Evolutionary Algorithm Using Marginal Histogram Models in Continuous Domain

Shigeyoshi Tsutsui, Martin Pelikan,
and David E. Goldberg

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue
Urbana, Illinois 61801 USA
Phone: 217-333-0897
Fax: 217-244-5705
Web: http//www-illigal.ge.uiuc.edu

Evolutionary Algorithm Using Marginal Histogram Models in Continuous Domain

Shigeyoshi Tsutsui, Martin Pelikan, and David E. Goldberg

Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
117 Transportation Building,
{shige, pelikan, deg}@illigal.ge.uiuc.edu

Abstract: Recently, there has been a growing interest in developing evolutionary algorithms based on probabilistic modeling. In this scheme, the offspring population is generated according to the estimated probability density model of the parents instead of using recombination and mutation operators. In this paper, we propose an evolutionary algorithm using a marginal histogram to model the parent population in a continuous domain. We propose two types of marginal histogram models: the fixed-width histogram (FWH) and the fixed-height histogram (FHH). The results showed that both models worked fairly well on test functions with no or weak interactions among variables. Especially, FHH could find the global optimum with very high accuracy effectively and showed good scale-up with the problem size.

1. Introduction

Genetic Algorithms (GAs) [Goldberg 89] are widely used as robust searching schemes in various real world applications, including function optimization, optimal scheduling, and many combinatorial optimization problems. Traditional GAs start with a randomly generated *population* of candidate solutions (individuals). From the current population, better individuals are selected by the *selection* operators. The selected solutions produce new candidate of solutions by being applied *recombination* and *mutation* operators.

Recombination mixes pieces of multiple promising sub-solutions (*building blocks*) and composes solutions by combining them. GAs should therefore work very well for problem that can be somehow decomposed into subproblems. However, fixed, problem-independent recombination operator often break the building blocks or do not mix them effectively [Pelikan 00b].

Recently, there has been a growing interest in developing evolutionary algorithms based on probabilistic models. In this scheme, the offspring population is generated according to the estimated probabilistic model of the parent population instead of using traditional recombination and mutation operators. The model is expected to reflect the problem structure, and as a result it is expected that this approach provides more effective mixing capability than recombination operators in the traditional GAs. These algorithms are called the probabilistic model-building genetic algorithms (PMBGAs). In PMBGAs, better individuals are selected from initially randomly generated population like in the standard GAs. Then, the probability distribution of the selected set of individuals

is estimated and new individuals are generated according to this estimate, forming candidate of solutions for the next generation. The process is repeated until the termination conditions are satisfied.

According to [Pelikan 00], PMBGAs can be classified into three classes depending on the complexity of models they use; (1) no interactions, (2) pairwise interactions, and (3) multivariate interactions. In models with no interactions, interactions among variables are treated independently. Algorithms in this class work well on problems which have no interactions among variables. These algorithms include the population based incremental learning (PBIL) algorithm [Baluja 94], the compact genetic algorithm (cGA) [Harik 98], and the univariate marginal distribution algorithm (UMDA) [Mühlenbein 96]. In pairwise interactions, some pairwise interactions among variables are considered. These algorithms include the mutual-information-maximization input clustering (MIMIC) algorithm [De Bonet 97], the algorithm using dependency trees [Baluja 97]. In models with multivariate interactions, algorithms use models that can cover multivariate interactions. Although the algorithms require increased computational time, they work well on problems which have complex interactions among variables. These algorithms include the extended compact genetic algorithm (ECGA) [Harik 99] and the Bayesian optimization algorithm (BOA) [Pelikan 99, 00a].

Several attempts to apply PMBGAs in continuous domain have been made. These include continuous PBIL with Gaussian distribution [Sebag 98] and a real-coded variant of PBIL with iterative interval updating [Servet 97]. In [Gallagher 99], the PBIL is extended by using a finite adaptive Gaussian mixture model density estimator. That allows the algorithm to deal with multimodal distribution. All above algorithms do not cover any interactions among the variables. In the estimation of Gaussian networks algorithm (EGNA) [Larranaga, 99], a Gaussian network is learned to estimate a multivariate Gaussian distribution of the parent population. In [Bosman 99], two density estimation models, i.e., the normal distribution (normal kernel distribution and normal mixture distribution), and the histogram distribution are discussed. These models are intended to cover multivariate interaction among variables. In [Bosman 00a], it is reported that the normal distribution models have showed good performance. In [Bosman 00b], a normal mixture model combined with a clustering technique is introduced to deal with non-linear interactions.

In this paper, we propose an evolutionary algorithm using marginal histograms to model promising solutions in a continuous domain. We propose two types of marginal histogram models: the fixed-width histogram (FWH) and the fixed-height histogram (FHH). The results showed both models worked fairly well on test functions which have no or weak interactions among variables. FHH could find the global optimum with high accuracy effectively and showed good scale-up behavior. Section 2 describes the two types of marginal histogram models. In Section 3 empirical analysis is given. Future work is discussed in Section 4. Section 5 concludes the paper.

2. Evolutionary Algorithms using Marginal Histogram Models

This section describes how marginal histograms can be used to (1) model promising solutions and (2) generate new solutions by simulating the learned model. We consider two typed of histograms: the fixed-width histogram (FWH) and the fixed-height histogram (FHH).

