

An Investigation on Sampling Technique for Multi-objective Restricted Boltzmann Machine

Vui Ann Shim, Kay Chen Tan and Jun Yong Chia

Abstract—Estimation of distribution algorithms are increasingly gaining research interest due to their linkage information exploration feature. Two main mechanisms which contribute towards the success of the algorithms are probabilistic modeling and sampling method. Recent attention has been directed towards the development of probabilistic building technique. However, research on the sampling approach is less developed. Thus, this paper carries out an investigation on sampling technique for a novel multi-objective estimation of distribution algorithm – multi-objective restricted Boltzmann machine. Two variants of a new sampling technique based on energy value of the solutions in the trained network are proposed to improve the efficiency of the algorithm. Probabilistic information which is usually clamped into marginal probability distribution may hinder the algorithm in producing solutions that have high linkage dependency between variables. The proposed approach will overcome this limitation of probabilistic modeling in restricted Boltzmann machine. The empirical investigation shows that the proposed algorithm gives promising result in term of convergence and convergence rate.

I. INTRODUCTION

ESTIMATION of distribution algorithms (EDAs) [4],[10] are a new computing paradigm in the field of Evolutionary Computation (EC). EDAs are well-known for their exploitation of the explicit probability distribution of the selected subpopulation. Similar to other EC, survival of the fittest is one of the key concepts in EDAs. However, no genetic operators (crossover and mutation) are used. Instead, the genetic operators are replaced by the building of a representative probabilistic model of the previously selected individuals. The new solutions are then produced through sampling of the corresponding probabilistic model. The probabilistic modeling technique can be classified into univariate, bivariate and multivariate modeling [4]. Univariate modeling is simple and easy to implement, but does not utilize linkage information in guiding the search. This may hinder the algorithm when solving complex problems. Bivariate or multivariate modeling improves the ability of algorithms by using linkage information to explore the search space, but with increasing of complexity.

Due to the success of EDAs in single objective optimization [4], [7] the implementation of EDAs in multi-objective (MO) [24-26] framework is gaining more research interest [9], [12]. Several multi-objective estimation of

distribution algorithms (MOEDAs) have been developed. Mixture-based multi-objective iterated density-estimation evolutionary Algorithm (MIDEA) [18] is one of the first EDA being developed with both discrete and real number representations. Other MOEDAs with binary representation are Bayesian MO optimization algorithm (BMOA) [5], MO hybrid EDA (MOHEDA) [6], MO hierarchical Bayesian optimization algorithm (mohBOA) [13], MO extended compact genetic algorithm (mECGA) [15] and MO parameterless genetic algorithm (moPGA) [17]. MOEDAs with real number representation are MO Parzen-based EDA (MOPED) [1], Voronoi-based EDA (VEDA) [11], Decision tree based MOEDA (DT-MEDA) [22], regularity model based MOEDA (RMEDA) [21], and MO neural based EDA (MONEDA) [8].

Recently, Tang *et al.* [16] modeled a novel estimation of distribution algorithm based on restricted Boltzmann machine in the framework of multi-objective optimization – multi-objective restricted Boltzmann machine (MORBM). Restricted Boltzmann machine (RBM) [14], [19] is an energy-based stochastic neural network with unsupervised learning. The network is a two layered architecture with one input layer and another hidden layer. The network learns the distribution of the input data and stores this information in the network's weights and biases. The stability of the network is measured through the energy function of the network and training stops whenever the network reaches a certain degree of energy equilibrium. The learning characteristic in RBM differentiates it from other EDAs. This characteristic is hypothesized to give flexibility to the network in determining the linkage information of the input data. The capability of the network in capturing the linkage information performs well compared to EDAs with univariate probabilistic modeling.

Even though MORBM can model the multivariate dependency between variables, the final probability distribution is clamped into marginal distribution of each decision variables. Subsequently, the sampling is carried out based on the marginal distribution to produce new solutions. This feature reduces the efficiency of the algorithm in guiding the search in cases where variables have high linkage dependency. In order to fully utilize the information provided from the trained network, we proposed sampling by using the information of energy equilibrium. Two variants of sampling techniques are created based on this idea. The empirical result shows that the proposed sampling technique gives promising result in term of convergence and convergence rate.

The rest of this paper is organized as follow: Section II describes the architecture of RBM as well as its adaptation in

Vui Ann Shim, Kay Chen Tan, and Jun Yong Chia are with the Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, 117576, Singapore {e-mail: (g0800438, eletankc, g0900313)@nus.edu.sg}.

MO framework. Section III introduces the proposed sampling technique. Result and implementation are outlined in section IV and section V concludes this paper.

II. MULTI-OBJECTIVE RESTRICTED BOLTZMANN MACHINE (MORBM)

RBM [14] is an energy-based neural network with unsupervised learning. The network consists of two layers of neuron (input and hidden layers) and the layers are connected through weights and biases. The input and hidden units could be modeled using binary or continuous state. In this paper, only the binary case is considered. The indices of visible and hidden units are denoted by i and j respectively, w_{ij} is the weight of the connection between the i^{th} visible unit and the j^{th} hidden unit, b_i is the bias for visible unit and b_j is the bias for hidden unit. The network's weight is symmetric and there is no connection within the same layer. The architecture of the network is shown in Fig. 1.

