

# An Energy-Based Sampling Technique for Multi-Objective Restricted Boltzmann Machine

Vui Ann Shim, Kay Chen Tan, and Chun Yew Cheong

**Abstract**—Estimation of distribution algorithms are gaining increased research interest due to their advantage in exploiting linkage information. This paper examines the sampling techniques of a restricted Boltzmann machine-based multi-objective (MO) estimation of distribution algorithm (REDA). The behaviors of the sampling techniques in terms of energy levels are rigorously investigated, and a sampling mechanism that exploits the energy information of the solutions in a trained network is proposed to improve the search capability of the algorithm. The REDA is then hybridized, with a genetic algorithm and a local search based on an evolutionary gradient approach, to enhance the exploration and exploitation capabilities of the algorithm. Thirty-one benchmark test problems, which consist of different difficulties and characteristics, are used to examine the efficiency of the proposed algorithm. Empirical studies show that the proposed algorithm gives promising results in terms of inverted generational distance and nondominance ratio in most of the test problems.

**Index Terms**—Estimation of distribution algorithms (EDAs), evolutionary gradient search, genetic algorithm (GA), multi-objective (MO) optimization, restricted Boltzmann machine, sampling technique.

## I. INTRODUCTION

ESTIMATION of distribution algorithms (EDAs) [1]–[5], also known as iterated density estimation algorithms (IDEAs) [6] and probabilistic model building genetic algorithms (PMBGAs) [7], are a new computing paradigm in the field of evolutionary computation (EC). EDAs are well known for their ability to exploit explicit probability distributions of the selected subpopulation. Similar to other EC techniques [8], [9], survival of the fittest is one of the key concepts in EDAs. However, genetic operators (crossover and mutation that are widely used in genetic algorithms) are not used in EDAs. Instead, they are replaced by a representative probabilistic model of the previously selected individuals. New solutions are produced through sampling the corresponding probabilistic model. The probabilistic modeling technique can be classified

into univariate, bivariate, and multivariate [1]. 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 uses linkage information in the decision space to improve the search ability of the algorithms, but is generally more complex.

Due to the success of EDAs in single-objective optimization [10]–[14], the implementation of EDAs for multi-objective (MO) optimization [15] is gaining research interest. Several MO estimations of distribution algorithms (MOEDAs) have been developed. Recently, Tang *et al.* [16] modeled a novel EDA based on a restricted Boltzmann machine for MO optimization (REDA). A restricted Boltzmann machine (RBM) [17]–[21] is an energy-based stochastic neural network with unsupervised learning. The network has a two-layer architecture comprising an input layer and a hidden layer. The network learns the distribution of the input data and stores this information in the network weights and biases. The stability of the network is measured through the energy function of the network and training stops when the network reaches a certain degree of energy equilibrium. The learning characteristic of an RBM differentiates it from the other EDAs. This characteristic gives flexibility to the network in determining the probability distribution and the linkage information of the input data. While the REDA has been shown to be able to solve high-dimensional problems with a large number of decision variables and objective functions [16], its performance, and those of MOEDAs in general, is particularly dependent on the probabilistic modeling and sampling techniques used. Over the past few years, many modeling approaches, including Bayesian tree, decision tree, and principle component analysis (PCA), have been studied. On the other hand, sampling techniques for MOEDAs are less studied and developed.

This paper studies the sampling procedure of the REDA. Even though the REDA considers multivariate dependencies between the decision variables in its network training, the final probability distribution is constructed in the form of marginal distribution of the decision variables. Subsequently, sampling is simply carried out based on the marginal distribution to produce new solutions. This feature may reduce the efficiency of the algorithm in exploring the space of potential solutions in cases where the number of decision variables is high, and when the decision variables have strong linkage dependencies. In order to fully utilize the information provided by the trained network (energy value), the characteristics of

Manuscript received August 24, 2011; revised July 26, 2012; accepted December 19, 2012. Date of publication January 22, 2013; date of current version November 26, 2013. This work was supported by the Singapore Ministry of Education Academic Research Fund Tier 1.

V. A. Shim is with the Institute for Infocomm Research, A\*STAR, 138632, Singapore (e-mail: shimva@i2r.a-star.edu.sg).

K. C. Tan is with the Department of Electrical and Computer Engineering, National University of Singapore, 117576, Singapore (e-mail: elctankc@nus.edu.sg).

C. Y. Cheong is with the Institute of High Performance Computing, A\*STAR, 138632, Singapore (e-mail: cheongcy@ihpc.a-star.edu.sg).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEVC.2013.2241768

the sampled solutions, in terms of the energy equilibrium, are examined. This is important since the energy value is derived from the dependency information of the input data captured by the network. This investigation leads us to propose a new sampling mechanism that makes use of the energy value information. EDAs are particularly weak in generating a diverse set of solutions [16], [22]. Thus, the REDA with energy-based sampling mechanism (REDA/E) is hybridized with a genetic algorithm (GA) and a local search based on an evolutionary gradient search approach (HREDA-E). The efficiency of the HREDA-E is rigorously examined under 31 benchmark test instances. Performance indicators, such as inverted generational distance (IGD) [23] and non-dominance ratio (NR) [24] are used to evaluate the performance of the proposed algorithm. The empirical studies show that HREDA-E gives promising results in terms of IGD and NR in most of the test problems.

The rest of this paper is organized as follows. Section II presents the literature review. Section III describes the architecture of an RBM and its adaptation to the MO framework. Section IV presents an investigation on the probabilistic modeling and sampling techniques. Section V introduces the proposed algorithm. Experimental setups and test functions are illustrated in Section VI. Results and further analyses are outlined in Section VII and Section VIII concludes this paper.

## II. LITERATURE REVIEW

Mixture-based MO iterated density-estimation evolutionary algorithm [25] is one of the first EDAs developed in MO framework with both discrete and real-number representations. Several strategies were employed to construct the probabilistic model, including a mixture of discrete univariate factorization, a mixture of tree factorization, a mixture of continuous univariate factorization, and a mixture of conditional Gaussian factorization. The mixture-based approach is an elegant approach to enhancing the population diversity.

Other MOEDAs with binary-number representation include Bayesian MO optimization algorithm (BMOA) [26], MO hybrid EDA [27], MO hierarchical Bayesian optimization algorithm (mohBOA) [28], MO extended compact genetic algorithm (meCGA) [29], and MO parameter-less genetic algorithm (moPGA) [30]. In BMOA [26], the Bayesian optimization algorithm with binary decision trees was used to capture the mutual dependencies between the decision variables. The probability information is constructed in the built binary decision tree. This algorithm is effective in solving simple MO 0/1 knapsack problems. Additional computational resource is required for harder test instances.

Another approach that utilized Bayesian optimization algorithm as its probabilistic modeling approach is the mohBOA [28]. This algorithm employed the NSGA-II's non-dominance sorting framework by replacing the genetic operators with a Bayesian-based modeling approach and a sampling operator.  $k$ -mean clustering was also adapted to divide the objective space into different regions for modeling purposes. The mohBOA algorithm showed promising results in generating the Pareto solutions for discrete scalable deceptive problems.

Sastry *et al.* [29] studied the characteristics and optimization performance of MO extended compact GA (meCGA) on a class of bounding adversarial problems with scalable decision variables. Soh and Kirley [30] suggested an moPGA to solve scalable problems. The algorithm was constructed by incorporating the extended compact GA as its modeling approach, competent mutation as its enhanced searching operator, clustering as a diversity enhancement approach, and an external  $\epsilon$ -Pareto archive to preserve the promising solutions found. Binary deceptive problems with scalable decision variables and DTLZ problems with scalable objective functions were used to test the effectiveness of the algorithm.

MOEDAs with real-number representation include MO Parzen-based EDA (MOPED) [31], Voronoi-based EDA [32], decision tree-based MOEDA (DT-MEDA) [33], regularity model-based MOEDA (RM-MEDA) [22], and MO neural-based EDA [34]. In [31], the Parzen estimator, a non-parametric technique, was used to estimate the population density of the promising solutions. In order to improve the population diversity, a spreading technique that utilized Parzen estimator in the objective space was also proposed. The results indicated that MOPED takes less number of fitness evaluations to reach a satisfying performance.