2.1 General description of the algorithm

The algorithm starts by generating an individual population of candidate solutions at random. Promising solutions are then selected using any popular selection scheme. A marginal histogram model for the selected

solutions is constructed and new solutions are generated according to the built model. New solutions replace some of the old ones and the process is repeated until the termination criteria are met. Therefore, the algorithm differs from traditional GAs only by a method to process promising solutions in order to create the new ones. The pseudo-code of the algorithm follows:

0. Set the generation counter $t \leftarrow 0$
1. Generate the initial population $P(0)$ randomly.
2. Evaluate functional value of the vectors (individuals) in $P(t)$.
3. Select a set of promising vectors $S(t)$ from $P(t)$ according the functional values.
4. Construct a marginal histogram model $M(t)$ according to the vector values in $S(t)$.
5. Generate a set of new vectors $O(t)$ according to the marginal histogram model $M(t)$.
6. Evaluate the new vectors in $O(t)$.
7. Create a new population $P(t+1)$ by replacing some vectors from $P(t)$ with $O(t)$.
8. Update the generation counter, $t \leftarrow t + 1$.
9. If the termination conditions have not been satisfied, go to step 3.
10. Obtain solutions in $P(t)$.
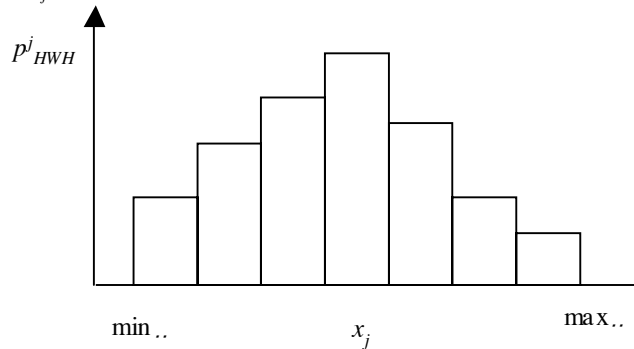
2.2 Marginal fixed-width histogram (FWH)

The histogram is the most straightfoward way to estimate probability density. The possibility of using histograms to model promising solutions in continuous domains was discussed in [Bosman 99].

2.2.1 Model description

In the FWH, we divide the search space $[\min_j, \max_j]$ of each variable x_j $(j=1,...n)$ into H $(h_j = 0, 1,..., H\text{-}1)$ bins. Then the estimated density $p^j_{\text{FWH}}[h]$ of bin h_j is:

$$P^j_{\text{FWH}}[h] = \left|\left\{V[i][j] \;\middle|\; i \in \{0,1,...,N-1\} \wedge h \leq \frac{V[i][j] - \min_j}{\max_j - \min_j} H < h+1\right\}\right| \Big/ N \qquad (1)$$

where $V[i][j]$ is the value of variable x_j of individual i in the population and N is the population size. Since we consider a marginal histogram, the total size of all the bins should be $H \times n$. Figure 1 shows an example of marginal FWH on variable x_j.

Figure 1: Marginal fixed-width histogram (FWH)

4

The bin width ε can be determined by choosing appropriate number of bins depending on the required precision for a given problem. For multimodal problems, we need consider some maximum bound on ε value. Let us consider the worst case scenario where the peak with the global optimum of h_1 has smaller width w_1 than ε and the second best local optimum has a much wider peak of width w_2 (Figure 2). Moreover, let us assume that all solutions in each peak are very close to the peak (as it should be the case later in the run). In this case, after the random sampling at generation 0, the mean fitness of individuals in the bin where the second best solution is located is h_2 and mean fitness of individuals in the bin where the best solution is located is $h_1 \times w_1/(2\varepsilon)$. With most selection schemes, if $h2 > h_1 \times w_1/(2\varepsilon)$ holds, then the probability density of the bin where the second best solution is located increases more than the probability density of bin where the best solution is located. As a result, the algorithm converges to the second best optimum instead of the global one. To prevent this, the following relationship must hold:

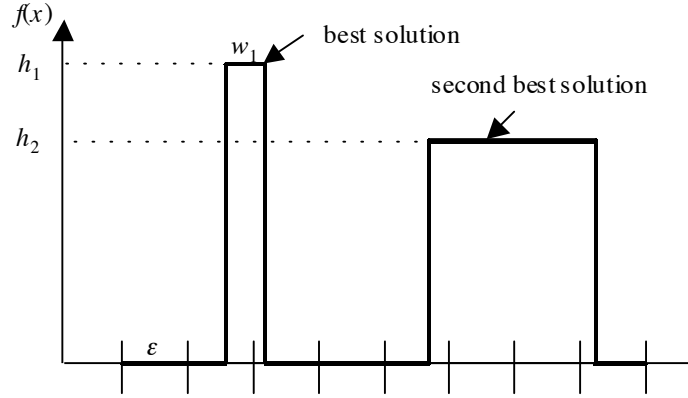$$\varepsilon < \frac{w_1 \times h_1}{2 \times h_2} \; .$$ (2)

Figure 2 Fitness function with multimodality

2.2.2 Sampling methods

In the FWH, new individuals are generated as follows: first a bin is selected according to the probability given by Eq. (1). Then an individual is generated by generating a number from the bin with uniform distribution. This is repeated until all individuals are obtained. The simplest method to sample each bin is to use a roulette wheel (RW). However, RW has a certain amount of stochastic sampling error. Let S be the total number of samples. Then the distribution of R, the number of samples from bin h_j, is

$$P_S^{hj}(R) = \binom{S}{R} \times \left(P_{FWH}^j[h]\right)^R \left(1 - P_{FWH}^j[h]\right)^{S-R}, \; R = 0,1,...,S$$ (3)
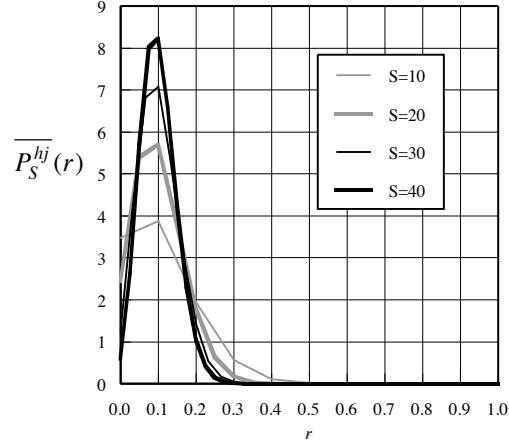
and the expected value of R is

$$\overline{R} = S \times P_{FWH}^j[h] \; .$$ (4)

