

# Real-coded Estimation of Distribution Algorithm by Using Probabilistic Models with Multiple Learning Rates

Masahiro Nakao<sup>a</sup>, Tomoyuki Hiroyasu<sup>b</sup>, Mitsunori Miki<sup>c</sup>, Hisatake Yokouchi<sup>b</sup>, Masato Yoshimi<sup>c</sup>

<sup>a</sup>*Center for Computational Sciences, University of Tsukuba, Ibaraki, Japan*

<sup>b</sup>*Department of Life and Medical Sciences, Doshisha University, Kyoto, Japan*

<sup>c</sup>*Faculty of Science and Engineering, Doshisha University, Kyoto, Japan*

---

## Abstract

Here, a new Real-coded Estimation of Distribution Algorithm (EDA) is proposed. The proposed EDA is called Real-coded EDA using Multiple Probabilistic Models (RMM). RMM includes multiple types of probabilistic models with different learning rates and diversities. The search capability of RMM was examined through several types of continuous test function. The results indicated that the search capability of RMM is better than or equivalent to that of existing Real-coded EDAs. Since better searching points are distributed for other probabilistic models positively, RMM can discover the global optimum in the early stages of the search.

**Keywords:** estimation of distribution algorithm, optimization, continuous function

---

## 1. Introduction

Estimation of Distribution Algorithms (EDAs)[1, 2, 3] has been expected to solve optimization problems efficiently. The optimization problem is to find design variables that give maximum or minimum value to fitness function under constraint condition. The optimization problems are generally described as follows.

$$\text{minimize(maximize)} f(x), \quad \text{subject to } x \in F \quad (1)$$

Several types of EDAs have been proposed as new evolutionary algorithms based on probabilistic models with substantial mathematical background. Bit-string type EDAs were developed in early EDA studies[4, 5, 6]. Recently, Real-coded EDAs the search points of which are expressed with real value vectors have been introduced[7, 8].

Continuous Population-Based Incremental Learning (PBILc)[9] is a simple Real-coded EDA. It was reported that PBILc can search effectively for finding good solutions to continuous test functions with high dimensions. PBILc builds probabilistic models with estimation of the distribution that utilizes the current promising population, past population, and learning rate.

To obtain better solutions, it is important for EDA to control diversity of the population similar to GA[10]. However, for EDA to keep the diversity of the population, it requires a great deal of time to obtain a good solution. This is because the population converges early. On the other hand, when EDA loses the diversity of the population, the population converges early into local optima. Thus, the diversity of the population is very important[11].

---

*Email address:* [mnakao@ccs.tsukuba.ac.jp](mailto:mnakao@ccs.tsukuba.ac.jp) (Masahiro Nakao)

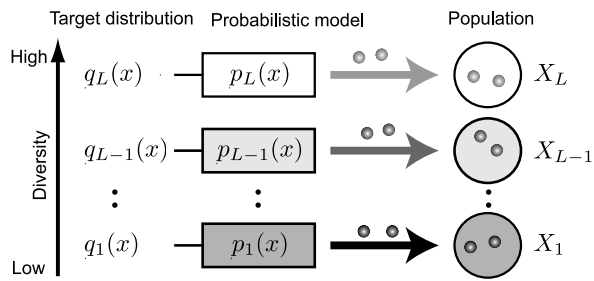


Figure 1: Sampling

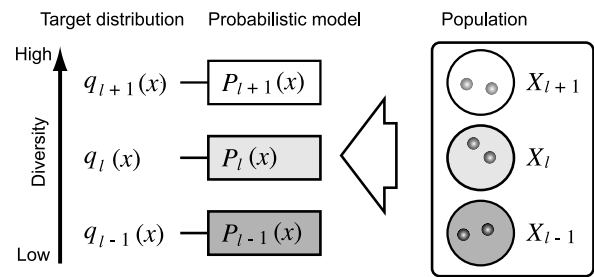


Figure 2: Estimation

Hierarchical Importance Sampling (HIS)[12] is an EDA that pays attention to diversity control of the population. The main mechanism of HIS is to search using multiple populations with different diversities. Although HIS has been confirmed to be effective for distributed test functions, it is difficult to obtain a good solution for continuous test functions with some probabilistic models[13].