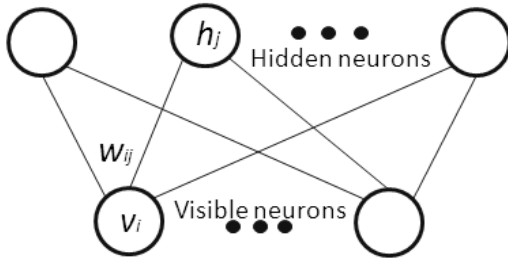


Fig. 1. Network architecture of RBM

The adaptation of RBM as EDA in MO framework (MORBM) had been carried out in [16]. In this section, the main feature of MORBM together with their mathematical formulation and process flow will be presented. Assuming that the test problems consist of n decision variables and m objective functions to be optimized, which can be formularized as

Minimize: $F(x) = (f_1(x), f_2(x), \dots, f_m(x))$

Where $x = (x_1, x_2, \dots, x_n)$

The overall framework of MORBM is presented as below:

Step (1) Initialization: At generation $t=0$, randomly generate N initial individuals $Pop(0)$, and evaluate the fitness of all solutions in $Pop(0)$.

Step (2) Fitness assignment: For all solutions in $Pop(t)$, assign fitness to each individual by Pareto ranking and Crowding distance [2].

Step (3) Selection: Based on the fitness assigned in step (2), select N individuals based on binary tournament selection to form a new subpopulation.

Step (4) Probabilistic modeling based on RBM: From the selected subpopulation, a RBM network is built and then trained until the stopping criterion is reached. Let the objective functions, consisting of n variables, be optimized. In the implementation, n variables imply n input units in RBM. Two main procedures are involved in the construction of the probabilistic model. The first procedure involves training the network, and the second involves the construction of a probabilistic model. For network training, three phases are carried out. Firstly, the positive phase will

construct the state of hidden units given the input values. Secondly, the reconstruction is carried out once to reconstruct the state of input units and the hidden units. Finally, phase three will update the weights and biases of the network. The same training procedures will be carried out until the stopping criterion is fulfilled. The construction of the probabilistic model will begin when the training terminates. The overall pseudo code for the probabilistic modeling is presented in Fig. 2.

Step (5) Sampling technique: Direct sampling from marginal distribution is implemented. N new solutions (P) are generated through sampling according to the following equation:

$$x_i = \begin{cases} 1 & \text{if } random(0,1) \leq P(v_i = 1) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where x_i is the variable i .

Step (6) Elitism: Select N solutions from $P \cup Pop(t)$ to form $Pop(t+1)$.

Step (7) Stopping Criteria: If stopping criterion is met, terminate. Else $t = t + 1$, and go to Step (2).

Begin

%%Network Training

Do While ("maximum training epoch is reach")

%%Positive Phase

1. Construct the conditional probability for hidden unit given the input value $\langle v_i \rangle_0$ according to

$$P(h_j|v) = \varphi(\sum_i w_{ij}v_i - b_j) \quad (2)$$

where $\varphi(x) = \frac{1}{1+e^{-x}}$ is the logistic function.

2. From $P(h_j|v)$, sample the state of the hidden unit $\langle h_j \rangle_0$.

%%Negative Phase

3. Based on the state of the hidden unit h_j , reconstruct the state of the input $\langle v_i \rangle_1$ by sampling according to

$$P(v_i|h) = \varphi(\sum_j w_{ij}h_j - b_i) \quad (3)$$

where $\varphi(x)$ is the logistic function.

4. Reconstruct again the state of the hidden unit $\langle h_j \rangle_1$ based on $\langle v_i \rangle_1$.

%%Weights updating

5. Update the weight and biases according to

$$w'_{ij} = w_{ij} + \epsilon(\langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_1) \quad (4)$$

$$b'_i = b_i + \epsilon(\langle v_i \rangle_0 - \langle v_i \rangle_1) \quad (5)$$

$$b'_j = b_j + \epsilon(\langle h_j \rangle_0 - \langle h_j \rangle_1) \quad (6)$$

End Do

%%Construction of probabilistic model

6. Compute energy for all solutions in subpopulation according to

$$E(v, h) = -\sum_{i=1}^n \sum_{j=1}^m v_i h_j w_{ij} - \sum_{i=1}^n v_i b_i - \sum_{j=1}^m h_j b_j \quad (7)$$

7. Compute the marginal probability according to

$$P(v_i = 1) = \frac{P(v^+) + avg(P(v))}{P(v^+) + P(v^-) + r_i * avg(P(v))} \quad (8)$$

where $P(v^+) = \sum_{h=1}^m e^{-E(v=1, h)}$ is the marginal cost of v_i when the cardinality of $v_i = 1$, and $P(v^-) = \sum_{h=1}^m e^{-E(v=0, h)}$ is the marginal cost of v_i when the cardinality of $v_i = 0$, and $avg(P(v))$ is the average cost of cardinality

End

Fig. 2. Pseudo code for MORBM

III. PROPOSED SAMPLING TECHNIQUE

One of the main characteristics of MORBM is its capability to learn the multivariate dependencies between variables. This information is stored in the synaptic weights and biases of the network. The final probability distribution is constructed by clamping this information into the marginal probability for each decision variable or input unit in the network. The offspring for next generation are then sampled from the built probabilistic model.

The direct sampling applied in MORBM may, however, limit the production of appropriate solutions if the decision variables are highly correlated or in high dimension. This is because marginal probability distribution considers the distribution of the particular decision variable but not the correlation between variables. One way to tackle this problem is to sample infinite number of solutions. This may increase the possible combination of the solutions and thus increase the chance of producing fitter individuals. However, sampling of an infinitely large number (M times) of solutions may lead to an increase in population size and the fitness evaluations. It is known that some real world problems are very time consuming; thus, the impracticality of such an algorithm.