Let us define a "normalized number of samples" $r = R/S$, then distribution of r is obtained as

$$\overline{P_S^{hj}}(r) = S \times P_S^{hj}(R/S), r = [0,1] \; .$$ (4)

Figure 3 shows an example of distribution r for various number of S. $F_{FWH}^j[h] = 0.1$ is assumed. Then the expected value of r should be

5

Figure 3: An example of distribution r for various values of S

$$\overline{r} = \overline{R}/S$$
$$= S \times P_{\text{FWH}}^{j}[h]/S \tag{5}$$
$$= 0.1.$$

In Figure 3, we can make sure the existence of the stochastic sampling error, although as S becomes larger, the probability density $\overline{P_{S}^{hj}}(r)$ around $r = 0.1$ increases.

One well known sampling method to reduce this sampling error is Baker's stochastic universal sampling (SUS), which was proposed for proportional selection operator [Baker 87]. In this method, the integer part of the expected number $\overline{R}$ is deterministically sampled and the fraction part of $\overline{R}$ is stochastically sampled without duplicate. We extend this method to sample from marginal histogram models. We call this method *E-SUS* (extended SUS).

A pseudo C code of E-SUS for FWH is described in Figure 4. To apply this method, The variable "expected" in line 13 is the expected sampling number for bin h of variable x_j. For instance, if the probability of a particular bin is 0.155 and we want to sample 100 individuals, we would expect 15.5 copies of representatives of the bin.

```
1   // S: sampling individual number, n: number of variables, H: number of bins
2   // p[j][h]: probability density of bin h of variable x_j
3   // l[j][h]: left edge position of of bin h of variable x_j
4   // Rand(): generate random number from [0, 1.0]
5   V[S][n]; // array for sampled vectors
6   xh[S]={0,1,2,……S-1}; // array for random permutation of bin position
7   for(int j=0; j<n; j++){
8       Shuffle(xh); // get a permutation xh[] by shuffling for each parameter
9       double ptr = Rand();
10      double sum = 0.0;
11      int k = 0;
12      for(int h=0; h<H; h++){
13          double expected = p[j][h]*S;
14          for(sum += expected; sum>ptr; ptr++)
15              V[xh[k++]][j] = l[j][h]+(l[j][h+1]-l[j][h])*Rand();
16      }
17  }
```

Figure 4: Pseudo C code of the E-SUS for FWH

The presented algorithm will generate 15 copies and then another copy with probability 0.5. To obtain a random combination among variables, we must shuffle the sampling sequence for each dimension (line 8). In the experiments in Section 3, we use both the RW and the E-SUS.

2.3 Marginal fixed-height histogram (FHH)

2.2.1 Model description

In the FWH each bin has the same width. On the other hand, in the fixed-height histogram (FHH) each bin has the same height. That means that each bin contains the same number of points. See Figure 5 for an example. The important feature of the FHH is that the bins in dense regions are narrower and thus the accuracy of modeling in the important regions increases. In context of evolutionary algorithms, the width of bins around high peaks decreased as more sample points are located in these areas. Since the probability of generating a point from each bin in the FHH is the same, we expect both the density as well as the accuracy improve in promising region of the search space. This is the distinguished characteristics of the FHH. Figure 6 shows this more clearly. Figure 6 (a) is an example of the FHH for a population which has a normal distribution $N(0, 1)$ for variable x_j in the range [-5, 5] as shown in Figure 6 (b). Twenty (20) bins are used. In this example, more points are sampled around $x_j = 0.0$ because the bin width around $x_j = 0.0$ is narrower and therefore there are more bins for the same area.
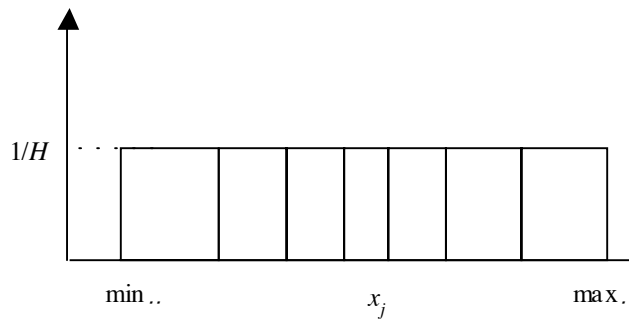
Figure 5: Marginal fixed height histogram (FHH)
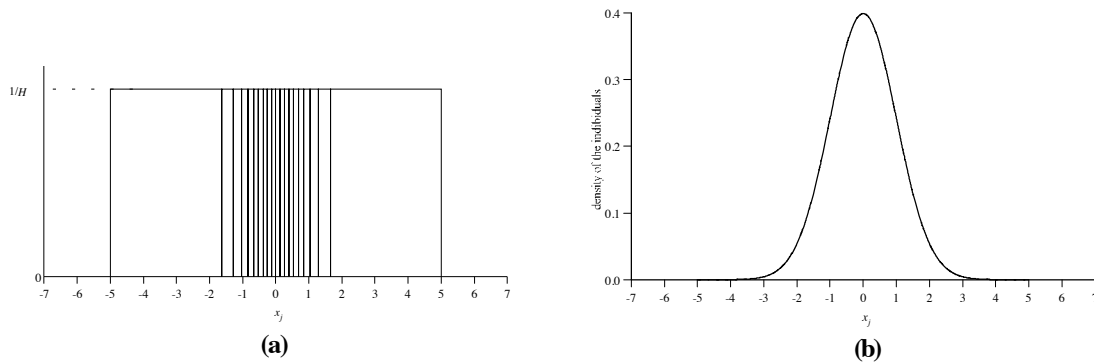
Figure 6: An example of the FHH of normal density

7

3.3.2 Sampling methods

Sampling method for the FHH is basically the same as that of FWH except the probability density for FHH has the same value $1/H$ for all bins. The pseudo C code of the E-SUS for FHH is the same with that for FWH except line 13 in Figure 4. The line 13 for the FHH should be:

<center>"double expected = (double)S/H;".</center>

3. Empirical Study

To evaluate the marginal histogram models proposed in Section 2, we ran two histogram models using four test functions. The experimental methodology, test functions used, and experimental analysis are described in this section.