In [33], an approach that uses a decision tree as its probabilistic modeling technique (DT-MEDA) was proposed. The correlations between the decision variables are captured in terms of conditional probability that is derived from the structure of the tree. Offspring solutions are then produced by sampling the tree from the root node to the left node. The results showed that DT-MEDA outperforms or is comparable to NSGA-II in some simple continuous global optimization test problems. Martí *et al.* [34] designed an algorithm that uses growing neural gas network as its probabilistic modeling technique. The growing neural gas network is a self-organizing neural network based on a neural gas model. This model creates an ordered cluster of input data set; a new cluster will then be inserted based on the topology and cumulative errors.

In [22], a regularity model-based MO estimation of distribution algorithm (RM-MEDA) was designed. This algorithm is based on the idea that there exist regularity patterns of the Pareto front over the evolutionary processes. By taking these regularity patterns into consideration, the Pareto optimal front of an MO optimization problem could be predicted. A local PCA was used to capture the regularity patterns of the solution set. The algorithm showed promising scalability performance in terms of decision variables. Zhou *et al.* [35] generalized the idea employed in RM-MEDA and proposed an improved version of RM-MEDA, named probabilistic model-based MO evolutionary algorithm or MMEA, to approximate the set of Pareto optimal solutions in both the decision and objective spaces.

Application of MOEDAs in real-world problems is a promising field to be explored in the near future. In [36], MO univariate marginal distribution algorithm (MOUMDA) was used to optimize the mixed analogue-digital signal circuits. In [37], a Gaussian model-based EDA was employed to optimize the design of a radio frequency identification design. The optimization performance is then enhanced by hybridizing

the proposed algorithm with a particle swarm optimization algorithm. The hybrid algorithm succeeded in generating a set of tradeoff solutions.

### III. RESTRICTED BOLTZMANN MACHINE BASED ESTIMATION OF DISTRIBUTION ALGORITHM

REDA [16] is an EDA that uses RBM for probabilistic model building and applies other canonical operators from state-of-the-art MOEAs [38]–[42] for MO optimization. This section will first provide an introduction of the RBM and then describe the working framework of the REDA.

#### A. Restricted Boltzmann Machine

RBM [17]–[21], [43] is an energy-based stochastic binary neural network. The training of the network is based on unsupervised learning. The network consists of two layers of neurons (input and hidden layers) and the layers are connected through weights and biases. The RBM is similar to the classical Boltzmann machine [44], [45], except that no connection is allowed between the neurons within the same layer. Visible units serve as data vectors and hidden units act as latent variables. Since connections are prohibited between the neurons in the hidden layer, the hidden units are conditionally independent. Furthermore, the visible units can be updated in parallel given the hidden states. The input and hidden units can be modeled in both binary and Gaussian states. The indices of the 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 the  $i$ th visible unit, and  $d_j$  is the bias for the  $j$ th hidden unit. The network's weights are symmetrical. The architecture of the network is shown in Fig. 1. The weights and biases of an RBM define the energy function of the network. The energy function is given as follows:

$$E(v, h) = - \sum_i \sum_j v_i h_j w_{ij} - \sum_i v_i b_i - \sum_j h_j d_j. \quad (1)$$

When the RBM has converged to the equilibrium state at the end of the training process, the probability distribution of any global state can be determined according to the following equation:

$$p(v, h) = \frac{\exp(-E(v, h))}{\sum_{x, y} \exp(-E(x, y))}. \quad (2)$$

The numerator is defined by the energy of that particular state, while the denominator is defined by the energy of all the other global states. By summing all the configurations of the hidden units, the marginal probability distribution of a visible unit is determined as shown in the following equation:

$$p(v) = \frac{\sum_h \exp(-E(v, h))}{\sum_{x, y} \exp(-E(x, y))}. \quad (3)$$

Contrastive divergence learning [17], [19], [46], [47] is used to train the network. Two phases are carried out in the learning

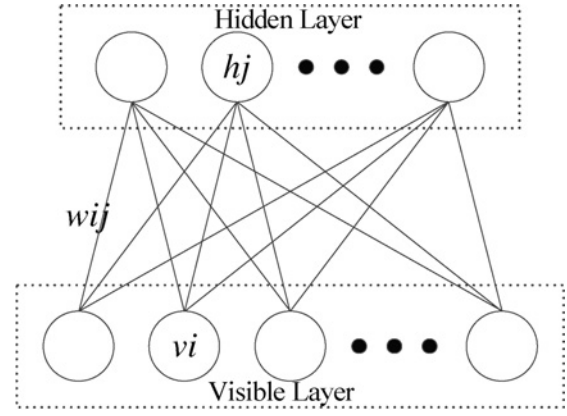


Fig. 1. Network architecture of an RBM.

process. In the first phase, the data vectors are input into the visible units. Subsequently, the hidden states are updated. In the second phase, the reverse procedure, where the visible states are reconstructed given the hidden states, is carried out. The same procedure is repeated until the stopping criterion is fulfilled. This training procedure is illustrated in Fig. 2. After the training process, the weights are updated according to the following equation:

$$\Delta w_{ij} = \epsilon (\langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_S) \quad (4)$$

where  $\langle v_i h_j \rangle_0$  is the average value of  $v_i h_j$  at the start of the training process (0-step reconstruction) and  $\langle v_i h_j \rangle_S$  is the average value of  $v_i h_j$  after  $S$ -step reconstruction.

#### B. REDA Work Flow

The adaptation of the RBM as an EDA in the MO framework (REDA) has been carried out in [16]. In this section, the main features of REDA, together with its mathematical formulation and process flow, will be presented. Without loss of generality, the MO optimization problem can be formulated for the minimization case as follows.

Minimize

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))$$

subject to  $\mathbf{x} \in \theta$

where  $\mathbf{x}$  consists of  $n$  decision variables,  $x = (x_1, x_2, \dots, x_n)^T$ ,  $\theta$  is the decision space,  $\mathbf{f}$  consists of  $m$  objective functions,  $f \in R^m$ , and  $R^m$  is the objective space.

The process flow of REDA is summarized in the pseudocode in Algorithm 1. First, at generation 0 ( $g = 0$ ),  $N$  individuals are generated by performing initialization with a marginal probability distribution of 0.5 [48], [49]. The generated solutions are stored in  $Pop(g)$ . All the solutions in  $Pop(g)$  are then evaluated to obtain their objective values. The fitness of the solutions is determined by the Pareto ranking and crowding distance [50]. The process continues by selecting promising solutions. Any selection mechanism proposed for standard MOEAs can be used. In this implementation, a binary tournament selection is applied [51], [52]. Each tournament involves two randomly selected individuals and the one with the smaller rank will be chosen. However, if both the solutions have the same rank, the solution with a

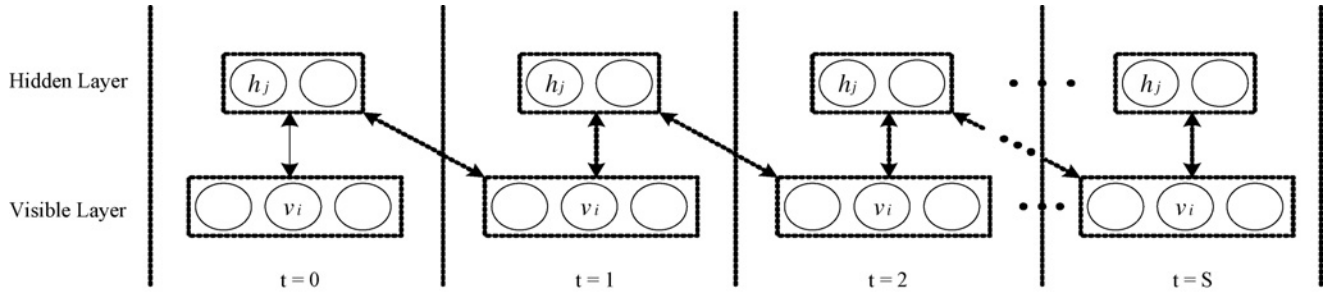


Fig. 2. Training procedure of an RBM based on contrastive divergence.

greater crowding distance will be selected. This selection procedure repeats until  $N$  solutions have been chosen.