This paper proposes a novel method, Real-coded EDA using Multiple Probabilistic Models (RMM), which utilizes probabilistic models with multiple learning rates. RMM refers to PBILc to build a probabilistic model and HIS to exchange information in populations with different diversities. After describing the algorithm of RMM, the performances of RMM are compared with those of existing EDAs, and we investigate the utility of probabilistic models with multiple learning rates.

## 2. Hierarchical Importance Sampling

In general EDA, only one population and one probabilistic model are used. Therefore, if the population has been converged and the global optimum is outside the population range, it is difficult to find the global optimum. HIS[12] is another type of EDA that uses multiple populations ( $X_1, \dots, X_L$ ) and probabilistic models ( $p_1(x), \dots, p_L(x)$ ) with different diversities. HIS searches using all populations and probabilistic models, and is designed that some populations are converged and others are not. Thus, HIS performs not only to find good solutions quickly but also to avoid converging into the local optimum.

HIS has two main operators, which are performed repeatedly sampling and estimation. In sampling, all probabilistic models create search points independently (Fig. 1). In estimation, all the populations are merged into one pool from  $X_l, X_{l+1}$  and  $X_{l-1}$ . Then, new probabilistic models are built from the good search points, which are selected from the merged pool (Fig. 2).

In HIS, the diversity of the new created population has to be determined and the target distribution ( $q_1(x), \dots, q_L(x)$ ) of the probabilistic model should be settled for each model. The target distribution is designed as a model the diversity of which is changed sequentially. In HIS, when a new probabilistic model is created, the number of selected search points is not determined but the target distribution is fixed. Therefore, search points should be selected to build a new probabilistic model that has the target distribution. Thus, the number of selected points is small when the diversity of the target model is small, and conversely the number of selected points is large when the diversity of the target model is large.

This section described several mechanisms of HIS. These mechanisms help HIS to control the diversity of populations. Therefore, HIS can search effectively.

## 3. Proposal of Real-coded EDA by Using Probabilistic Models with Multiple Learning Rates

### 3.1. Design Policy

In this section, RMM that uses multiple populations and probabilistic models similar to HIS is proposed. RMM refers to a probabilistic model and method for generating search points of PBILc[9]. PBILc uses a simple rule for updating the probabilistic model based on the normal distribution ( $N(\mu_i, \sigma_i^2)$ ,  $i$  is a dimension number) in real-coded EDAs, and we feel that facilitates to investigate a effectiveness of using multiple probabilistic models.

- 
1. Generate search points using uniform distribution in each population ( $X_1, \dots, X_L$ ).
  2. Calculate the function values of all search points in populations.
  3. Initialize average ( $\mu_i$ ) and variance ( $\sigma_i^2$ ) in each probabilistic model. The  $\mu_i$  is equal to the design variable of the best search point in the population. The  $\sigma_i^2$  is set as a squared size of 1/4 of domain.
  4. do{
    - For  $l = 1$  to  $L$ 
      - (1) Select good search points from  $X_{l-1}, X_l, X_{l+1}$ .
      - (2) Update probabilistic model  $p_l$  according to eq. 3, 4 and learning rate  $\beta_l$ .
      - (3) Generate next population from probabilistic model  $p_l$ .
      - (4) Replace population  $X_l$  with the generated population.
 }until(stopping criterion reached)
- 

Figure 3: Pseudo Code

The proposed algorithm has a mechanism to alter the learning rate used for updating each probabilistic model to generate various populations. PBILc uses variance ( $\sigma_i^2$ ) in a probabilistic model, and the value of variance can be controlled by the learning rate.

### 3.2. Algorithm

There are five parameters in RMM: number of probabilistic models and population couples ( $L$ ), number of search points generated by one probabilistic model ( $C$ ), cutting rate ( $K$ ), maximum learning rate ( $\beta_1$ ), and minimum learning rate ( $\beta_L$ ). The pseudo-code of RMM is shown in Fig. 3.

First, RMM initializes all populations ( $X_1, \dots, X_L$ ) and probabilistic models ( $p_1, \dots, p_L$ ). In the initialization of population, it generates search points by uniform distribution. For the initial probabilistic model, the average ( $\mu_i$ ) of normal distribution's parameter the model has that is determined as a design variable of best search point in the initialized population. The variance ( $\sigma_i^2$ ) should be a large value to search widely in the first stage. Thus, the variance is set a squared size of 1/4 of domain.