3.1 Experimental methodology

3.1.1 Evolutionary model

The basic evolutionary model we used in these experiments is similar to that of $(\mu+\lambda)$-ES [Schewefel 95]. Let the population size be N, and let it, at time t, be represented by $P(t)$. The population $P(t+1)$ is produced as follows (Figure 7):

1. Marginal histogram model is developed from $P(t)$
2. $K \times N$ new individuals are sampled according to the built model
3. The new individuals are evaluated
4. Individuals in $P(t)$ and $K \times N$ new individuals are ranked and the best N individuals are selected forming $P(t+1)$.

    We used $K = 1$ in our experiments in Sub-sections 3.2.1 and 3.2.2.

3.1.2 Test Functions

The test functions we used are summarized in Table 1, which includes the 20-variable two peaks function F_{TwoPeaks}, the 20-variable Rastrigin function ($F_{\text{Rastrigin}}$), the 10-variable Griewank function F_{Griewank}, and the 5-variable Schfewel function F_{Schfewel}.

    Function F_{TwoPeaks} is the sum of a sub-function similar to the commonly used deceptive function in binary domain and has a broader and lower peak, and a more narrow and higher peak for each variable x_j as shown in Figure 8. This function is designed to see the mixing capability and the scale-up behavior of the algorithm. This function has no interactions among variables. Other three function are commonly used in the evolutionary computation literature. Rastrigin function has also no interactions among variables. Griewank function has weak interactions
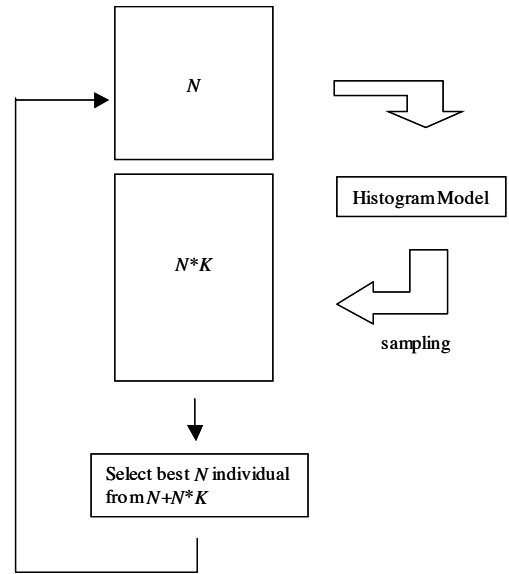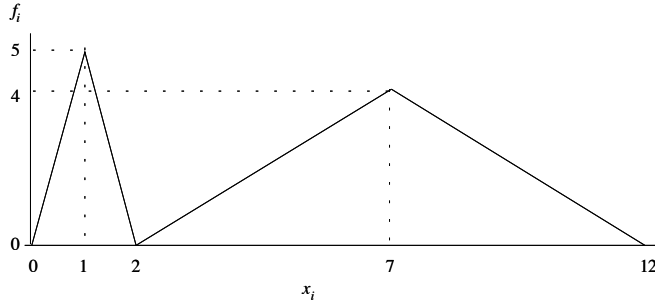

Figure 7: Evolutionary model

Table 1: Test functions

Function	Domain	Interactions*	H^{**}
$F_{\text{Two-peaks}} = n \times 5 - \sum_{i=1}^{n} f_i$	$[0, 12]^n$, $n = 20$	non	120
$F_{\text{Rastrigin}} = n \times 10 + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i))$	$[-5, 5]^n$, $n = 20$	non	100
$F_{\text{Griewank}} = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n}\left(\cos\left(x_i / \sqrt{i}\right)\right)$	$[-5, 5]^n$, $n = 10$	weak	100
$F_{\text{Schwefel}} = \sum_{i=2}^{n}\left[\left(x_1 - x_i^2\right)^2 - \left(x_i - 1\right)^2\right]$	$[-2, 2]^n$, $n = 5$	medium	40

* Interactions among variables ** Number of bins in each variable

Figure 8: A sub-function of function F_{TwoPeaks}

among variables. Schfewel function has medium-scale interactions among variables.

As to the number of bins (H) for each variable, we set H so that the initial bin width ε is 0.1. For example, for function F_{TwoPeaks}, the number of bins for each variable is $12.0/0.1 = 120$.

3.2 Analysis of results

The experiments in this section focus on (1) convergence properties of the FWH and the FHH with resolution of bin width 0.1, (2) convergence properties of the FHH with high precision, and the (3) scale-up behavior of the FHH on function F_{TwoPeaks} for problem size ranging from 10 to 100 variables.

3.2.1 Convergence properties of the FWH and the FHH with resolution of 0.1

The accuracy of the FWH is at most ε (i.e., 0.1). In our experiments, we evaluated the algorithms by measuring the number #OPT of runs in which the algorithm succeeded in finding the global optimum and the mean number of function evaluations (MNE) to find the global optimum in those runs where it found the optimum. We defined the successful detection of the solution as being within $\pm \varepsilon$ of the actual optimum point. Let $(o_1,...,o_n)$ be the optimal solution of a function. Then, if all values of variables $(x_1,...,x_n)$ of the best individual are within the range $[(o_j-\varepsilon), (o_j+\varepsilon)]$ for all j, we assumed the algorithm to have found the optimal solution. Although the width

of each bin for FHH changes during the evolution, we used the same criteria for the FHH in this experiment. Twenty (20) runs were performed. In each run, the initial population $P(0)$ was randomly initialized in the original search space. Each run continued until the global optimum was found or a maximum of 200,000 function evaluations was reached.

