finding the solutions in the test functions where there is a correlation relationship among the design

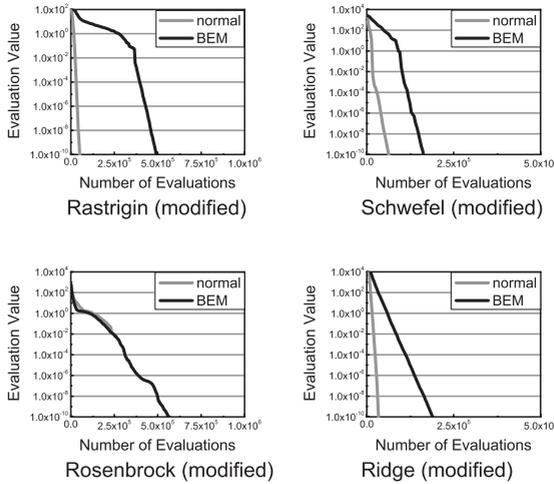


Fig. 6. History of the average of evaluation values on functions with an optimum at the edge of search space

variables. On the other hand, the PMBGA without PCA is good at finding the optimum in the functions where there is no correlation relationship among the design variables. From these results, the new model of PMBGA that is based on the distributed environment scheme where PCA is performed in only half of the subpopulations is proposed. The DPMBGA is very useful for finding the optimum in the functions whether or not there is a correlation relationship among the design variables or not.

Secondly, the results of the DPMBGA are compared to those of UNDX with MGG. This comparison shows that the DPMBGA has higher search capability.

In the DPMBGA, the Principle Component Analysis is used to analyze the correlation between the design variables. However, the PCA does not work effectively for finding the optimum solutions in some test functions. The reason for this problem is the third discussion. Numerical experiments conclude that one of the reasons the PCA operation prevents the effective search in the function is stagnation of the renewal of the individuals.

Finally, the DPMBGA is used to find the solutions in the functions where the optimums are located at the edge of the feasible region. In the DPMBGA, when the new individuals violate the constraints, the solutions are pulled back on the boundary. This operation is compared with the BEM. Numerical experiments illustrate that the operation in the proposed method is better than BEM in these test functions.

References

1. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley (1989)
2. Pelikan, M., Goldberg, D.E., Lobo, F.: A Survey of Optimization by Building and Using Probabilistic Models. Technical Report 99018, IlliGAL (1999)
3. Larranaga, P., Lozano, J.A.: Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Academic Publishers (2001)
4. Ono, I., Satoh, H., Kobayashi, S.: A Real-Coded Genetic Algorithm for Function Optimization Using the Unimodal Normal Distribution Crossover. Transactions of the Japanese Society for Artificial Intelligence **14** (1999) 1146–1155
5. Takahashi, M., Kita, H.: A Crossover Operator Using Independent Component Analysis for Real-Coded Genetic Algorithms. Proc. 13th SICE Symposium on Decentralized Autonomous Systems (2001) 245–250
6. Eshelman, L. J. and Shaffer, J. D.: Real-coded Genetic Algorithms and Interval Schemata. Foundation of Genetic Algorithms Morgan Kaufmann Publishers (1993) 182–202
7. Bäck, T., Hoffmeister, F., Schwefel, H.P.: A Survey of Evolution Strategies. Proc. 4th International Conference on Genetic Algorithms (1991) 2–9
8. Hansen, N.: Invariance, Self-Adaptation and Correlated Mutations and Evolution Strategies. Parallel Problem Solving from Nature – PPSN VI, 6th International Conference, 2000, Proceedings (2000) 355–364
9. Tanese, R.: Distributed Genetic Algorithms. Proc. 3rd International Conference on Genetic Algorithms (1989) 434–439
10. Cantú-Paz, E.: Efficient and accurate parallel Genetic Algorithms. Kluwer Academic publishers (2000)
11. Miki, M., Hiroyasu, T., Kaneko, M., Hatanaka, K.: A Parallel Genetic Algorithm with Distributed Environment Scheme. IEEE Proceedings of Systems, Man and Cybernetics Conference SMC'99 (1999)
12. Sato, H., Ono, I., Kobayashi, S.: A New Generation Alternation Model of Genetic Algorithms and its Assessment. Transactions of the Japanese Society for Artificial Intelligence **12** (1997) 734–744
13. Yamamura, M., Satoh, H., Kobayashi, S.: An Analysis on Generation Alternation Models by using the Minimal Deceptive Problems. Transactions of the Japanese Society for Artificial Intelligence **13** (1998) 746–756
14. Someya, H., Yamamura, M.: Toroidal Search Space Conversion for Robust Real-Coded GA. Proc. 13th SICE Symposium on Decentralized Autonomous Systems (2001) 239–244
15. Tsutsui, S.: Multi-parent Recombination in Genetic Algorithms with Search Space Boundary Extension by Mirroring. Proc. the 5th International Conference on Parallel Problem Solving from Nature (PPSN V) (1998) 428–437