To change the diversity of each population, the maximum learning rate ( $\beta_1$ ) is set to the probabilistic model with the lowest diversity, and the minimum learning rate ( $\beta_L$ ) is set to the probabilistic model with the highest diversity. The other learning rate ( $\beta_l, (l = 2 \dots L - 1)$ ) is determined for the value to become small sequentially. In this paper, the learning rate is determined using the following equation:

$$\beta_l = \beta_1 \cdot \left( \frac{\beta_L}{\beta_1} \right)^{\frac{l-1}{L-1}} \quad (2)$$

To update the probabilistic model, it selects  $3 \times C \times K$  good search points from three populations ( $X_{l-1}, X_l, X_{l+1}$ ). If the case of  $l = 1$  or  $l = L$ , it selects  $2 \times C \times K$  good search points from two populations ( $X_1 \cdot X_2$  or  $X_{L-1} \cdot X_L$ ).

RMM builds each probabilistic model by using selected search points. The average ( $\mu_i$ ) and variance ( $\sigma_i^2$ ) of normal distribution's parameter is calculated by eq. 2, 3 and 4 similar to PBILc.

$$\mu_i^{t+1} = (1 - \beta)\mu_i^t + \beta(X^{best1} + X^{best2} - X^{worst}) \quad (3)$$

$$\sigma_i^{t+1} = (1 - \beta)\sigma_i^t + \beta \sqrt{\frac{\sum_{j=1}^k (X^j - \bar{X})^2}{k}} \quad (4)$$

As shown in eq. 3, the next generation average ( $\mu_i^{t+1}$ ) is calculated by a linear combination of two best and the worst search points in population. In eq. 4, the next generation diversity ( $\sigma_i^{t+1}$ ) is updated from the diversity of current good search points. The  $k (= 3 \times C \times K \text{ or } 2 \times C \times K)$  is a number of high-ranking search points. The  $\bar{X}$  is an average value in current populations.

Finally, each updated probabilistic model generates  $C$  new search points and current populations are replaced with them.

Table 1: Test functions

Function Name	Formula ( $S$ is design variable domain)	Dependency	Feature
Sphere	$\sum_{i=1}^n x_i^2 \quad S = [-5.12, 5.12]^n$	No	most basic function
Ellipsoid	$\sum_{i=1}^n (1000^{i-1/n-1} x_i)^2 \quad S = [-5.12, 5.12]^n$		weaker ill-scaled single-peak function
$k$ -tablet	$\sum_{i=1}^k x_i^2 + \sum_{i=k+1}^n (100x_i)^2 \quad (k = n/4) \quad S = [-5.12, 5.12]^n$		stronger ill-scaled single-peak function
Ackley	$20 - 20 \exp(-0.2 \sqrt{1/n \sum_{i=1}^n x_i^2}) + e - \exp(1/n \sum_{i=1}^n \cos(2\pi x_i)) \quad S = [-32.768, 32.768]^n$		weaker multi-peak function
Rastrigin-1.0	$10n + \sum_{i=1}^n ((x_i - 1.0)^2 - 10 \cos(2\pi(x_i - 1.0))) \quad S = [-5.12, 5.12]^n$		stronger multi-peak function
Schwefel	$418.98289n + \sum_{i=1}^n x_i \sin(\sqrt{ x_i }) \quad S = [-512, 512]^n$		stronger multi-peak function there is suboptimal solution in part away
Bohachevsky	$\sum_{i=1}^{n-1} (x_i^2 + 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) - 0.4 \cos(4\pi x_{i+1}) + 0.7) \quad S = [-5.12, 5.12]^n$	Yes	weaker multi-peak function
Griewank	$1.0 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left( \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) \quad S = [-512, 512]^n$		same as above
Ridge	$\sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2 \quad S = [-64, 64]^n$		the good area is along straight line
Rosenbrock-Chain	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1.0)^2) \quad S = [-2.048, 2.048]^n$		strong dependence is between adjoined variables
Rosenbrock-Star	$\sum_{i=1}^{n-1} (100(x_1 - x_i^2)^2 + (x_i - 1.0)^2) \quad S = [-2.048, 2.048]^n$		strong dependence is between $x_1$ and other variables
Schaffer	$\sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.25} \times (\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1.0) \quad S = [-100, 100]^n$		suboptimal solutions are in a circular pattern