Table 2 shows the results of this experiment. We used both the RW and the E-SUS as the sampling method. We also show the results with BLX-α [Eshelman 93], a traditional two-parent crossover operator in continuous domain for comparison.

First, let us see the difference in the performance of the two sampling methods, RW and E-SUS. On almost all experiments with some exceptions, the performance with sampling method E-SUS, which has a smaller stochastic sampling error, is much better than the performance with sampling method RW, which has a larger stochastic sampling error. For example, on function F_{TwoPeaks}, the FWH with RW (shown as FWH/RW) found optimal solution 20 times with the population size of 600. The MNE is 14,620.9. On the other hand, the FWH with E-SUS (shown as FWH/E-SUS) found optimal solution 20 times with the population size of 300 and the MNE is 7,178.3, almost the half of the FWH/RW. Next, let us see the difference of performance between the FWH and the FHH. All of the results of the FHH both with RW and E-SUS showed better performance than FWH. For example, on function F_{TwoPeaks}, FHH/RW found optimal solution 20 times with population size of 300 and the MNE is 8,321.8. This is much better than the performance of FWH/RW.

Finally, BLX-α operator showed relatively good performance on functions F_{Griewank} and F_{Schfewel}. But it showed poorer performance on functions $F_{\text{Rastrigin}}$ and F_{TwoPeaks}. Especially on function F_{TwoPeaks}, BLX-α could not found optimal solution at all showing very poor mixing capability.

Table 2: Convergence property of the FWH and the FHH with resolution of 0.1

Model	Function	n	Population size											
			100		200		300		400		600		800	
			#OPT	MNE	#OPT	MNE	#OPT	MNE	#OPT	MNE	#OPT	MNE	#OPT	MNE
FWH/RW	$F_{\text{Two-peaks}}$	20	0	-	1	4,938.0	7	7,852.4	16	9,854.1	20	**14,620.9**	20	18,808.0
	$F_{\text{Rastrigin}}$	20	0	-	1	7,851.0	6	9,204.0	12	13,154.2	20	**19,396.6**	20	24,762.7
	F_{Griewank}	10	0	-	1	6,527.0	4	9,515.8	14	11,908.6	16	17,884.3	20	**23,198.2**
	F_{Schwefel}	5	13	1,188.9	16	1,927.4	19	3,175.5	20	**3,900.2**	20	5,282.0	20	6,828.4
FWH/E-SUS	$F_{\text{Two-peaks}}$	20	1	2,562.0	14	4,926.5	20	**7,178.3**	19	9,609.8	20	14,016.6	20	18,537.9
	$F_{\text{Rastrigin}}$	20	1	3,478.0	13	6,770.6	20	**9,720.2**	20	12,616.4	20	18,518.5	20	24,113.5
	F_{Griewank}	10	2	2,779.0	7	5,783.3	15	9,220.9	19	12,439.8	19	17,517.2	19	22,502.3
	F_{Schwefel}	5	16	1,342.6	19	1,962.3	19	3,302.9	20	**4,012.7**	20	5,924.8	20	7,597.9
FHH/RW	$F_{\text{Two-peaks}}$	20	20	17,952.4	20	9,443.2	20	**8,321.8**	20	10,305.9	20	14,950.5	20	19,442.9
	$F_{\text{Rastrigin}}$	20	20	50,099.8	20	46,139.8	20	21,933.1	20	**17,566.2**	20	22,847.3	20	28,990.9
	F_{Griewank}	10	17	5,445.1	19	6,561.2	18	9,521.2	16	14,076.5	20	**17,233.4**	19	22,909.5
	F_{Schwefel}	5	20	**2,758.9**	20	4,644.6	20	3,821.2	20	5,969.8	20	6,257.1	20	7,101.6
FHH/E-SUS	$F_{\text{Two-peaks}}$	20	20	9,800.4	20	**5,405.8**	20	7,679.9	20	9,982.4	20	14,380.3	20	19,180.1
	$F_{\text{Rastrigin}}$	20	20	16,515.7	20	**8,004.2**	20	10,177.6	20	12,690.1	20	18,177.4	20	24,879.6
	F_{Griewank}	10	18	3,187.4	18	5,680.6	20	**8,199.6**	20	10,962.5	20	15,782.6	20	20,614.3
	F_{Schwefel}	5	20	3,526.7	20	**2,994.3**	20	3,483.3	20	5,020.2	20	6,234.8	20	8,829.2
BLX-α	$F_{\text{Two-peaks}}$	20	0	-	0	-	0	-	0	-	0	-	0	-
	$F_{\text{Rastrigin}}$	20	0	-	1	28,521.0	4	47,959.8	11	62,776.0	17	94,994.9	20	**124,825.9**
	F_{Griewank}	10	20	**3,316.3**	20	6,347.7	20	9,037.2	20	11,947.3	20	17,483.0	20	23,209.9
	F_{Schwefel}	5	20	**3,305.3**	20	3,861.3	20	5,397.8	20	7,141.0	20	10,158.8	20	12,688.0

3.2.2 Convergence properties of the FHH

As described in Section 2.2, the bin width of FHH around the global optimum becomes narrower as the evolution proceeds. As a result, the FHH can find much more accurate solution. Figure 9 shows a typical change of bin positions of variable x_1 in a run with FHH/RW, which success to converge to the global optimum, on function $F_{Ranstrigin}$. We can observe that the FHH gathers bin positions to the converged point as evolution proceeds. The population size in the above experiment was set to 800.