To deal with this problem, energy is taken into consideration. Energy value will serve as a main criterion to choose N solutions out of $N \times M$ solutions, where M is a multiplier with $M > 1$. The energy value is used as a lower energy implies that the solutions are in stable state while a higher energy means that the solutions are not in energy equilibrium. The proposed sampling algorithm will, therefore, prefer solutions with lower energies. As probabilistic modeling only models the previous best topology, the solutions which are located inside the topology are stable (lower energy) in term of energy equilibrium. On the other hand, the solutions out of this topology (higher energy) may consider unstable but not unfit. Therefore, it is better to give the algorithm the flexibility to choose solutions with high energy in order to increase the exploration in searching.

Basing on this argument, the following sampling technique is proposed (Fig. 3). First of all, $N \times M$ solutions will be sampled according to Eq. 1 and stored in z . z will become input data for RBM. Based on z , the corresponding state for hidden unit h_j will be constructed according to Eq. 2. By using h_j , z , and the previously trained weights and biases, the energy for each of the solution in z is computed

Begin

Given marginal probability of each variable, $P(v_i)$

1. Sample $N \times M$ solutions according to Eq. 1 and store in z .
2. Reconstruct the state for hidden units based on all solutions produced in (1), according to Eq. 2.
3. Compute the energy value for all solutions according to Eq. 7.
4. Sort the energy in increasing order.
5. Apply selection procedure to select N individuals to form final offspring.

End

Fig. 3. Pseudo code for proposed sampling technique

according to Eq. 7. After which, the energies together with their indices are sorted in an increasing order. Finally, selection procedure will be applied to select N individuals out of $N \times M$ solutions. Selection procedure is crucial in determine the fitter solutions.

Two selection procedures are proposed in this paper in order to view the different performance under different selection scheme.

A. Uniform selection scheme

Under this scheme, the indices of the solutions are uniformly selected. The process is shown in the following pseudo code, where $\text{Pop}(t)_{i,j}$ is the population i with bit position j at generation t .

```

For i=1:N
  For j=1:Variable Number x Variable Bit
    RandA = Generate uniform random value between[0, NxM]
    Pop(t)i,j = z (RandA)
  End For
End For

```

Fig. 4. Pseudo code for uniform selection scheme

B. Inverse exponential selection scheme

Under this scheme, higher probability is given to individuals with lower energy while a lower probability is given to individuals with higher energy. The pseudo code for this scheme is presented in Fig. 5.

```

For i=1:N
  For j=1:Variable Number x Variable Bit
    RandB = Generate uniform random value between [ $\alpha * \min$ ,  $\alpha * \max$ ]
    RandC = (Exp (RandB) - Exp ( $\alpha * \min$ )) *  $\frac{N \times M}{\text{Exp}(\alpha * \max) - \text{Exp}(\alpha * \min)}$ 
    Pop(t)i,j = z (RandC)
  End For
End For

```

Fig. 5. Pseudo code for inverse exponential selection scheme

\min and \max are the value that determine the range of the random values. In order to give the flexibility to change the range of the random value, α is added to the algorithm. For ease and simplicity, \min and \max are permanently assigned a value of 0.01 and 1.0, respectively. In this way, α is the only parameter that determines the probability selection for the indices. The probability curves are shown in Fig. 6. In the figure, it is observed that smaller α will give quite an even probability to all indices. The probability of choosing indices with smaller energy will increase and the probability of selecting solutions with higher energy will decrease with the increment of α .

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section outlines the implementation and parameter settings of the simulations. The results of the simulations would also be presented. Further investigations would be carried out in the last subsection to examine the effect of alpha (α) and multiplier (M) in the evolution process.

A. Implementation

Eight benchmark test instances (ZDT [23] and DTLZ [3]) are chosen to test the performance of the proposed algorithm. The test problems consist of different characteristics and difficulties. ZDT problems have two objective functions while DTLZ have three objective functions to be optimized simultaneously. The definition and characteristics of the test problems are summarized in Table 1.

In order to show the effectiveness of the proposed algorithm, three other algorithms (MORBM, NSGAI, and MOUMDA) are taken into comparison with the proposed algorithms. MORBM [16] is essential in this comparison since the proposed algorithm is designed based on MORBM. NSGAI [2] is chosen since it is a favorite benchmark MOEA and this algorithm generally could achieve good results for most of the test problems. MOUMDA is another MOEDA based on univariate marginal distribution algorithm (UMDA) [10]. The basic architecture of MOUMDA is quite similar with MORBM. The main difference is that MORBM utilizes multivariate modeling while MOUMDA uses univariate modeling.

The simulation in this study is run on an Intel Core2 Duo CPU, 1.66GHz notebook personal computer and all the simulation coding are implemented in C++. The experimental settings are as follow:

- Population size: 100 for all test problems
- Stopping criteria: 100 generation for all test problems
- Independent runs: 10 runs
- Representation: Binary representation for all algorithms
- Bits per variable: 10 bits
- Hidden unit in RBM: 20 for ZDT problems while 5 for DTLZ problems as suggested in original paper [16]
- Training epoch in RBM: 10 for all test instances
- Learning rate in RBM: 0.1
- α in inverse exponential selection scheme: 5.5
- Multiplier M in the proposed sampling technique :10
- Crossover rate in NSGAI: 0.8
- Mutation rate in NSGAI: $1/(\text{Variable size} * \text{Variable bit})$