## 4. Experimentation

### 4.1. Experimental setting

To evaluate the search capability of RMM, several types of numerical experiment were performed. To compare the performance, PBILc and Distributed Probabilistic Model-Building Genetic Algorithm (DPMBGA)[14] were performed as conventional EDAs. DPMBGA also uses distributed populations similar to RMM. The results of RMM were also compared with those of REX<sup>star</sup>[15], which is a real-coded GA and has high search capability.

In the experiments, continuous numerical test functions were used as summarized in Table 1. There were 12 functions, each of which had different characteristics: dependency, ill-scale, multi-peak, etc. In each function, the dimension  $n$  was set to 20. When the best search point reaches below  $1.0 \times 10^{-7}$  before the number of evaluations reaches  $2 \times 10^6$ , it is judged that the global optimum is discovered.

In these experiments, in addition to evaluating the search capability, several types of parameter of RMM and PBILc were applied and the best parameter was determined. The combination of parameters of RMM is shown in Table 2. The parameter of DPMBGA was set according to the reference article[14] (Table 3). The parameter of PBILc was set equal that of RMM as much as possible (Table 4).

### 4.2. Results

All simulations were performed 20 trials with the same parameters and all combinations of parameters were examined. In the results, among the same algorithms, the simulation that derived the global optimum in 20 times and that had a smaller average evaluations than other simulations is illustrated.

Table 2: Parameters of RMM

Couple of probabilistic model and populations $L$	4, 6, 8, 10
Search points generated by one probabilistic model $C$	5, 10, 15, 20, 25, 30
Cutting rate $K$	0.75, 0.66, 0.50, 0.33, 0.25, 0.20
Maximum learning rate $\beta_1$	0.001, 0.002, $\dots$ , 0.009 0.01, 0.02, $\dots$ , 0.09 0.10, 0.15, $\dots$ , 0.50
Minimum learning rate $\beta_L$	same as above (only $\beta_1 > \beta_L$ )

Table 3: Parameters of DPMBGA

Search points	512
Cutting rate	0.250
Elites	1
Islands	32
Rate of immigration	0.0625
Interval of immigration	5
Archive size	100
Amplification Rate	2.000
Mutation rate	0.050
Independent component analysis	ON (if the function with dependency)

Table 4: Parameters of PBILc

Search points $P$	20, 40, $\dots$ , 300
Cutting rate $K$	The same cutting rate in Table 2
Learning rate $\alpha$	The same maximum learning rate in Table 2

Table 5: Results

Func.	RMM	PBILc	DPMBGA	REX <sup>star</sup>
Sph.	$3.4 \times 10^3$	<b><math>2.8 \times 10^3</math></b>	$1.3 \times 10^5$	$6.9 \times 10^3$
Elli.	$1.0 \times 10^4$	$3.3 \times 10^4$	$1.7 \times 10^5$	<b><math>8.5 \times 10^3</math></b>
$k$ -tab.	<b><math>8.4 \times 10^3</math></b>	$3.0 \times 10^4$	$1.6 \times 10^5$	$1.1 \times 10^4$
Ackl.	<b><math>6.9 \times 10^3</math></b>	$7.7 \times 10^3$	$2.7 \times 10^5$	$1.4 \times 10^4$
Rast.	<b><math>1.2 \times 10^5</math></b>	$1.4 \times 10^5$	$6.0 \times 10^5$	<b><math>1.2 \times 10^5</math></b>
Schw.	<b><math>1.1 \times 10^5</math></b>	$1.4 \times 10^6$	$8.0 \times 10^5$	(No Data)
Boha.	$6.4 \times 10^3$	<b><math>6.2 \times 10^3</math></b>	$1.6 \times 10^5$	$1.5 \times 10^4$
Grie.	<b><math>5.8 \times 10^3</math></b>	$8.6 \times 10^3$	$3.0 \times 10^5$	(No Data)
Ridg.	<b><math>5.4 \times 10^4</math></b>	$9.1 \times 10^4$	$3.5 \times 10^5$	(No Data)
R <sub>Chain</sub>	$1.3 \times 10^5$	–	$9.5 \times 10^5$	<b><math>4.7 \times 10^4</math></b>
R <sub>Star</sub>	$6.3 \times 10^4$	–	$4.5 \times 10^5$	<b><math>5.5 \times 10^4</math></b>
Scha.	<b><math>5.4 \times 10^4</math></b>	$9.4 \times 10^4$	–	$7.7 \times 10^4$