To see the convergence properties of the FHH, we ran FHH model on the test functions in Table 1. Population size of 800 was used for all functions in this experiments. Figure 10 shows the changes of mean functional value of the populations until function evaluations reached 100,000. Ten (10) runs has been performed.

With RW, the FHH converged to local optima several times on functions $F_{TwoPeaks}$, $F_{Rastrigin}$, and $F_{Griewank}$. However, with E-SUS the FHH converged to global optima 9 out of 10 times on function $F_{Rasrigin}$. On function $F_{Schfewel}$, the FHH failed to converge to the global optimum.

Functions $F_{TwoPeaks}$ and $F_{Rastrigin}$ have no interactions among variables and function $F_{Griewank}$ has weak interactions among variables. Function $F_{Schfewel}$ has medium-scale interactions among variables. With a marginal histogram, it is clearly difficult to solve problems which have medium-scale or stronger interactions among variables.

Results of the BLX-α operator has also shown in the figure. BLX-α performed quite poorly on function $F_{TwoPeaks}$. On functions $F_{Rastrigin}$ and $F_{Griewank}$, it worked similar to the FHH with E-SUS. On function $F_{Schfewel}$ it worked better than the FHH and slowly but gradually converged to the global optimum.
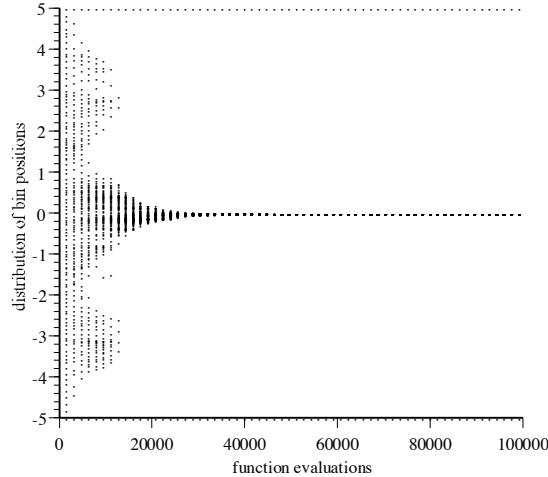

Figure 9: Typical changes of bin positions in parameter x_1 with FHH/RW on function $F_{Griewank}$

4.2.3 Scale-up behavior of the FHH

This section focus on the scale-up behavior of the FHH with E-SUS using function $F_{TwoPeaks}$ with the number of variables from 10 to 100 with step 10. We used a different evolutionary model, so called MGG [Satoh 96]. The optimum of function $F_{TwoPeaks}$ is located at (1.0,1.0,...,1.0). We assumed the optimum solution was found if the best individual in the population has a vector value of
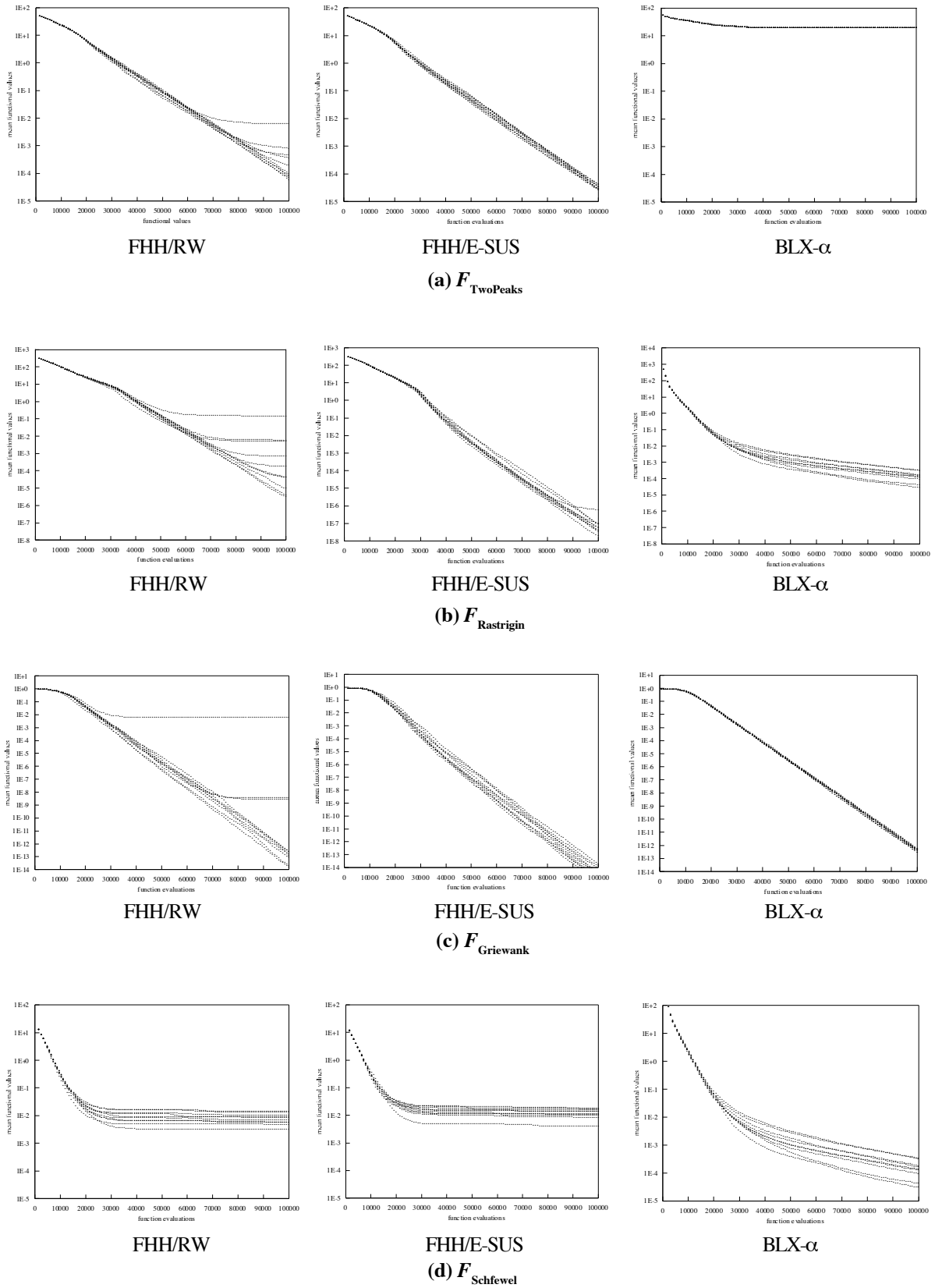
(a) F_{TwoPeaks}