From the selected individuals, an RBM network is built and then trained until the stopping criterion is reached. Note that the RBM network can handle both binary and real-valued input data. However, this paper is only concerned with the RBM with binary input. In the implementation,  $n$  decision variables in the test problems imply  $n$  input units of the RBM,  $N$  individuals in a population imply  $N$  training data of the RBM, and the alleles of a chromosome  $\mathbf{x}$  are identical to the input vector  $\mathbf{v}$  of the RBM. The number of hidden units is determined by the user. Two main procedures are involved in the construction of the probabilistic model. The first procedure involves the training of the network, while the second involves the construction of a probabilistic model. For network training, three phases are carried out. First, the positive phase constructs the states of the hidden units given the input values, by performing Gibbs sampling according to the following equation:

$$p(h_j|\mathbf{v}) = \varphi\left(\sum_i w_{ij}v_i - d_j\right) \quad (5)$$

where  $\varphi(y) = 1/(1 + e^{-y})$  is the logistic function. Second, the states of the input and hidden units are reconstructed based on (6) and (5), respectively

$$p(v_i|\mathbf{h}) = \varphi\left(\sum_j w_{ij}h_j - b_i\right). \quad (6)$$

Although an infinite number of reconstructions can theoretically be carried out, one-step reconstruction, as suggested in [20], is performed in this paper. Finally, the last phase updates the weights and biases of the network according to

$$w'_{ij} = w_{ij} + \epsilon(\langle v_i h_j \rangle_0 - \langle v_i h_j \rangle_1) \quad (7)$$

$$b'_i = b_i + \epsilon(\langle v_i \rangle_0 - \langle v_i \rangle_1) \quad (8)$$

$$d'_j = d_j + \epsilon(\langle h_j \rangle_0 - \langle h_j \rangle_1). \quad (9)$$

This training procedure is repeated until the stopping criterion is fulfilled. The stopping criterion is based on the maximum number of training epochs predefined by the user. The construction of the probabilistic model will begin as soon as

the training process terminates. The marginal probability of each decision variable is computed according to

$$p(v_i = 1) = \frac{\sum_{l=1}^N \delta_l(v_i^+) + \phi}{\sum_{l=1}^N \delta_l(v_i^+) + \sum_{l=1}^N \delta_l(v_i^-) + r_i \phi} \quad (10)$$

where  $N$  is the population size,  $\delta_l(v_i^+) = \sum_{j=1}^H e^{-E(v_i^+=1, h_j)}$  is the marginal cost of  $v_i^+$  when the cardinality of  $v_i^+ = 1$ ,  $H$  is the number of hidden units,  $\delta_l(v_i^-) = \sum_{j=1}^H e^{-E(v_i^-=0, h_j)}$  is the marginal cost of  $v_i^-$  when the cardinality of  $v_i^- = 0$ ,  $\phi = (\sum_{l=1}^N \delta_l(v_i)) / N$  is the average cost of cardinality, and  $r_i$  is the number of different values that  $v_i$  may take. In the binary case,  $r_i$  is 2. The pseudocode for the probabilistic modeling is summarized in Algorithm 2. Subsequently,  $N$  offspring solutions are generated using the simple sampling technique [48], [49] as follows:

$$x_i = \begin{cases} 1, & \text{if random}(0, 1) \leq p(v_i = 1) \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where  $x_i$  is the  $i$ th decision variable and  $\text{random}(0, 1)$  is a randomly generated value between  $[0, 1]$ . In order to avoid losing fitter solutions, the parent population  $Pop(g)$  is combined with the offspring population  $Pop2(g)$  in an archive with a size of  $2N$  individuals. Fitness is reassigned to all the solutions in the archive according to the Pareto ranking and crowding distance.  $N$  solutions with the lowest ranks or largest crowding distances are selected to form a new population  $Pop(g)$ . The simulation terminates when the stopping criterion for the evolution is met. Otherwise, the evolution process repeats with  $g = g + 1$ .

#### IV. INVESTIGATING PROBABILISTIC MODELING AND SAMPLING TECHNIQUES

Much research has been conducted to design new EDAs based on various machine learning approaches [53], [54] or probabilistic graphical models [55]–[59]. However, not much work has been conducted to analyze the structures and the behaviors of the probabilistic modeling and sampling techniques [60]–[62]. This knowledge can be used to design practical theoretical models, problem-based operators, and enhancement approaches [63]. In this section, we carry out an investigation on how well the states are being reconstructed by the RBM over different training epochs, how to effectively train an RBM, and what can be elucidated from the energy values of an RBM.

**Algorithm 1** Pseudocode of REDA**Begin**

1. **Initialization:** At generation  $g = 0$ , randomly generate  $N$  solutions with a marginal probability of 0.5 to form the initial population,  $Pop(g)$
  2. **Evaluation:** Evaluate all solutions in the population to obtain their objective values
- Do while (maximum number of fitness evaluations is not reached)**
3. **Fitness assignment:** Apply the Pareto ranking and crowding distance over the population. Each solution consists of two values which represent its fitness, one is the rank of the domination and another is the level of crowded
  4. **Selection:** Select  $N$  solutions using the binary tournament selection operator
  5. **Modeling:** Train an RBM and then build a probabilistic model  $p(x)$  to represent the probability distribution of the solutions
  6. **Sampling:** Apply simple probability sampling technique to sample  $N$  offspring solutions. Store them in  $Pop2(g)$
  7. **Evaluation:** Calculate the objective values of all offspring solutions in  $Pop2(g)$
  8. **Archiving:** Store the parents  $Pop(g)$  and child solutions  $Pop2(g)$  in an archive. Perform the Pareto ranking and crowding distance over the solutions in the archive
  9. **Elitism:** Select  $N$  solutions with the lowest Pareto rank or highest crowding distance from the archive to form the new population  $Pop(g + 1)$ .  $g = g + 1$

**End Do****End***A. State Reconstruction in an RBM*

In an RBM, the neurons between two layers are fully connected via weighted synaptic connections. However, there is no intra-layer connection. These weight connections are used by the neurons to communicate their activations to one another. During the learning process, the activation states of the network comprise a Boltzmann probability distribution. The quality of training of the network corresponds directly to the effectiveness at which the algorithm learns the probability distribution. Moreover, the network captures the energy of the data to elucidate the relationships between the decision variables. This distribution-based model allows the RBM to globally learn the probability distribution of the decision variables by considering the interdependencies of the data.

In order to understand what is reconstructed by an RBM model and how well the network can reconstruct the data points (input data), the distributions of the input and reconstructed points (100 data points) in the decision space for the POL problem [64] are presented in Fig. 3. Dark circles are the input data and blank circles are the reconstructed data. The POL problem is chosen because it only consists of two decision variables, which allows the decision frontier to be easily visualized. The training error at generation  $g$  is measured by the Euclidean distance between the real data points (input data) and the reconstructed data points calculated according to the following equation:

$$Error_g = \sum_{i=0}^N \sum_{j=0}^n (x_{i,j} - x'_{i,j})^2 \quad (12)$$

**Algorithm 2** Pseudocode of the probabilistic modeling in REDA**Begin**

%%Network Training (implemented in the fifth step of Algorithm 1)  
**Do while (maximum number of training epochs is not reached)**

%%Positive Phase

1. Construct the conditional probability of the hidden units given the visible (input) values  $p(h_j|\mathbf{v})$  according to (5)
2. From  $p(h_j|\mathbf{v})$ , sample the states of the hidden units  $< h_j >_0$

%%Negative Phase

3. Construct the conditional probability of the visible units given the states of the hidden units  $p(v_i|\mathbf{h})$  according to (6). Reconstruct the states of the visible units  $< v_i >_1$  by sampling the constructed conditional probability of (6)
4. Construct the conditional probability of the hidden units given the sampled visible (input) values  $p(h_j|\mathbf{v})$  according to (5). Reconstruct again the states of the hidden units  $< h_j >_1$  by sampling the constructed conditional probability of (5)

%%Updating of weights

5. Update the weights and biases according to (7)–(9)

**End Do**

%%Construction of probabilistic model

6. Compute energy values for all solutions in the population according to (1)
7. Compute the marginal probability according to (10).

**End**

where  $N$  is the population size,  $n$  is the number of decision variables,  $x_{i,j}$  is the real data point, and  $x'_{i,j}$  is the reconstructed data point. From Fig. 3, it is observed that with light training (upper left subfigure), the training error is 1400, indicating that the algorithm may not correctly model the distribution of the data. The reconstruction is improved with further training (upper right subfigure) that reduces the training error to 750.

When the number of hidden neurons (50 units) and training epochs (10000 epochs) are sufficiently large, the network succeeded in reconstructing most of the real data points (lower right subfigure). In this case, the training error is 25. Therefore, an RBM is able to model the exact data points and the distribution of the solutions is captured in the synaptic weights of the network. This observation that a sufficient number of hidden units in the network would guarantee improvement in the training error has been proven mathematically in [43]. In other words, the network can represent any discrete distribution exactly when the number of hidden units is very large.