The results are shown in Table 5, where “–” indicates that the algorithm could not find a global optimum. The standard deviations of all results were sufficiently small from 0.7% to 12.1% compared with the average value. The results of REX<sup>star</sup> are from the reference article[15].

Table 5 shows that RMM can find the global optimum more rapidly than PBILc excluding Sphere and Bohachevsky function. These results indicate that the performance of RMM is superior to that of DPMBGA. Compared with RMM and REX<sup>star</sup>, the differences in average evaluations of RMM were from 0.4 to 1.2 times except Rosenbrock-Chain function. Thus, the performance of RMM is almost equivalent to that of REX<sup>star</sup>.

## 5. Discussion

### 5.1. Parameters

Table 6 shows the parameters that yielded the best performance. First, we discuss the maximum and minimum learning rates ( $\beta_1$  and  $\beta_L$ ). In Rastrigin-1.0 and Schwefel functions do not have dependency between design variables where the values of  $\beta_1$  and  $\beta_L$  are close. On the other hand, in Ridge and Rosenbrock-Chain functions have dependency between design variables where those are 10.0 and 12.5 times different. When the values of  $\beta_1$  and  $\beta_L$  are close, the convergence speeds of all populations are rapid because each population is similar. However, the population does not cause the initial convergence in a multi-peak function and the global optimum can be discovered. This is because multiple probabilistic models are estimated for one design variable. When the values of  $\beta_1$  and  $\beta_L$  are markedly different, the diversity of the population is maintained because the convergence speed of each population is different. In functions with a strong dependency, RMM can search effectively because the search with maintenance of population diversity is important.

Next, we focus on the number of probabilistic models ( $L$ ). To determine the effects of the value of  $L$ , all the search points ( $L \times C$ ) are fixed as 120 and the value of  $L = 4, 6, 8, 10$  is changed. Other parameters are as shown in Table 6. The target test functions are Rastrigin-1.0, Schwefel, Ridge and Rosenbrock-Chain. The other experiment

Table 6: The best parameters of RMM

Function	$L$	$C$	$K$	$\beta_1$	$\beta_L$
Sphe.	4	5	0.33	0.25	0.20
Elli.	10	5	0.33	0.45	0.35
$k$ -tab.	8	5	0.33	0.30	0.25
Ackl.	4	5	0.33	0.35	0.15
Rast.	4	5	0.50	0.01	0.009
Schw.	10	10	0.66	0.25	0.20
Boha.	8	5	0.25	0.35	0.20
Grie.	6	5	0.33	0.45	0.25
Ridg.	4	5	0.50	0.50	0.05
$R_{Chain}$	4	5	0.50	0.50	0.04
$R_{Star}$	4	10	0.25	0.50	0.05
Scha.	6	10	0.25	0.30	0.25

Table 7: Number of probabilistic models and search capability

Function	$L = 4$	$L = 6$	$L = 8$	$L = 10$
Rast.	20/20 $1.5 \times 10^6$	20/20 $1.3 \times 10^6$	20/20 $1.2 \times 10^6$	<b>20/20</b> <b><math>1.1 \times 10^6</math></b>
Schw.	8/20 $1.3 \times 10^5$	13/20 $1.4 \times 10^5$	16/20 $1.4 \times 10^5$	<b>19/20</b> <b><math>1.4 \times 10^5</math></b>
Ridg.	<b>20/20</b> <b><math>2.0 \times 10^6</math></b>	0/20 -	0/20 -	0/20 -
$R_{Chain}$	<b>20/20</b> <b><math>9.4 \times 10^5</math></b>	20/20 $1.0 \times 10^6$	20/20 $1.8 \times 10^6$	20/20 $2.8 \times 10^6$