(b) $F_{\text{Rastrigin}}$

(c) F_{Griewank}

(d) F_{Schfewel}

**Figure 10: Convergence property of the FHH**

12

$$\underset{i=1,..,n}{\forall} \quad 1.0 - 0.005 \leq x_i < 1.0 + 0.005 \quad . \tag{6}$$

We did experiment for each number of parameters increasing population size from 200 with step 200 until optimal solution was obtained 10 times in 10 runs.

Figure 11 shows the population size needed to find optimal solution 10 times. The population size increased with $O(n)$. Figure 12 shows the mean number of function evaluations (MNE). The MNE increased with almost $O(n^{1.56})$.
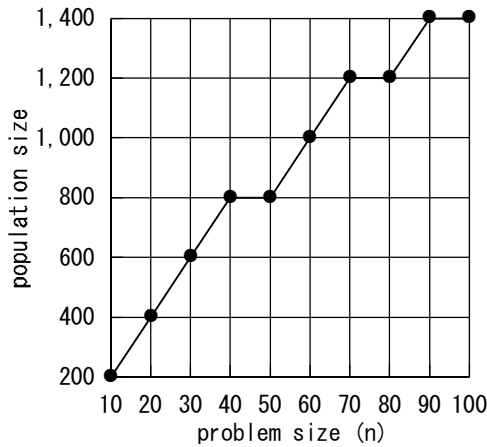
Figure 11: Population size needed to find optimal 10 times in 10 runs
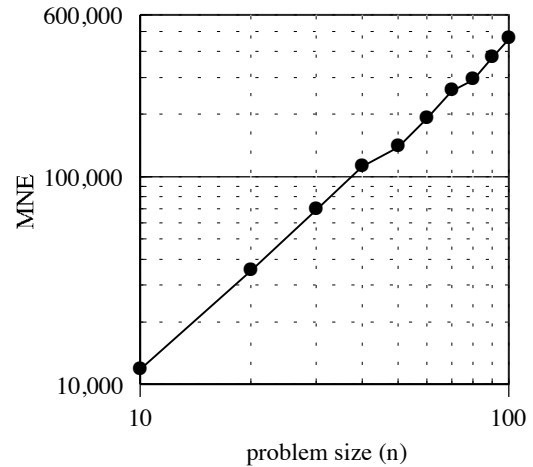
Figure 12: Mean number of function evaluations (MNE) needed to find optimal 10 times in 10 runs

4. Future work

As was observed in Section 3, the marginal histogram models showed good performance on functions which have no or weak interactions among variables. Especially the FHH/E-SUS has a good feature. It can search the solution with high precision and have a good scale-up capability. However, the marginal histogram did not work on functions which have a medium level of linkage among variables. Of course, it will not work well on functions which have strong interactions among variables.

The natural extension of the marginal histogram models is the multivariate histogram models. However, an exponentially growing amount of bins in the form of H^n is required [Bosman 99]. However, if we can divide a problem into some sub-problems of size m ($m<n$) using multivariate interaction model such as ECGA [Harik 98] or BOA [Pelikan 99, 00a], we may construct the FHH for each such sub-problem. Many types of FHH for such sub-problems can be considered. For example, let (x_1, x_2) be the parameters of a sub-problem which has a strong interaction between variables x_1 and x_2 and let us assume we can identify the interaction on-line or off-line. Figure 13 is an example of forming a kind of FHH for the sub-problem. In the figure, we first divide the population into two rectangles on x_1 axis so that each rectangle has the same number of individuals. Then, we divide each rectangle into two smaller rectangles on x_2 axis so that again each smaller rectangle has the same

13

number of individuals. We repeat this division recursively while each small rectangle has a number of individuals greater than or equal to defined constant value of K_{min}. Each obtained rectangle corresponds to bin in the marginal histogram. Each rectangle is sampled with equal probability. Individuals are generated by uniform sampling in each rectangle. Figure 14 provides results of both the marginal FHH/RW and the recursively divided FHH/RW on the sub-function of original two dimensional Rosenbrock function

$$F_{\text{Rosenbrock}} = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2 ,$$
$$- 2.048 \le x_1, x_2 < 2.047. \tag{7}$$

which has a strong interaction between variables x_1 and x_2. Population size of 3000 and K_{min} value of 2 are used. From this figure, we may see this FHH is working well on this function. Exploring more general FHH models remains a topic for future study.
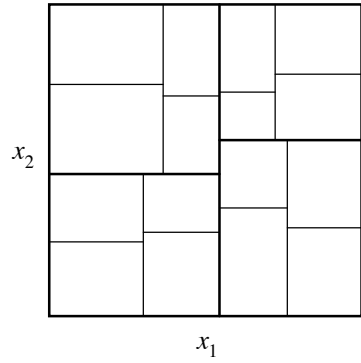