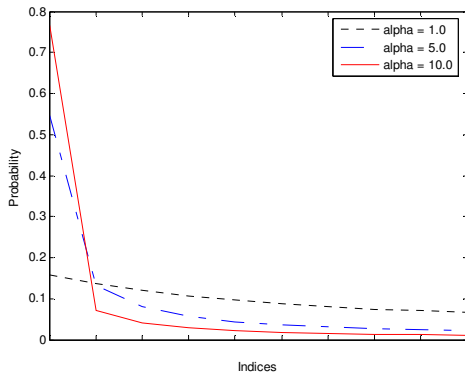


Fig. 6. Probability selection for different value of α

TABLE I
TEST PROBLEMS

Problems	Objective Functions	Characteristics
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - \sqrt{f_1(x)/g(x)})$ $g(x) = 1 + 9(\sum_{i=2}^n x_i)/(n-1)$	Convex Pareto front, $n=100$ $x_i \in [0,1]$
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - (f_1(x)/g(x))^2)$ $g(x) = 1 + 9(\sum_{i=2}^n x_i)/(n-1)$	Non-convex Pareto front, $n=100$, $x_i \in [0,1]$
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - \sqrt{f_1(x)/g(x)} - f_1(x)/g(x) \sin(10\pi f_1(x)))$ $g(x) = 1 + 9(\sum_{i=2}^n x_i)/(n-1)$	Non-continuous convex Pareto front, $n=100$, $x_i \in [0,1]$
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x)(1 - \sqrt{f_1(x)/g(x)})$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$	Many local optimal convex Pareto front, $n=10$, $x_i \in [0,1]$
ZDT6	$f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(x) = g(x)(1 - (f_1(x)/g(x))^2)^{0.25}$ $g(x) = 1 + 9[\sum_{i=2}^n x_i/(n-1)]^{0.25}$	Non-uniform concave Pareto front $n=100$, $x_i \in [0,1]$
DTLZ1	$f_1(x) = (1/2) x_1 x_2 \dots x_{M-1} (1 + g(x_M))$ $f_2(x) = (1/2) x_1 x_2 \dots x_{M-1} (1 - x_{M-1}) (1 + g(x_M))$ \vdots $f_{M-1}(x) = (1/2) x_1 (1 - x_2) (1 + g(x_M))$ $f_M(x) = (1/2) (1 - x_1) (1 + g(x_M))$, Where $g(x) = 100 * (X_M + \sum_{x_i \in X_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)))$	Linear Pareto Optimal front $n=12, M=3$, $x_i \in [0,1]$
DTLZ2	$f_1(x) = (1 + g(x)) \cos(x_1 \pi/2) \dots \cos(x_{M-1} \pi/2)$ $f_2(x) = (1 + g(x)) \cos(x_1 \pi/2) \dots \sin(x_{M-1} \pi/2)$ \vdots $f_M(x) = (1 + g(x)) \sin(x_1 \pi/2)$, where $g(x) = \sum_{x_i \in X_M} (x_i - 0.5)^2$	Spherical Pareto Optimal front $n=12, M=3$, $x_i \in [0,1]$
DTLZ3	$f_1(x) = (1 + g(x)) \cos(x_1 \pi/2) \dots \cos(x_{M-1} \pi/2)$ $f_2(x) = (1 + g(x)) \cos(x_1 \pi/2) \dots \sin(x_{M-1} \pi/2)$ \vdots $f_M(x) = (1 + g(x)) \sin(x_1 \pi/2)$, Where $g(x) = 100 * (X_M + \sum_{x_i \in X_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)))$	Many local Pareto Optimal Front, $n=12, M=3$, $x_i \in [0,1]$

B. Simulation Result

The empirical result is presented using the Inverted Generational Distance (IGD) indicator. IGD [20], [21] is a performance metric which measures the proximity as well as diversity between the evolved Pareto front and optimal Pareto front. A smaller value for IGD implies better performance in term of proximity to the optimal Pareto front and a wider distribution along optimal Pareto front. Box plot is used to show the performance of various algorithms under 10 simulation runs, while line curve is used to present the convergence trace. The legend for the convergence trace curve is shown in Fig. 7. The algorithm's index for box plot

is presented in Table 2. MORBM/E refers to MORBM with inverse exponential selection scheme while MORBM/U is MORBM with uniform selection scheme.

Fig. 8 shows the (a) convergence trace and (b) IGD result of various algorithms at 10,000 fitness evaluations for ZDT1. From the figure, it is observed that MORBM/E outperforms other algorithm in both convergence and convergence rate. No improvement is visualized for MORBM/U. This may be due to the fact that most of the fitter solutions have smaller energy value since smaller energy means the solutions are stable under the modeled probability distribution. The implementation of inverse exponential selection scheme means that a solution with smaller energy value stands a higher chance of being chosen, thus giving a better performance. For MORBM/U, equal probability selection is given to all individuals. Therefore, the selected offspring consist of an equal number of lower and higher energy values, which is quite similar to the condition of original MORBM. MOUMDA and NSGAIi perform worse than all versions of MORBM. NSGAIi incorporates stochastic recombination which requires larger fitness evaluations to evolve for better performance and MOUMDA models the probability distribution without consider any linkage information.