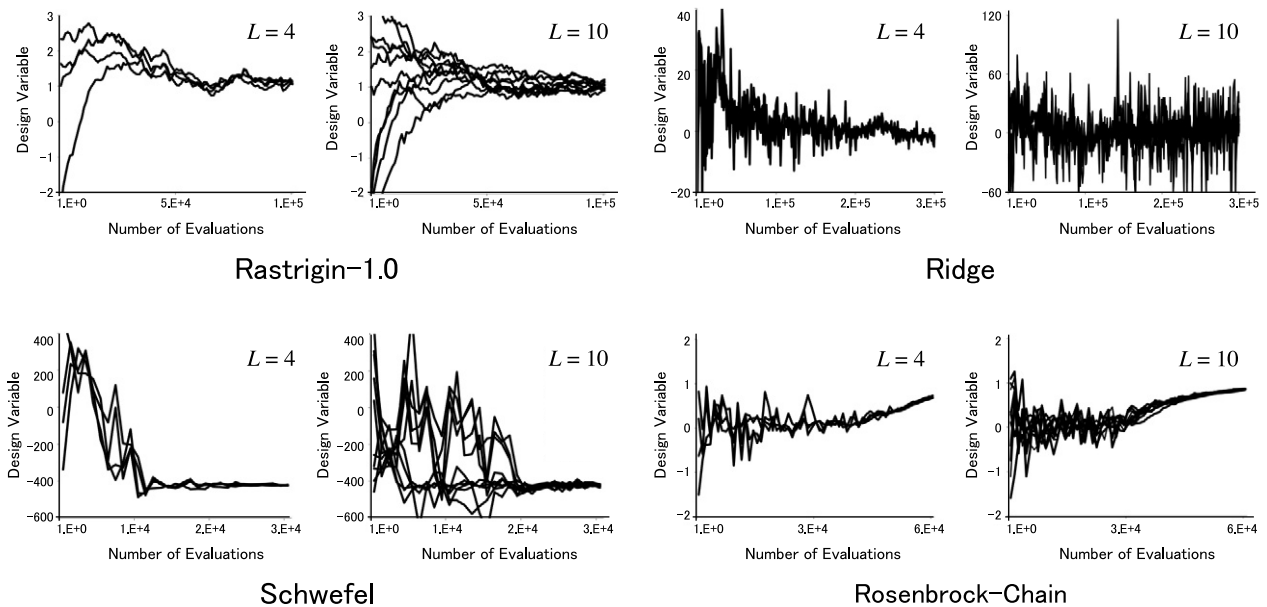
Table 8: The best 10 performance parameters in single-peak functions

Function	Dependency	$L$	$C$	$K$	$\beta_1$	$\beta_L$
Sphe.	No	4, 6	5	0.20~0.33	0.20~0.50	0.10~0.40
Elli.		4~10	5	0.33~0.66	0.25~0.50	0.15~0.45
$k$ -tab.		4~10	5	0.33~0.66	0.30~0.50	0.15~.35
Ridg.	Yes	4	5, 10	0.25~0.66	0.35~0.50	0.03~0.10
$R_{Chain}$		4	5, 10	0.33~0.66	0.40~0.50	0.02~0.05
$R_{Star}$		4	10, 15	0.20~0.33	0.45~0.50	0.03~0.06

environments are the same as Section 4. The results are shown in Table 7. The upper row shows that the number of discovered the global optimum in 20 trials and the lower row shows that the average of the evaluations when the global optimum was discovered. Table 7 indicates that the performance is high in multi-peak functions with a large value of  $L$  and in single-peak functions with a small value of  $L$ . The larger the value of  $L$  is, the higher the probability of searching for a different area is. Thus, it is easy to discover the optimum point because the probability of population convergence in the suboptimal solution is small.

To verify the points mentioned above, the histories of the averages of each probabilistic model with  $L = 4$  and  $L = 10$  were examined. In Fig. 4, the transition of only one design variable among 20 dimensions is described. As the results of multi-peak functions, each population searches for a more different area with larger value of  $L$ . As the results of single-peak functions, all populations search for almost the same area with all value of  $L$ . In single-peak functions, the smaller the value of  $L$  is, the larger the value of search points generated by one probabilistic model  $C$  is when all search points are fixed. Generally, the greater the number of search points is, the better the local search capability is. To solve single-peak functions quickly, it is better to have greater local search capability. Therefore, the search capability increases with small value of  $L$ . For the multi-peak functions, the search capability for various areas is essential. Therefore, the value of  $L$  should be high. However, Table 6 shows that the values of the best value of  $L$  of Ellipsoid and  $k$ -tablet functions, which are single-peak functions, are 10 and 8. These results were in contrast to the above assumption.