*B. Change in Energy Function Over Generations*

The weight update process in an RBM requires calculating the gradient of log-likelihood of the input data. The gradient is minimal when the reconstructed data is exactly similar to the input stimuli. Contrastive divergence training [46] aims to obtain the weights of the network that minimize the energy level and training error of the network. The primary understanding is that the minimal energy level and training error can be achieved when sufficient number of hidden units and training epochs are applied. This is because the learning capability of the network is determined by the number of hidden units. A larger number of hidden units gives extra flexibility for the network to model the global distribution of the input stimuli, and thus could yield better convergence. On the other hand, contrastive divergence training will require a large number of training epochs to train the network well.

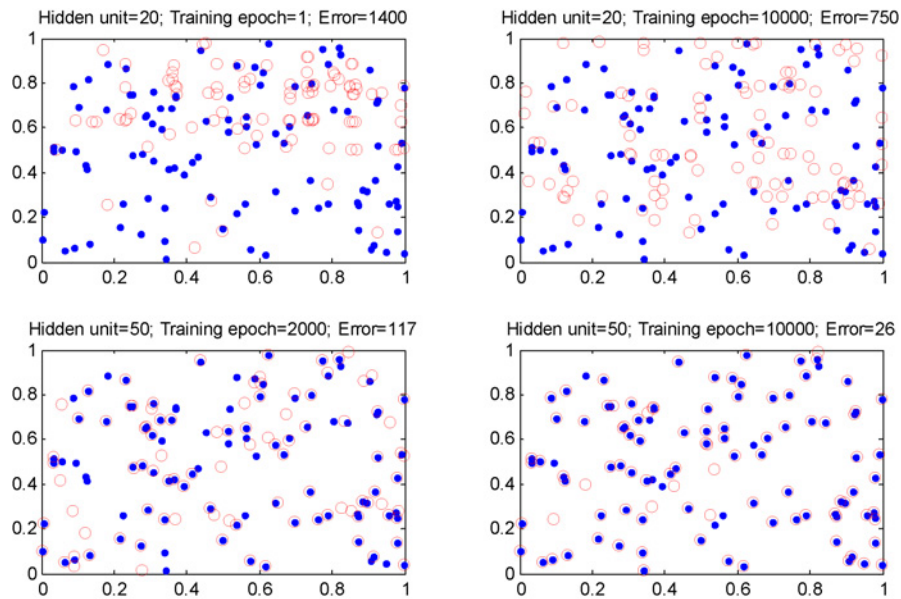


Fig. 3. Distribution plots of the input data points (dark circles) and reconstructed data points (blank circles) generated by an RBM. Different number of hidden units and training epochs are used, which result in different training errors and, thus, different sets of reconstructed data.

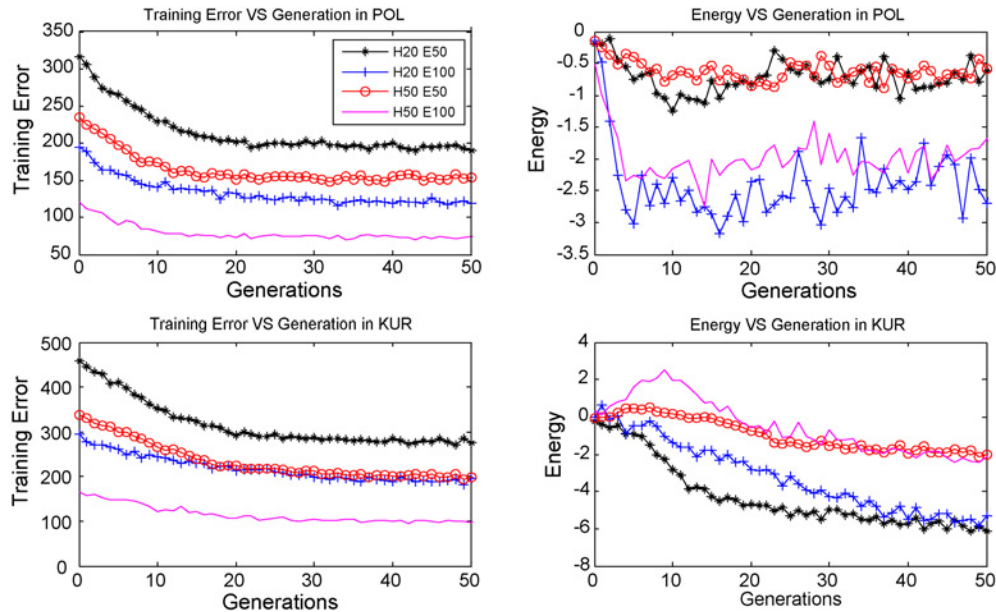


Fig. 4. Training error and energy value versus generation produced by an RBM for different numbers of hidden units and training epochs.

When the RBM is modeled as an EDA, another factor that can reduce the energy level and training error is the number of generations of an optimization process. Over generations, the training error and energy level of the network are reduced. This is shown in Fig. 4, where H20 E50 represents an RBM setting consisting of 20 hidden units and 50 training epochs. The data is obtained by running REDA on POL and KUR [65] problems. The results are averaged over 10 independent runs with a population size of 100. Note that POL has two decision variables, while KUR has three decision variables. From Fig. 4, it is observed that the energy level and training error are reduced over generations. This observation suggests that extensive training is unnecessary during the earlier generations because the network more accurately models the distribution of the solutions toward the end of the evolution. This is most

likely due to the reduction in the size of more promising search regions when the search converges to near optimal points. By taking this into consideration, the computational time of the algorithm can be improved by eliminating unnecessary training of the network in each generation.

The energy values are also reduced over generation. However, the decrement in the training error is not proportional to the decrement in the energy equilibrium of the network (the upper figures). The network's energy equilibrium is dependent on the input and hidden states. This complex relationship prevents training epochs and hidden units from directly influencing the energy equilibrium of the network. This may be caused by the fact that certain choices of number of hidden units and training epochs may cause the network to be overtrained or trapped in local optima. Thus, the setting of

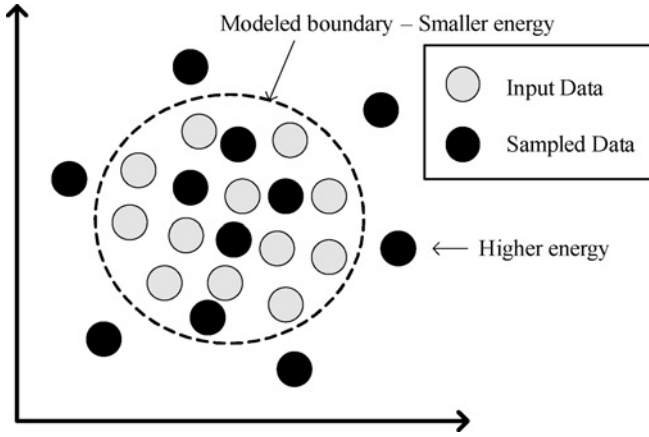


Fig. 5. Description of solutions with different energy levels in the objective space of an MO function. The x-axis is the first objective function and the y-axis is the second objective function.

the hidden units should be carefully chosen. In [16], hidden units of 5–20 are suggested.

#### C. What Can Be Elucidated From the Energy Values of an RBM?

In EDAs, the two main mechanisms that determine the success of the algorithms are probabilistic model construction and sampling technique. The core purpose of the probabilistic modeling is to learn the probability distribution of the candidate solutions by considering the dependencies between the decision variables. By using the probability density of the known solutions, the probability distribution of the unknown solutions can be studied. In EDAs, the parents are the known solutions while the offspring are the unknown solutions. If the characteristics of the offspring solutions can be predicted, this additional information can be taken into consideration during the optimization process.

In an RBM, the energy-based model captures the probability distribution of the parent solution set by associating a scalar energy value from the network to each solution. Over the training and evolutionary processes, the energy is reduced to a certain level of thermal equilibrium. Thus, we can elucidate that the solutions that are located inside the boundary regions of the parent solutions have a lower energy value. On the other hand, the solutions that are located outside of the boundary regions of the parent solutions may have a higher energy level. In pattern recognition, a lower energy level suggests that a test sample is more likely to belong to a certain class of patterns. However, this is not the case in EDAs as a lower energy level does not mean that the solutions are fitter, and vice versa. Fig. 5 explains this claim. Fig. 5 shows a set of input data (parent solutions) that was modeled by an RBM and a set of sampled data (offspring solutions) that was sampled from an RBM in an objective space with two objective functions. The solutions located inside the modeled boundary may have a lower energy value, and the solutions located outside of the modeled region may have a higher energy level. If a selection scheme only selects solutions with a lower energy, the exploitation can be enhanced; however, the exploration is poor since the search only focuses on the boundary regions that have been modeled

by the RBM. On the other hand, if the selection scheme only selects solutions with a higher energy level, the exploration can be enhanced. However, the exploitation is poor. Thus, a selection scheme should give the chance to select both solutions with lower and higher energy levels. The proposed energy-based sampling mechanism in this paper is based on this observation.