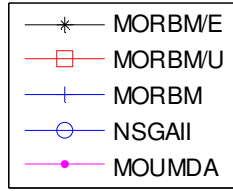


Fig. 7. Legend for convergence trace's curve

TABLE II
ALGORITHM'S INDEX

Index	Algorithm
1	MORBME
2	MORBM/U
3	MORBM
4	NSGAIi
5	MOUMDA

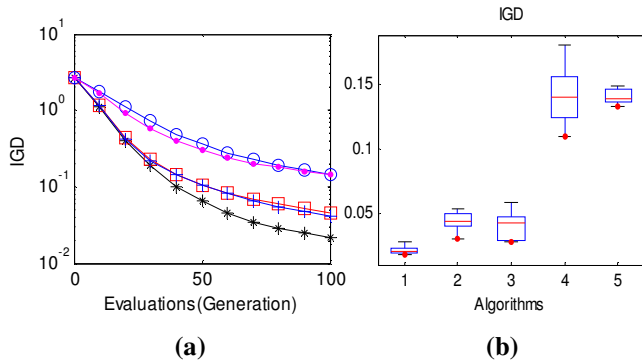


Fig. 8. (a) Convergence trace and (b) IGD for ZDT1

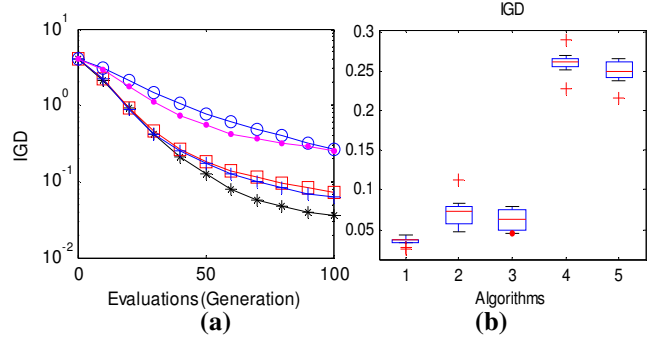


Fig. 9. (a) Convergence trace and (b) IGD for ZDT2

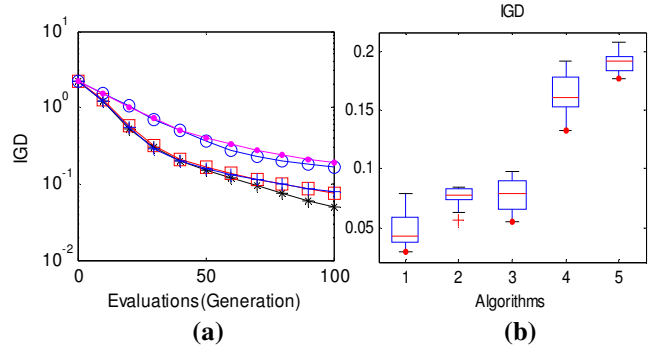


Fig. 10. (a) Convergence trace and (b) IGD for ZDT3

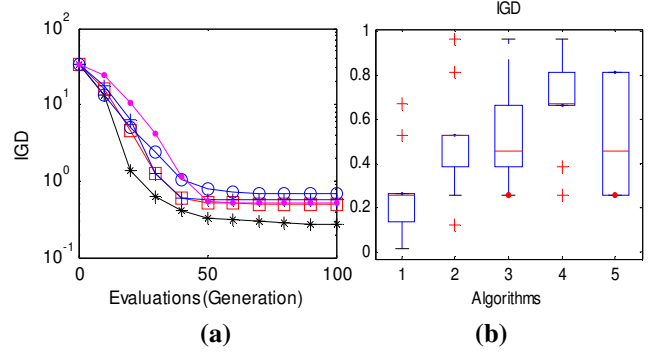


Fig. 11. (a) Convergence trace and (b) IGD for ZDT4

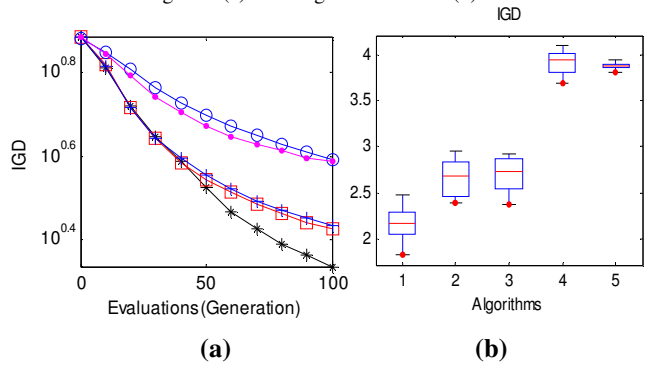


Fig. 12. (a) Convergence trace and (b) IGD for ZDT6

ZDT2 and ZDT3 are characterized with non-convex and non-continuous Pareto optimal front. The performance trace and IGD at 10,000 fitness evaluations are similar to the one observed in ZDT1. The results are illustrated in Figs. 9 and 10. ZDT4 is characterized by many local optimal fronts. Due to the difficulty of this problem, the decision variable is set to 10 as proposed in [19].