The best 10 parameters of all single-peak functions were investigated in Table 8. Table 8 shows that the value of  $L$  is different for functions that do not have dependency between design variables. On the other hand, the smallest value of  $L$  is only 4 for the functions that have dependency between design variables. Moreover, the difference between the values of  $\beta_1$  and  $\beta_L$  of functions that do not have dependency was smaller than that of functions that do have dependency. When the difference between the values of  $\beta_1$  and  $\beta_L$  is small, the convergence of populations is high because each population is similar. If the value of  $C$  is fixed, the amount of calculation is increased in one generation with high value of  $L$ . However, the influence of the values of  $\beta_1$  and  $\beta_L$  on population convergence is larger than the value of  $L$  because the landscape of the functions that do not have dependency is simple. Therefore, it is assumed

Figure 4: History of  $\mu_i$  of each probabilistic model

that the influence of the value of  $L$  on performance is small in simple functions. A global optimum can be found in Ackley, Rastrigin-1.0, Griewank and Schaffer functions if the value of  $L$  is small because these functions do not fall easily into the local optimum because their landscapes have a single peak from a broader standpoint.

Based on these discussions, the value of  $L$  is set to a small value for multi-peak functions and to a large value for single-peak functions.

### 5.2. Effectiveness of multiple learning rates

In this section, we discuss the movement width of the populations of RMM and PBILc. In the function with strong dependency between design variables, such as Ridge or Rosenbrock-Chain function, population convergence occurs easily. Therefore, the population needs to reach the global optimum before population convergence. Eq. 2 determines the movement width of the population generated by the probabilistic models of RMM. While PBILc builds a probabilistic model from only one population, RMM builds a probabilistic model from three populations. Therefore, the movement width of the RMM population is often larger than that of PBILc because RMM uses three populations that differ in their diversity.

To determine the movement width, we examine for Ridge and Rosenbrock-Chain function. The parameters of RMM are shown in Table 6 and those of PBILc are the same values of RMM. The result is shown in Fig. 5. Fig. 5 indicates that the width of the population of RMM is larger than that of PBILc. The width of the population of RMM repeats the increase and decrease, while that of PBILc always decreases. The width of the population of RMM changes more easily than that of PBILc because RMM builds probabilistic model from multiple populations. Moreover, Fig. 5 shows that the width of the population of RMM is large with a high learning rate, while that of PBILc is small with a high learning rate. Eq. 2 and 3 indicate that the width of the population is large with a high learning rate and the variance of the population tends to be small. However, the width of the population is small because the smaller the variance is, the smaller the range of population generated by a probabilistic model is. Therefore, the width of the population of PBILc becomes small when the learning rate is high. In the case of RMM, the variance of the population can be maintained more easily than that of PBILc due to the use of multiple populations. Thus, the width of the population of RMM is larger when the learning rate is high.

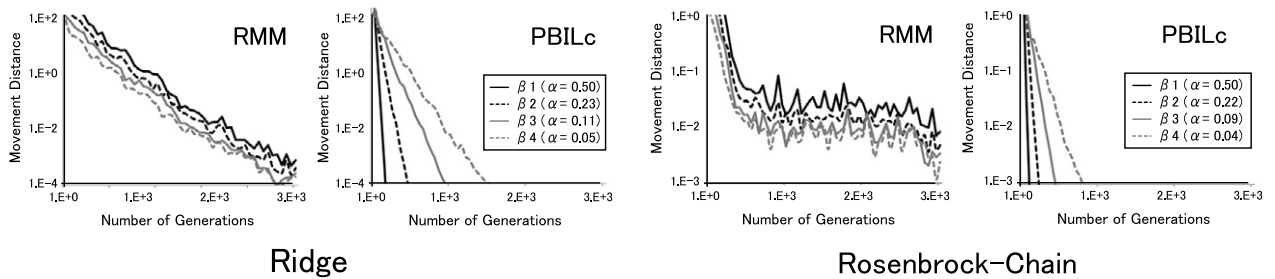


Figure 5: The movement width of population

## 6. Conclusions

In this paper, a new type of real-coded EDA, RMM, was proposed. RMM can form multiple populations with high diversity of variables by setting different learning rates in each probabilistic model. To evaluate its search capability, RMM was applied to find the solution in 12 continuous test functions. The results showed that the search capability of RMM is better than or equivalent to those of existing real-coded EDAs.