## V. PROPOSED ALGORITHMS

### A. Proposed Sampling Mechanism

One of the main characteristics of REDA is its capability of learning the multivariate dependencies between the decision 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 of each decision variable or input unit in the network. The offspring for the following generation is subsequently sampled from the constructed probabilistic model. The simple sampling technique applied in REDA may, however, limit the production of appropriate solutions if the decision variables are highly correlated or have a high dimension. This is because, during sampling, marginal probability distribution considers the distribution of the particular decision variable, but not the correlation between the decision variables. As a result, the sampled solutions have difficulties following the correlated distribution. One way to tackle this problem is to sample an infinite number of solutions. This may increase the number of possible combinations of the solutions and thus increase the chance of producing fitter individuals. However, sampling an infinitely large number of solutions may lead to an increase in the number of fitness evaluations and computational time. It is known that some real-world problems are very time consuming and such an algorithm would not be practical. To deal with this problem, an energy value is taken into consideration. First,  $N \times M$  solutions are generated. Then, the energy value will serve as the main criterion for forming new  $N$  solutions from the alleles of the  $N \times M$  solutions, where  $M > 1$  is a multiplier. A lower energy level implies that the solution is in a more stable state, while a higher energy level means that the solution is not in energy equilibrium. The energy-based sampling mechanism will, therefore, prefer the alleles of solutions with lower energy levels.

As probabilistic modeling only models the previous best topology, the solutions that are located inside the modeled topology are stable (lower energy level) in terms of energy equilibrium and are generally fit. On the other hand, the solutions outside the modeled topology (higher energy level) may be considered unstable but not unfit (as discussed in Section IV-C). This means that the solutions with higher energy levels may be the promising solutions that are not modeled by the network and thus will be worth preserving to the next generation. Therefore, it is required to give the algorithm the flexibility of choosing the alleles of solutions with high energy levels to achieve a more explorative search.

Based on this argument, an energy-based sampling mechanism is proposed (Algorithm 3). This mechanism is applied in the sixth step of the REDA's pseudocode in Algorithm 1. It



**Algorithm 3** Pseudocode of the energy-based sampling mechanism**Begin**

%%Replace the sixth step of the REDA pseudocode in Algorithm 1

%%Given the marginal probability of each decision variable,  $p(v_i)$ 

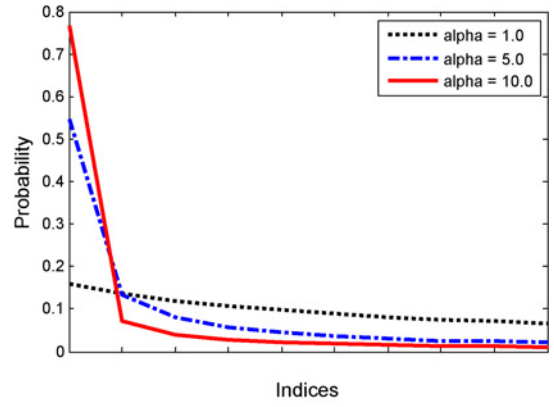
1. Sample  $N \times M$  solutions according to (11) and store them in  $Z$
2. Construct the conditional probability of the hidden units given the sampled visible data in  $Z$  according to (5). Reconstruct the states of hidden units of all solutions produced in step. 1, by sampling the constructed conditional probability of (5)
3. Compute the energy values for all solutions according to (1)
4. Sort the energy values in an increasing order
5. Apply selection procedure to form new  $N$  individuals as the final offspring

**End**

should be highlighted that binary tournament selection (step 4 of Algorithm 1) is not applied in this paper since losing some promising solutions and allowing certain solutions to be selected several times, may not be advantageous. Referring to Algorithm 3, first,  $N \times M$  solutions are sampled according to (11) and stored in  $Z$ , which is then used as input for the RBM. Then, the corresponding states for each hidden unit  $h_j$  are constructed by sampling the constructed conditional probability of (5). Subsequently, the energy value for each of the solutions in  $Z$  is computed according to (1). After which, the energy values, along with their corresponding indices, are sorted in an increasing order. Finally, a selection procedure is applied to form new  $N$  solutions using the alleles of the  $N \times M$  solutions.

The selection procedure is crucial in forming the new solutions. In this paper, we propose an inverse exponential selection scheme (IESS). Under this scheme, there is a higher probability of selecting the alleles of individuals with lower energy values. The pseudocode for this scheme is presented in Algorithm 4, where  $Pop(t)_{i,j}$  is the new  $i$ th solution's  $j$ th bit position at generation  $t$  and  $Z_{RandC,j}(t)$  is the sorted  $RandC$ th solution's  $j$ th bit position at generation  $t$ .  $min$  and  $max$  determine the range of the random values. In order to allow the flexibility of changing the range of the random values,  $\alpha$  is added to the algorithm. For simplicity,  $min$  and  $max$  are permanently assigned values of 0.01 and 1.0, respectively. In this way,  $\alpha$  is the only parameter that determines the probability at which an allele of an individual will be selected. The probabilities of selecting each allele of the individuals for different values of  $\alpha$  are shown in Fig. 6.

In the figure, the  $x$ -axis refers to the index of solutions, in which the solutions are sorted in an increasing order of energy levels, and the  $y$ -axis is the probability for each solution to be selected. It is observed that a smaller value of  $\alpha$  will result in a more even chance of selecting any allele of the individuals. The probability of selecting the alleles of the individuals with lower energy values increases and the probability of selecting the alleles of the solutions with higher energy values decreases with the increase in the value of  $\alpha$ . This scheme is designed based on the observation that solutions with lower energy values are located in the topology modeled by the previously selected population. These solutions are moderately fit. When

Fig. 6. Selection probability of IESS with different values of  $\alpha$ .**Algorithm 4** Pseudocode of the inverse exponential selection scheme (IESS)**Begin**

%%Implemented in the fifth step of the energy-based sampling mechanism in Algorithm 3

**For**  $i = 1 : N$ **For**  $j = 1 : n \times b$  $RandB = \text{random}(\alpha \times min, \alpha \times max)$  $P = \text{Exp}(RandB) - \text{Exp}(\alpha \times min)$  $Q = \frac{N \times M}{\text{Exp}(\alpha \times max) - \text{Exp}(\alpha \times min)}$  $RandC = P \times Q$  $Pop(t)_{i,j} = Z_{\lfloor RandC \rfloor, j}(t)$ **End For**  $j$ **End For**  $i$ **End**where  $n$  is the number of decision variables,  $b$  is the number of bits per variable, and  $\text{random}()$  is the uniform random number

the newly formed population consists mostly of the alleles of such solutions, the overall probabilistic model will not produce any solution that is far from promising regions. Furthermore, individuals with higher energy values located outside of the previously modeled region may be fit or unfit. Therefore, the alleles of these solutions are given a lower probability of being selected to further increase the exploration capability of the algorithm.