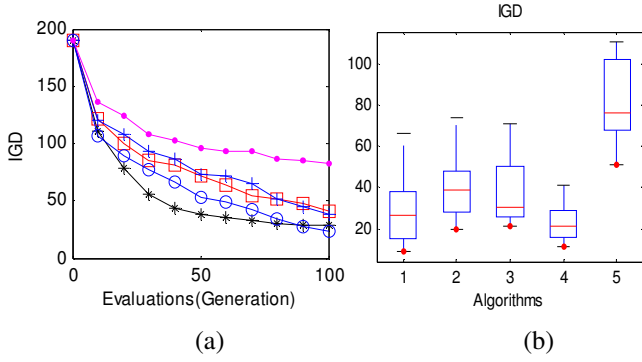


Fig. 13. (a) Convergence trace and (b) IGD for DTLZ1

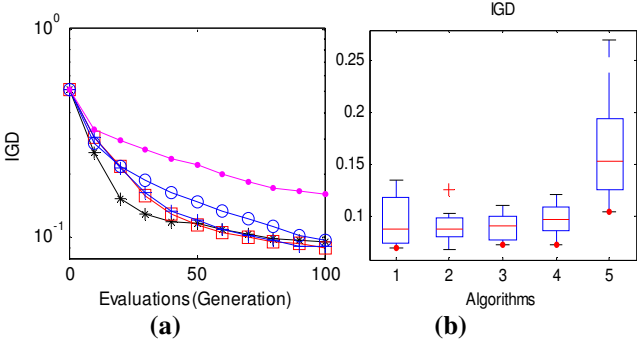


Fig. 14. (a) Convergence trace and (b) IGD for DTLZ2

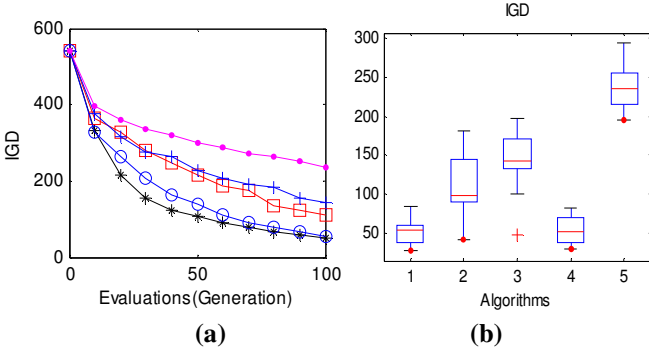


Fig. 15. (a) Convergence trace and (b) IGD for DTLZ3

Fig. 11 shows the (a) convergence trace and (b) IGD value at 10,000 fitness evaluations for ZDT4. It can be observed that all algorithms are unable to produce a global optimal Pareto front. The performance of MORBM improved with the implementation of inverse exponential selection scheme. ZDT6 is another challenging problem which is characterized by the non-uniformity non-convex Pareto optimal front. Fig. 12 shows the result for this problem. It is observed that MORBM/E gave the best performance. The improved performance is due to the incorporations of the proposed sampling technique which may samples more fitter solutions and these solutions are then being selected by inverse exponential selection scheme.

DTLZ problems are well-known being able to scale-up to any number of decision variables as well as objective functions. Since we are not dealing with scalable problem, all the test instances are fixed to 3 objective functions and 12 decision variables. DTLZ1 is well-known to have many local linear Pareto optimal fronts. Fig. 13 shows the (a)

convergence trace and b) IGD of various algorithms at 100 generations for DTLZ1. It is clear that NSGAII performs better than MORBM, MORBM/U and MOUMDA. The incorporation of inverse exponential selection scheme (MORBM/E) improves the performance of original MORBM and it outperforms NSGAII in terms of convergence rate. It produces a final convergence which is comparable to that of NSGAII's. A similar result is observed for DTLZ2 and DTLZ3 as presented in Figs. 14 and 15.

From this observation, we can conclude that sampling technique with inverse exponential selection scheme succeeded in improving the performance of MORBM while uniform selection scheme does not result in any significant improvement. This observation suggests that greater probability should be assigned to solutions with smaller energy, while smaller probability should be assigned to individuals with higher energy.

C. Further Analyses

There are two parameters (α and M) which can affect the performance of the proposed algorithm. In this section, further investigations are carried out to investigate the influence of these parameters in optimization. It is certain that the incorporation of selection scheme in MORBM will increase the computational time. Therefore, we also carried out a computational time analysis to show how costly will the proposed feature be.

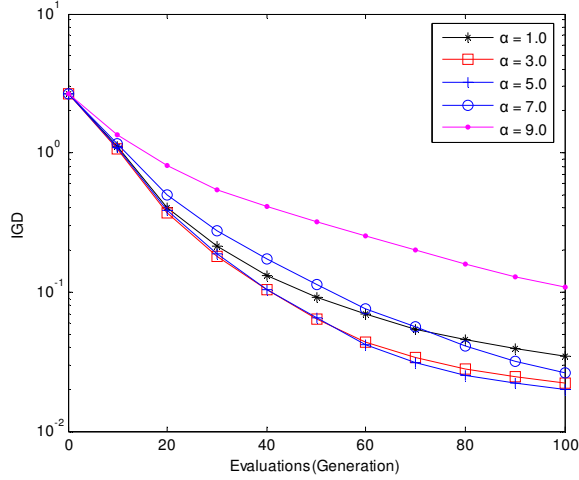
1) Effect of α , α

The function of α is to determine the factor for exponential ($e^{\alpha x}$). Let x be a fix value and α be the only parameter that determines the probability of selection as shown in Fig. 6. In this section, the effect of different selection probability in optimization is examined.