The relation of the number of probabilistic models and the search capability of RMM was also discussed. The numerical experiments indicated the following two tendencies. First, RMM with many probabilistic models has stronger search capability for functions with multi-peak landscape structures. When RMM has many probabilistic models, the different points were used for searching in each probabilistic model. For functions with a multi-peak structure, it is important to have a search with a high diversity of solutions to find the global optimum. Second, on the other hand, RMM with a small number of probabilistic models has greater search capability for functions with a single-peak landscape structure. In this case, RMM requires greater local search capability and this helps to find the global optimum.

## Reference

- [1] Mühlenbein, H. and Paass, G.: From recombination of genes to the estimation of distributions I. Binary parameters., *Parallel Problem Solving from Nature*, pp. 178–187 (1996)
- [2] Larranaga, P. and Lozano, J. A. e.: *Estimation of Distribution Algorithm*, Kluwer Academic Publishers (2002)
- [3] Pelikan, M., Goldberg, D. E., and Lobo, F.: A survey of optimization by building and using probabilistic models., *Technical report, Computational Optimization and Applications*, 21(1), 5-20. Also IlliGAL Report, No. 99018 (2002)
- [4] Shumeet B.: *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*, Technical Report CMU-CS-94-163, Carnegie Mellon University (1994)
- [5] Jeremy S. De Bonet, Jr. Charles L. Isbell, and Paul Viola.: *MIMIC: Finding Optima by Estimating Probability Densities*. *Advance in Neural Information Processing Systems*, Vol. 9, pp. 424-430 (1997)
- [6] Martin P., David E. Goldberg, and Erick Cantú-Paz. *BOA: The Bayesian Optimization Algorithm*. *Proc. Genetic and Evolutionary Computation Conference 1999*, pp. 525-532 (1999)
- [7] Stephan Rudlof and Mario Köppen.: *Stochastic Hill Climbing with Learning by Vectors of Normal Distributions*. *Online Workshop on SOFT COMPUTING* (1996)
- [8] P. Bosman and D. Thierens.: *Continuous Iterated Density Estimation Evolutionary Algorithms Within The IDEA Framework*. *Proceedings of the OBUPM Workshop at the 2000 Genetic and Evolutionary Computation Conference*, pp. 197-200 (2000)
- [9] Sebag, M. and Ducoulombier, A.: *Extending population-based incremental learning to continuous search spaces*, *Proc. of the Parallel Problem Solving from Nature - PPSN V*, pp. 418–427 (1998)
- [10] Goldberg, D. E.: *Optimization and Machine Learning*, Addison-Wesley Publishing Company (1989)
- [11] V. Scott G. and Darrell W. : *Serial and Parallel Genetic Algorithms as Function Optimizers*, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 177-183 (1993)
- [12] Higo, T. and Takadama, K.: *Maintaining Multiple Populations with Different Diversities for Evolutionary Optimization Based on Probability Models*, *Information and Media Technologies*, Vol. 3, No. 2, pp. 362–374 (2008)
- [13] Higo, T.: *Research on the Importance Sampling Method for Evolutionary Algorithms Based on Probability Models*, Ph.D. thesis, Tokyo Institute of Technology (2008)
- [14] Tomoyuki, H., Mitsunori, M., Masaki, S., Hisashi, S., Shigeyoshi, T., Jack, D.: *Distributed Probabilistic Model-Building Genetic Algorithm*, *Proceedings of the 2003 Genetic and Evolutionary Computation Conference* (2003)
- [15] Shigenobu, K.: *The Frontiers of Real-coded Genetic Algorithms*, *Transactions of the Japanese Society for Artificial Intelligence* Vol. 24 No. 1 SP-N, pp. 147–162 (2009)