**B. A Hybrid REDA With Energy-Based Sampling Mechanism**

Even though MOEDAs have promising search capability, many researchers reported that EDAs are weak in generating a wide set of solutions [16], [22]. Thus, enhancement schemes that can improve the limitation of EDAs are required. In this section, the REDA with an energy-based sampling mechanism (REDA-E) is hybridized with a GA and a local search based on evolutionary gradient approach (HREDA-E). For the hybridization with GA, a simple hybridization scheme is applied by allowing  $T\%$  of the solutions in a population to be produced by GA, while the others are produced by REDA-E. Since REDA-E is a binary-based optimizer, the GA with a binary scheme is employed. In this paper, a single-point crossover and a bit-flip mutation are applied. REDA-E is also



---

**Algorithm 5** Pseudocode of an evolutionary gradient search algorithm

---

**Begin**

1. **Input:** Define initial step size  $\sigma_{t=0}$

**Do while (Stopping criterion is not met)**

For  $j=1:LS$  (Number of solutions undergoing local search)

2. **Initial solution:** Select a solution  $\mathbf{x}^j$  from the selection pool

3. **Reproduction:** Create  $L$  local neighbors  $\mathbf{r}^i, i \in (1, 2, \dots, L)$  by perturbing  $\mathbf{x}^j$  using normal mutation  $N(0, \sigma_t^2)$

4. **Evaluation:** Calculate the objective values of  $\mathbf{r}^i, \mathbf{f}(\mathbf{r}^i)$

5. **Direction:** Estimate the global gradient direction as follows:

$$\hat{\mathbf{v}} = \frac{\sum_{i=1}^L [\mathbf{f}(\mathbf{r}^i) - \mathbf{f}(\mathbf{x}^j)](\mathbf{r}^i - \mathbf{x}^j)}{\left\| \sum_{i=1}^L [\mathbf{f}(\mathbf{r}^i) - \mathbf{f}(\mathbf{x}^j)](\mathbf{r}^i - \mathbf{x}^j) \right\|}$$

6. **Offspring generation:**

$$\mathbf{y} = \mathbf{x}^j - \sigma_t \hat{\mathbf{v}}$$

7. **Mutation step size update:**

$$\sigma_{t+1} = \begin{cases} \sigma_t \varepsilon & \text{if } \mathbf{f}(\mathbf{y}) < \mathbf{f}(\mathbf{x}^j) \\ \sigma_t / \varepsilon & \text{otherwise} \end{cases}$$

8. **Solution update:**

if  $\mathbf{f}(\mathbf{y}) < \mathbf{f}(\mathbf{x}^j)$  then

$\mathbf{x}^j = \mathbf{y}$

end if

9. **Output:** Output  $\mathbf{x}^j$

**End For**

**End Do**

**End**

---

hybridized with an evolutionary gradient search (EGS) [66]. The EGS is a local search that uses the gradient information of the trajectory of solutions to predict the direction of movements in the search space. The pseudocode of the EGS is presented in Algorithm 5. In the implementation, we allow  $L\%$  of generations to activate the local search and  $LS\%$  of solutions to be allowed to undergo the local search. The overall process flow of the proposed algorithm, named a hybrid REDA-E (HREDA-E), is presented in Algorithm 6.

## VI. EXPERIMENTAL SETUP

This section presents the MO benchmark test problems used to evaluate the effectiveness of the proposed algorithm. Thirty-one benchmark test problems are outlined. The effectiveness of the algorithms is measured using IGD and NR. Finally, the parameter settings and implementations of the simulation runs are also presented.

### A. Test Problems

Thirty-one benchmark test instances with different characteristics have been selected to test the performance of the proposed algorithm. The test problems are 5 ZDT [42], 5 ZDT Type-1 [67], 5 ZDT Type-2 [67], 5 ZDT Type-3 [67], 3 DTLZ [68], 4 UF [69], and 4 WFG [70]. All of them are minimization problems. The characteristics of the test problems are summarized in Table I. In the table, the number of decision variables ( $n$ ), objective functions ( $m$ ), Pareto front geometries,

---

**Algorithm 6** Pseudocode of a hybrid REDA with energy-based sampling mechanism (HREDA-E)

---

**Begin**

1. **Initialization:** At generation  $g = 0$ , randomly generate  $N$  solutions with a marginal probability of 0.5 to form the initial population  $Pop(g)$

2. **Evaluation:** Evaluate all solutions in the population to obtain their objective values

**Do while (maximum number of fitness evaluations is not reached)**

3. **Fitness assignment:** Apply the Pareto ranking and crowding distance over the population. Each solution consists of two values which represent its fitness, one is the rank of the domination and another is the level of crowded

4. **Reproduction:** Build a probabilistic model  $p(x)$  to represent the distribution of the solutions (required by REDA). Perform the energy-based sampling mechanism

**For**  $i = 1 : N$

Generate a random value between  $[0,1]$  ( $u$ )

**if**  $u < T\%$

Create an offspring using IEES

**else**

Generate an offspring using GA. Binary tournament selection is applied to select parents for mating

**end if**

**End For**

5. **Evaluation:** Calculate the objective values of all child solutions

6. **Archiving:** Store the parent and child solutions in an archive. Perform the Pareto ranking and crowding distance over the solutions in the archive

7. **Elitism:** Select  $N$  solutions with the lowest Pareto rank or highest crowding distance from the archive to form the new population  $Pop(g+1)$ .  $g = g + 1$

8. **Local Search:** Generate a random number between  $[0,1]$  ( $u$ ) **if**  $u < L\%$  (% of generations to activate the EGS)

**For**  $i = 1 : N$

Generate another random value between  $[0,1]$  ( $u$ )

**if**  $u < LS\%$  (% of solutions undergoing local search)

Perform EGS to generate an offspring

**end if**

**End For**

**end if**

Perform archiving and elitism to the parent and child solutions to form the new population  $Pop(g)$ .

**End Do**

**End**

---

and other characteristics, including separable or nonseparable, unimodal or multimodal, and bias or nonbias, are highlighted.

Note that the ZDT Type-1–3 test problems are a variant of ZDT test problems that involve different types of transformations [67]. In this way, artificial linkages between the decision variables are introduced. Introducing linkage dependencies between the decision variables is one way of increasing the difficulty of a problem [50], [71], [72]. The idea is to introduce a transformation matrix that converts original variables  $\mathbf{x}$  into the corresponding variables  $\mathbf{y}$ . In this way, each of the variables is influenced by some other variables, and their relationships are described by the transformation matrix.

- 1) *Type-1 problems:* In these problems, the linkages are introduced between the decision variables in  $\mathbf{x}_I$  or  $\mathbf{x}_{II}$ , but no linkage is formed between the decision variables in  $\mathbf{x}_I$  and  $\mathbf{x}_{II}$ , where  $\mathbf{x}_I$  is the set of de-

TABLE I  
TEST INSTANCES

Instance	$m$	$n$	Geometry	SP/NS	U/M	Bias
ZDT1	2	100	Convex	SP	U	NO
ZDT2	2	100	Concave	SP	U	NO
ZDT3	2	100	Disconnected	SP	M	NO
ZDT4	2	100	Convex	SP	M	NO
ZDT6	2	100	Concave	SP	M	YES
ZDT1 Type-1-3	2	100	Convex	NS	U	NO
ZDT2 Type-1-3	2	100	Concave	NS	U	NO
ZDT3 Type-1-3	2	100	Disconnected	NS	M	NO
ZDT4 Type-1-3	2	100	Convex	NS	M	NO
ZDT6 Type-1-3	2	100	Concave	NS	M	YES
DTLZ1	3	20	Linear	SP	M	NO
DTLZ2	3	20	Concave	SP	U	NO
DTLZ3	3	20	Concave	SP	M	NO
UF1	2	30	Convex	NS	M	YES
UF2	2	30	Convex	NS	M	YES
UF3	2	30	Convex	NS	M	YES
UF4	2	30	Concave	NS	M	YES
WFG1	2	24	Concave, Mixed	NS	U	YES
WFG2	2	24	Convex, Disconnected	NS	U	NO
WFG3	2	24	Linear, Degenerate	NS	M	NO
WFG4	2	24	Concave	NS	M	NO

$n$  is the number of decision variables.  $m$  is the number of objective functions. Geometry is the shape of the Pareto optimal front. SP refers to separable. NS refers to non-separable. U refers to unimodal. M refers to multimodal.

cision variables in cost function I and  $\mathbf{x}_{II}$  is the set of decision variables in cost function II. The modified ZDT1 with Type-1 linkages is described as follows:

$$\begin{aligned}
 f_1(\mathbf{x}) &= x_1 \\
 f_2(\mathbf{y}) &= g(\mathbf{y}) \left( 1 - \left( \frac{f_1(\mathbf{x})}{g(\mathbf{y})} \right)^{\frac{1}{2}} \right) \\
 g(\mathbf{y}) &= 1 + \frac{9 \left( \sum_{i=2}^n y_i \right)}{n-1} \\
 \mathbf{y} &= \mathbf{T}\mathbf{x}, \mathbf{x} \in (x_2, \dots, x_n)
 \end{aligned}$$

where  $\mathbf{T}$  is an  $(n-1) \times (n-1)$  matrix with uniformly distributed values between  $[0,1]$ . In this type of problem, there is no linkage between the decision variables of cost functions  $f_1$  and  $f_2$ . The explicit linkages are only introduced in the  $g$  function, which is a function of  $f_2$ . Therefore, all the decision variables, except  $x_1$ , are explicitly linked. The same modifications are made to the ZDT2 to ZDT6 problems.