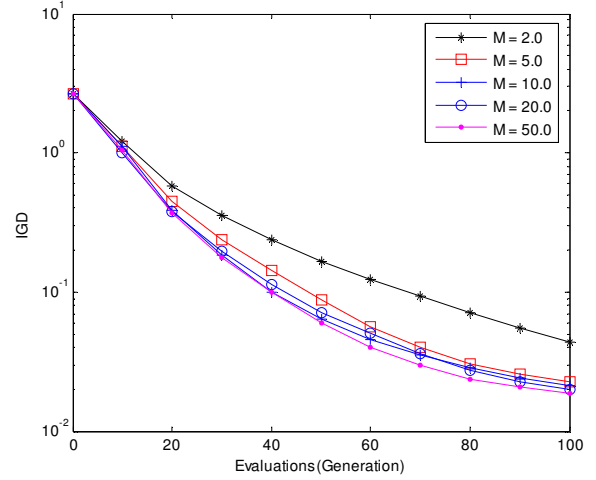
Fig. 16 shows the convergence trace for MORBM/E for a) ZDT1 and b) DTLZ1 under different values of α . It is observed that smaller value (1.0) and higher value (9.0) for α would not give good result over the generations, while α ranging from 3.0 to 7.0 gives acceptable performances. Smaller value of α will give a relatively even probability to selection of each solution, while higher value of α will give very large probability to selecting solutions with lower energy value. A set of offspring will have more unfit individuals if an equal probability is used to select the solutions. On the other hand, if too large probability selection is used to select individuals with smaller energy, less exploration of the search space is performed by the algorithm. The selected individuals will be located in the same topology as those modeled by probabilistic model. A set of offspring with too many solutions of lower energies or higher energies is not an ideal case in evolutionary process since exploration and exploitation must be balanced.

TABLE III.
COMPUTATIONAL TIME (S) USED BY MORBM/E IN ZDT1 AND DTLZ1 WITH DIFFERENT VALUE OF M

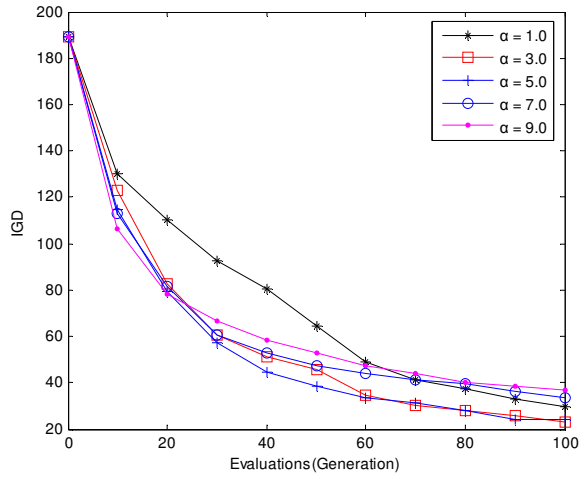
Problems	Multiplier, M					
	1	2	5	10	20	50
ZDT1	2.3848	2.6280	2.9019	3.3660	4.2039	6.9802
DTLZ1	0.1113	0.1334	0.1473	0.1733	0.2234	0.3742



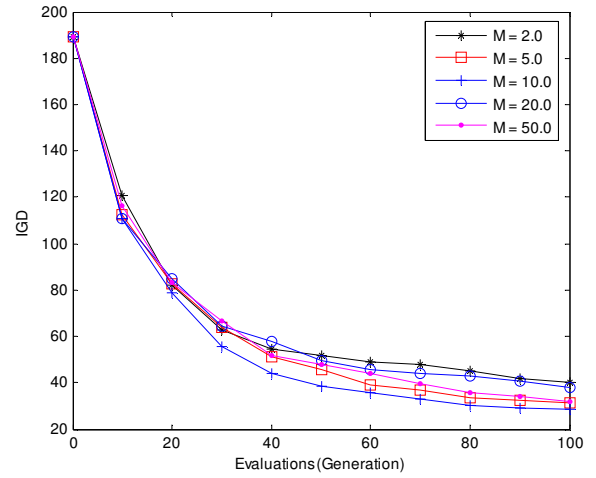
(a)



(a)



(b)



(b)

Fig. 16. Convergence trace for MORBM/E in a) ZDT1 and b) DTLZ1 with different value of α .

Fig. 17. Convergence trace for MORBM/E in a) ZDT1 and b) DTLZ1 with different value of M .

2) Effect of Multiplier, M

The fundamental idea of the proposed sampling technique is that sampling the population to an infinitely large number of solutions may increase the chance of producing fitter candidate solutions. However, such samplings are often impractical. Thus, a multiplier M is used to determine the number of the sampled individuals. It may be argue that M can affect the final performance of the algorithm. Thus, this section carries out an investigation to examine the effect of M and then suggests a possible range of M .

Fig. 17 shows the convergence trace for MORBM/E in a) ZDT1 and b) DTLZ1 with different value of M . It is observed that greater M (20 to 50) gave better performance in ZDT1 but not in DTLZ1. A smaller M (2) led to poorer performances while M around 5 to 10 gave generally good result in both the test instances. Therefore, we conclude that large value of M is unnecessary, and a M of around 5 to 10 is enough to evolve a good Pareto front.

3) Computational time Analysis

From section 3, it is clear that the proposed sampling algorithm will cause extra computational time due to the

sampling $N*M$ candidate solutions rather than N solutions, and re-computation of hidden state. A computational time analysis is carried out to determine how costly the proposed sampling technique will be. Table 3 presents the computational time (one generation) used by MORBM/E in ZDT1 and DTLZ1 with different value of M . From the table, the computational time is increased approximately 3 times when the multiplier is 50. However, only a slight increment in computational time is obtained for smaller value of M (2-10). Since M around 5-10 is able to give generally good results, the extra computational time can be considered as inexpensive.