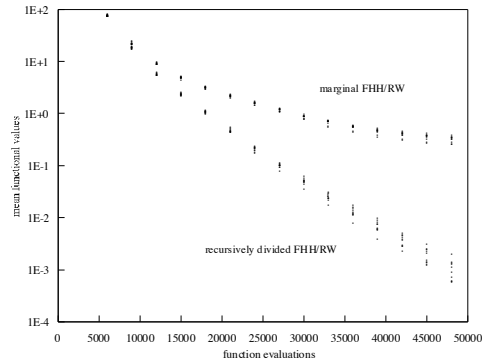**Figure 13: Recursively divided FHH in space x_1, x_2**   **Figure 14: Convergence processes on $F_{\text{Rosenbrock}}$**

5. Conclusions

In this paper, we have proposed an evolutionary algorithm using a probabilistic model with marginal histograms in continuous domain. Two types of marginal histogram models were used: the FWH (fixed-width histogram) and the FHH (fixed-height histogram). As the sampling methods, we used both the RW (roulette wheel) and the E-SUS (extended stochastic universal sampling) which is extended from a popular proportional selection operator in GAs.

Both the FWH and the FHH showed good performance on functions which have no or weak interactions among parameters. Comparing two sampling methods, the performance with the E-SUS, which has a smaller stochastic sampling error, showed much better than the performance with the RW, which has a larger stochastic sampling error. The FHH/E-SUS (FHH with E-SUS) showed very good performance. It could find the solutions with a very high precision which is being evolved along with individual solutions. The FHH/E-SUS had also a good scale-up behavior and it mixes solutions very efficiently.

However, the marginal histograms did not work well on function which have a medium (or larger) level of interactions among variables. Introducing models considering interactions among variables as discussed in Section 4 remains a topic for future work. In this paper, although we focused on evolutionary algorithm using probabilistic model with histogram model, it is reported that the probabilistic model using the normal mixture

model shows good performance [Bosman 00b]. To combine the histogram model with such a model might enhance the performance of those models because the histogram model has a high mixing capability of sub-solutions.

Acknowledgments

References

[Baker 87] Baker J. E: Reducing bias and inefficiency in the selection algorithm, *Proc. of the 2nd International Conference on Genetic Algorithms*, pp. 14-21 (1987).

[Baluja 94] Baluja, S.: Population-based incremental learning: A method for interacting genetic search based function optimization and coemptive learning, *Tech. Rep. No. CMU-CS-94-163*, Carnegie Mellon University (1994).

[Baluja 97] Baluja, S. and Davies: Using optimal dependency-trees for combinatorial optimization: learning the structure of the search space, *Tech. Rep. No. CMU-CS-97-107*, Carnegie Mellon University (1997)

[Bosman 99] Bosman, P. and Thierens, D.: An algorithmic framework for density estimation based evolutionary algorithms, *Tech. Rep. No. UU-CS-1999-46*, Utrecht University (1999).

[Bosman 00a] Bosman, P. and Thierens, D.: Continuous iterated density estimation evolutionary algorithms within the IDEA framework, *Proc. of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference GECCO-2000*, pp.197-200 (2000).

[Bosman 00b] Bosman, P. and Thierens, D.: Mixed IDEAs, *Tech. Rep. No. UU-CS-2000-45*, Utrecht University (2000).

[De Bonet 97] De Bonet, J. S., Isbell, C. L. and Viola, P.: MIMIC: Finding optima by estimating probability densities, In Mozer, M. C., Jordan, M. I., and Petsche, T. (Eds): *Advances in neural information processing systems*, Vol. 9, pp. 424-431. (1997).

[Eshelman 93] Eshelman, L. J. and Shaffer, J. D.: Real-coded genetic algorithms and interval-schemata, *Foundation of Genetic Algorithms 2*, Morgan Kaufmann, pp. 187-202 (1993).

[Goldberg 89] Goldberg, D. E.: *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley publishing company (1989).

[Harik 98] Harik, G., Lobo, F. G., and Goldberg, D. E.: The compact genetic algorithm, *Proc. of the International Conference on Evolutionary Computation 1998 (ICEC 98)*, pp. 523-528 (1998).

[Harik 99] Harik, G: Linkage learning via probabilistic modeling in the ECGA, *IlliGAL Technical Report 99010*,

University of Illinois at Urbana-Champaign, Urbana, Illinois (1999).

[Larranaga 99] Larranaga, P., Etxeberria, R., Lozano, J.A., and Pena, J.M.: Optimization by learning and simulation of bayesian and gaussian networks, *University of the Basque Country Technical Report EHU-KZAAIK -4/99* (1999).

[Mühlenbein 96] Mühlenbein, H and Paaß, G.: From recombination of genes to the estimation of distribution I. Binary parameters, *Proc. of the Parallel Problem Solving from Nature - PPSN IV*, pp. 178-187 (1996).

[Pelikan 99] Pelikan, M., Goldberg, D. E., and Cantu-Paz, E.: BOA: The Bayesian optimization algorithm, *Proc. of the Genetic and Evolutionary Computation Conference 1999 (GECCO-99)*, Morgan Kaufmann, San Francisco, CA (1999).

[Pelikan 00a] Pelikan, M., Goldberg, D. E., and Cantu-Paz, E.: Linkage problems, distribution estimate, and Bayesian network, *Evolutionary Computation*, Vol. 8, No. 3, pp. 311-340 (2000).

[Pelikan 00b] Pelikan, M., Goldberg, D. E., and Lobo, F. G. : A survey of optimization by building and using probabilistic models, *Computational Optimization and Applications*, Kluwer Academic Publishers (in press).

[Sebag 98] 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).

[Servet, I. L., Trave-Massuyes, L., and Stern, D.: Telephone network traffic overloading diagnosis and evolutionary computation techniques, *Proc. of the Third European Conference on Artificial Evolution (AE 97)*, pp. 137-144 (1997).

[Schewefel 95] Schefewel, H.-P.: *Evolution and optimum seeking*, Sixth-generation Computer Technology Series, Wiley (1995).