- 2) *Type-2 problems*: In Type-2 problems, linkages are introduced between all the decision variables, thereby making the first cost function correlated to the second cost function. The modified ZDT1 problem with Type-2 linkages is similar to Type-1 problems, except the transformation matrix  $\mathbf{T}$  is an  $n \times n$  matrix. The same modifications are made to the ZDT2 to ZDT6 problems.

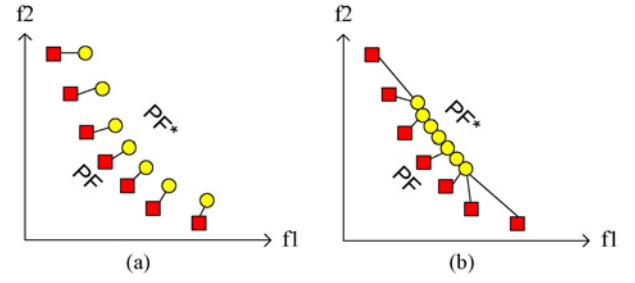


Fig. 7. Calculation of IGD for (a) evenly distributed and (b) unevenly distributed solutions.

- 3) *Type-3 problems*: The above transformation matrices involve only linear transformations. In order to further increase the difficulty of the problem, Type-3 problems involve non-linear mappings between  $\mathbf{y}$  and  $\mathbf{x}$ , presented as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \mathbf{T} \cdot \begin{bmatrix} \sqrt{x_1} \\ \sqrt{x_2} \\ \vdots \\ \sqrt{x_n} \end{bmatrix}$$

The same modifications are also made to the ZDT2 to ZDT6 problems.

UF test problems are a set of test problems used in CEC09 competition for unconstrained MO optimization. WFG test problems are a set of state-of-the-art and difficult test problems. Four position parameters and 20 distance parameters were applied.

### B. Performance Indicators

IGD and NR indicators are used to measure the performance of the algorithm.

1) *Inverted Generational Distance*: IGD [23], [73] measures the proximity and spread of the obtained front ( $PF^*$ ), compared to the Pareto optimal front ( $PF$ ). Mathematically, it is described as follows:

$$IGD = \frac{\sum_{\vartheta \in P} d(\vartheta, P^*)}{|P|} \quad (13)$$

where  $P$  is the set of uniformly distributed solution points on the Pareto optimal front,  $P^*$  is the set of obtained solution points, and  $d(\vartheta, P^*)$  is the Euclidean distance between solution  $\vartheta$  and the nearest solution in  $P^*$ . In (13), each of the candidate solutions in  $PF$  is compared with the solutions in the obtained front ( $PF^*$ ) to obtain the minimum Euclidean distance in the objective space, as illustrated in Fig. 7(a). In Fig. 7(b), it is observed that for obtained solutions that are not well-distributed along the Pareto optimal front, the IGD will be greater than that of a better-distributed set of solutions, as shown in Fig. 7(a).

2) *Non-Dominance Ratio*: NR [24] measures the quality of the solutions from various algorithms in the domination point of view. In this measurement, the solutions from all algorithms are pooled together and the non-dominated solutions are marked. Subsequently, the ratio of the non-dominated solutions

contributed by an algorithm is computed. The ratio of non-dominated solutions is the number of non-dominated solutions produced by an algorithm over the total number of non-dominated solutions that have been marked. The mathematical formulation is presented in

$$NR(PF_1^*, PF_2^*, \dots, PF_k^*) = \frac{|B \cap PF_1^*|}{|B|} \quad (14)$$

where  $|B|$  is the total number of non-dominated solutions composed from all the algorithms ( $k$  algorithms) that are involved in the comparison, and  $PF_1^*$  is the solution set under evaluation.

### C. Implementation

Five state-of-the-art algorithms, including REDA, NSGA-II, MOEA/D-DE, MMEA, and MOUMDA, are compared against the HREDA-E. REDA [16] is the predecessor of HREDA-E that uses a simple sampling technique. NSGA-II [50] is chosen since it is one of the famous MOEAs and the proposed algorithm utilizes most of its operators. The single point crossover and bit-flip mutation is used as the genetic operators of the NSGA-II. MOEA/D-DE [74] is a decomposition-based MOEA with differential evolution (DE) as its genetic operators. This algorithm was awarded the best algorithm in unconstrained MO optimization in the CEC09 competition.

MMEA [35] is one of the recently developed MOEDAs, which employs the regularity patterns of the solutions in predicting the Pareto front. MOUMDA is another MOEDA based on univariate marginal distribution algorithm (UMDA) [36]. The basic architecture of MOUMDA is quite similar to REDA. The main difference is that REDA utilizes multivariate modeling while MOUMDA uses univariate modeling. All the codes were written in C++ and the simulations were performed on an Intel(R) Core(TM)2 Duo CPU, 3.0 GHz. The experimental settings are listed in Table II.

## VII. EXPERIMENTAL STUDIES AND DISCUSSION

In this section, comparative studies are conducted to evaluate the performance of the six algorithms on the 31 benchmark test instances. Furthermore, investigations are carried out to analyze the effects of  $M$  and  $\alpha$  on the effectiveness of the proposed energy-based sampling mechanism. Finally, computational time analysis is conducted. The empirical results are presented using IGD and NR performance indicators. A smaller IGD value implies better performance in terms of closer proximity of the obtained solutions to the Pareto optimal front and a wider distribution of the obtained front along the Pareto optimal front. NR measures the ratio of non-dominated solutions among all the solutions generated by all the algorithms, and a higher value indicates that the algorithm produces more nondominated solutions.

### A. Comparison Results in Terms of IGD

Table III shows the results generated by six of the algorithms in terms of the IGD performance indicator. The mean and standard deviation of the IGD for the ten independent simulation runs are tabulated. The number in the parenthesis just beside the standard deviation is the rank of the algorithms.

TABLE II  
PARAMETER SETTINGS

Parameter	Setting
Population size	100 for problems with two objective functions, 300 for problems with three objective functions
Stopping criterion (number of fitness evaluations)	$300 \times \text{population size}$
Number of independent runs	10
$\alpha$ in HREDA-E	5.5
Multiplier $M$ in HREDA-E	10
T% in HREDA-E	50%
L% in HREDA-E	50%
LS% in HREDA-E	10%
Learning rate in RBM	0.1
Number of hidden units in RBM	20 for all variants of ZDT test problems and 5 for other test problems
Number of training epochs in RBM	Ten for all test problems
Crossover rate in GA	0.8
Crossover rate in DE	0.5
Mutation rate in GA	$\frac{1}{(\text{Variable\_size} \times \text{Variable\_bit})}$
Number of bits per variable in GA, HREDA-E, and REDA	15
Neighboring solutions in MOEA/D-DE	20
Parameter settings for MMEA	Same as [35]

ZDT are a set of simple test problems. Thus, we increase the decision variables from 30 (suggested in [42]) to 100 to test the performance of the algorithms in a large-dimensional search space. It is observed that HREDA-E outperforms the other algorithms in all ZDT test problems. The second best algorithm is REDA, which is the predecessor of HREDA-E. The performance of REDA is enhanced when the energy-based sampling technique and the hybrid mechanism are incorporated, resulting in the best performance of HREDA-E. MMEA, which is an EDA that takes into account the regularity patterns of the solutions, fails to yield promising results in ZDT test problems. This may be due to the lack of clear regularity patterns of the solution set, especially at the early stages of evolution. MOUMDA models the marginal probability distribution of each decision variable, without considering any linkage information. It is thus unable to model an overall fit solution set. As for MOEA/D-DE, the DE operator succeeds in exploring a wide range of search regions; however, its exploitation ability to locate the individuals at the optimal search space is weaker than NSGA-II. For ZDT4, which is a highly multimodal test problem, all the algorithms are unable to produce a global Pareto optimal front. It is likely that the algorithms are trapped in some local optima. This result suggests that REDA is unable to deal well with this kind of problem. Whenever a few solutions are trapped in any local optimum, the network will model the distribution of these suboptimal regions, which will then be treated as promising regions. A hybrid mechanism is one of the ways to enhance the performance of REDA. Thus, the solution set generated by HREDA-E in ZDT4 is the closest one from the Pareto optimal front.