V. CONCLUSIONS

In this study, a new sampling technique for MORBM is proposed. The sampling approach took advantage of the energy information of the solutions. The energy is used to determine a set of offspring to undergo the evolution process in next generation. The set of selected offspring is a combination of solutions with lower and higher energy. Two variants of selection schemes are presented. Uniform

selection scheme will give an equal probability to each solution without considering their energy value. On the other hand, inverse exponential selection scheme will assign a larger probability selection to solutions with lower energy and a smaller probability selection to individuals with higher energy value. The experimental result shows that the proposed algorithm with inverse exponential selection scheme could improve the performance of MORBM in term of convergence and convergence rate; sacrificing a little computational time. Further analyses are carried out to investigate the parameter setting in the proposed algorithm. Even through the modification in this paper has improved the performance of the previous version of MORBM, the experimental result shows that all algorithms are unable to converge to global Pareto optimal front for problem with many local optimal. Therefore, further investigation is required to overcome the limitation of this algorithm.

ACKNOWLEDGEMENT

This research was supported by Academic Research Fund (AcRF) under Project No.: R-263-000-480-112.

REFERENCES

- [1] M. Costa and E. Minisci, "MOPED: A Multi-objective Parzen-based Estimation of Distribution Algorithm for Continuous Problems," *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization*, page 282-294, 2003.
- [2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGAII," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
- [3] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimization Test Problems," *Proceedings of the 2002 Congress on Evolutionary Computation*, Vol. 1, pp. 825-830, 2002.
- [4] P. Larrañaga and J. A. Lozano, "Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation," Norwell, MA: Kluwer, 2001.
- [5] M. Laumanns and J. Ocenasek, "Bayesian Optimization Algorithms for Multi-Objective Optimization," *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, pp. 298-307, 2002.
- [6] H. Li, Q. Zhang, E. Tsang, and J. A. Ford, "Hybrid Estimation of Distribution Algorithm for Multi-objective Knapsack Problem," *Evolutionary Computation in Combinatorial Optimization*, pp. 145-154, 2004.
- [7] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, "Towards an New Evolutionary Computation: Advances on Estimation of Distribution Algorithms" Springer-Verlag, 2006
- [8] M. Luis, G. Jesús, B. Antonio, and M. M. José, "Solving Complex High-Dimensional Problems with the Multi-objective Neural Estimation of Distribution Algorithm," *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 619-626, 2009.
- [9] L. Marti, J. García, A. Berlanga, and J. M. Molina, "On the Model-Building Issue of Multi-Objective Estimation of Distribution Algorithms," *Hybrid Artificial Intelligence System*, pp. 293-300, Springer, Heidelberg, 2009.
- [10] H. Muhlenbein and G. Paaß, "From recombination of genes to the estimation of distribution algorithm I. Binary parameters," *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pp. 178-187, 1996.
- [11] T. Okabe, Y. Jin, B. Sendhoff, and M. Olhofer, "Voronoi-based estimation of distribution algorithm for multi-objective optimization," *Congress of Evolutionary Computation*, pp. 1594-1601, 2004.
- [12] Pelikan, M., Sastry, K., Goldberg, D.E.: *Multiobjective estimation of distribution algorithms*. In: Pelikan, M., Sastry, K., Cant'u-Paz, E. (eds.) *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*. Studies in Computational Intelligence, pp. 223–248. Springer, Heidelberg, 2006
- [13] Pelikan, M., Sastry, K., and Goldberg, D. Multiobjective hBOA, clustering, and scalability. *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp. 663-670, 2005.
- [14] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering", *Proceedings of the 24th international conference on Machine learning*, pp. 791–798, 2007.
- [15] K. Sastry, D. Goldberg, and M. Pelikan, "Limits of scalability of multiobjective estimation of distribution algorithms," *IEEE Congress on Evolutionary Computation*, 2005, vol. 3, pp. 2217–2224, 2005.
- [16] H. J. Tang, V. A. Shim, K. C. Tan and J. Y. Chia, "A novel multi-objective estimation of distribution algorithm based on restricted Boltzmann machine," *IEEE Congress on Evolutionary Computation*, 2010.
- [17] H. Soh and M. Kirley, "moPGA: Towards a New Generation of Multi-objective Genetic Algorithms," *IEEE Congress on Evolutionary Computation*, pp. 1702-1709, 2006.
- [18] D. Thierens and P. A. N. Bosman, "Multi-objective mixture-based iterated density estimation evolutionary algorithms," *Proceedings of the GECCO-2001 Genetic and Evolutionary Computation Conference*, pp. 663–670, 2001.
- [19] Tieleman, T. Training restricted Boltzmann machines using approximations to the likelihood Gradient. *Proceedings of the 25th Annual International Conference on Machine Learning*, pp.1064–1071, 2008.
- [20] A. Villalobos, M. A. Pulido, and C. A. C. Coello, "A Proposal to Use Stripes to Maintain Diversity in A Multi-objective Particle Swarm Optimizer," *In proceeding of the 2005 IEEE Swarm Intelligence Symposium*, pp. 22-29, 2005.
- [21] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm," *IEEE Transaction on Evolutionary Computation*, vol. 12, pp. 41-63, 2008.
- [22] X. Zhong and W. Li, "A decision-tree-based multiobjective estimation of distribution algorithm," *International Conference on Computational Intelligence and Security*, pp. 114-118, 2007.
- [23] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich, 1999.
- [24] C. A. Coello Coello, "Evolutionary multi-objective optimization: a historical view of the field," *Computational Intelligence Magazine, IEEE*, vol.1, no.1, pp. 28-36, Feb. 2006
- [25] C. A. Coello Coello, "The EMOO repository: a resource for doing research in evolutionary multiobjective optimization," *Computational Intelligence Magazine, IEEE*, vol.1, no.1, pp. 37-45, Feb. 2006
- [26] Y. C. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *Computational Intelligence Magazine, IEEE*, vol.4, no.3, pp.62-76, Aug. 2009