TABLE III  
SIMULATION RESULTS IN TERMS OF IGD VALUES OF VARIOUS ALGORITHMS ON ALL TEST INSTANCES

Instance	Algorithm					
	NSGA-II	MOEA/D-DE	MOUMDA	MMEA	REDA	HREDA-E
ZDT1	0.1314±0.0131(3)	0.4877±0.0475(5)	0.4373±0.1364(4)	0.5834±0.0412(6)	0.0178±0.0017(2)	<b>0.0042±0.0002(1)</b>
ZDT2	0.2133±0.0241(3)	1.1738±0.1291(5)	1.8690±0.1562(6)	1.1193±0.5212(4)	0.0332±0.0061(2)	<b>0.0045±0.0002(1)</b>
ZDT3	0.1062±0.0122(3)	0.5611±0.0681(5)	0.4335±0.0684(4)	0.6303±0.0249(6)	0.0282±0.0043(2)	<b>0.0053±0.0007(1)</b>
ZDT4	23.152±0.8489(3)	35.475±8.3686(4)	92.802±22.020(5)	95.752±26.880(6)	22.769±2.6157(2)	<b>0.1227±0.2554(1)</b>
ZDT6	4.1087±0.0682(3)	5.5828±0.1884(4)	6.6278±0.1725(6)	5.8760±0.3590(5)	2.4995±0.1165(2)	<b>0.0100±0.0023(1)</b>
ZDT1 Type-1	7.5037±0.5983(3)	25.698±4.2359(4)	75.600±11.383(6)	68.583±4.5372(5)	0.6243±0.1239(2)	<b>0.0063±0.0010(1)</b>
ZDT2 Type-1	9.8248±1.1122(3)	31.6601±5.5187(4)	77.782±14.059(5)	88.900±8.9666(6)	1.2391±0.2210(2)	<b>0.0106±0.0044(1)</b>
ZDT3 Type-1	7.0095±1.0475(3)	22.0505±4.0181(4)	79.251±6.7916(6)	67.081±3.1825(5)	0.5347±0.1209(2)	<b>0.0153±0.0103(1)</b>
ZDT4 Type-1	2434.2±431.22(3)	4701.4±1323.3(4)	6870.1±1557.6(5)	9919.6±2217.9(6)	541.70±113.29(2)	<b>264.68±23.173(1)</b>
ZDT6 Type-1	9.8471±0.2569(3)	14.101±0.5784(4)	16.546±0.3900(6)	15.870±0.4053(5)	2.2844±0.2680(2)	<b>0.0136±0.0046(1)</b>
ZDT1 Type-2	8.1279±1.1468(3)	25.180±4.4080(4)	76.783±7.7830(6)	69.251±4.8772(5)	0.6468±0.1628(2)	<b>0.0069±0.0019(1)</b>
ZDT2 Type-2	9.0214±0.7981(3)	29.586±6.0725(4)	74.573±16.043(5)	90.591±10.650(6)	1.1474±0.1396(2)	<b>0.0088±0.0019(1)</b>
ZDT3 Type-2	7.4139±0.7053(3)	23.817±4.5437(4)	77.619±6.8995(6)	67.628±5.4665(5)	0.5114±0.1071(2)	<b>0.0144±0.0083(1)</b>
ZDT4 Type-2	2413.4±526.87(3)	4017.1±828.49(4)	8052.7±1974.1(5)	11311±1533.6(6)	471.68±73.836(2)	<b>235.59±115.38(1)</b>
ZDT6 Type-2	8.0574±0.2621(3)	12.4935±0.3415(4)	14.8362±0.5115(6)	14.132±0.4265(5)	0.9559±0.3759(2)	<b>0.0238±0.0098(1)</b>
ZDT1 Type-3	46.807±4.0511(4)	34.446±8.7706(3)	178.78±6.8318(5)	191.24±13.903(6)	3.7280±0.5458(2)	<b>0.0239±0.0052(1)</b>
ZDT2 Type-3	50.934±2.9915(4)	33.481±5.4321(3)	172.343±10.762(5)	199.50±15.909(6)	5.0979±0.8894(2)	<b>0.1131±0.0607(1)</b>
ZDT3 Type-3	46.311±3.5323(4)	35.275±7.1135(3)	173.51±13.774(5)	190.83±13.860(6)	4.4629±0.5598(2)	<b>0.0487±0.0300(1)</b>
ZDT4 Type-3	5609.0±1147.9(4)	3210.4±779.42(3)	37261±5090.4(5)	48913±8672.1(6)	697.32±109.52(2)	<b>398.28±230.41(1)</b>
ZDT6 Type-3	12.732±0.1713(3)	12.887±0.7342(4)	18.123±0.2223(5)	18.468±0.2165(6)	2.3780±0.6626(2)	<b>0.3217±0.0795(1)</b>
DTLZ1	7.2726±1.9702(2)	9.2158±11.1997(3)	200.79±9.4261(5)	217.09±25.310(6)	23.671±4.9476(4)	<b>4.8016±5.3398(1)</b>
DTLZ2	0.0366±0.0014(2)	<b>0.0330±0.0005(1)</b>	0.0475±0.0039(5)	0.0528±0.0033(6)	0.0467±0.0051(4)	0.0418±0.0025(3)
DTLZ3	32.094±8.4911(3)	14.968±38.7895(2)	593.52±46.872(5)	665.12±35.469(6)	81.909±27.143(4)	<b>4.7336±7.4685(1)</b>
UF1	0.1395±0.0326(6)	<b>0.0763±0.0338(1)</b>	0.1171±0.0130(4)	0.1071±0.0214(2)	0.1717±0.0413(5)	0.1108±0.0329(3)
UF2	0.0734±0.0108(3)	0.0633±0.0462(2)	0.1043±0.0050(4)	0.2319±0.0298(6)	0.1141±0.0074(5)	<b>0.0536±0.0045(1)</b>
UF3	0.332±0.0333(5)	<b>0.1072±0.0434(1)</b>	0.2292±0.0116(3)	0.2173±0.0423(2)	0.4716±0.0532(6)	0.3016±0.0359(4)
UF4	0.0676±0.0029(2)	0.0859±0.0075(3)	0.1777±0.0035(6)	0.1014±0.0234(4)	0.1338±0.0125(5)	<b>0.058±0.0030(1)</b>
WFG1	1.4047±0.2267(4)	1.2398±0.0074(3)	2.0316±0.0274(6)	1.5469±0.0708(5)	1.1891±0.0215(2)	<b>1.1441±0.0081(1)</b>
WFG2	0.2112±0.0188(5)	0.1796±0.0281(3)	0.1818±0.0053(4)	0.1721±0.0437(2)	0.2991±0.0362(6)	<b>0.146±0.0623(1)</b>
WFG3	0.2251±0.0236(5)	<b>0.1726±0.0041(1)</b>	0.2065±0.0106(4)	0.1811±0.0030(2)	0.3313±0.0343(6)	0.1862±0.0093(3)
WFG4	0.0474±0.0151(2)	0.0839±0.0058(4)	0.1755±0.0373(6)	0.0974±0.0027(5)	0.0544±0.0099(3)	<b>0.0328±0.0056(1)</b>

The ZDT1 Type-1 test problems are a variant of the ZDT test problem after introducing linkage dependencies between the decision variables. It is observed that HREDA-E has the best optimization performance in terms of IGD, compared to the other algorithms in the comparison. Since the optimization performance of MMEA and MOUMDA is weak in ZDT test problems, its performance is expected to be weak as well in ZDT Type-1 test problems. Table III also shows that the IGD values generated by all algorithms in ZDT Type-1 test problems are higher than the IGD values in ZDT test problems. This indicated that the introduction of the linkage dependencies to the ZDT test problems has increased their difficulties. A similar observation can be made for ZDT Type-2 test problems. In ZDT Type-3 test problems, the performance of MOEA/D-DE is better than that of NSGA-II. Since ZDT Type-3 problems involve a non-linear transformation matrix, it can be concluded that MOEA/D-DE is less sensitive than NSGA-II in this particular case. The performance of all the algorithms deteriorates in Type-3 problems, where the IGD values were greater than those for the corresponding Type-1 and Type-2 problems. This is attributed to the introduction of a non-linear transformation matrix, as in Type-3 problems, whereas a linear transformation matrix is applied in Type-1 and Type-2 problems. Overall, HREDA-E achieves the best

performance in all types of ZDT test problems even through its generated front in ZDT4, ZDT4's variants, ZDT6, and ZDT6's variants can be improved further to obtain a better approximate Pareto optimal front.

DTLZ test problems can be scaled up to any number of decision variables and objective functions. Scalable problems will not be dealt with here since they are not within the scope of this paper. As such, the problems are fixed to three objective functions and 20 decision variables. Table III shows that HREDA-E obtains the best IGD values in DTLZ1 and DTLZ3, while MOEA/D-DE generates the best IGD value in DTLZ2. DTLZ1 and DTLZ3 are multimodal test instances. The hybrid mechanism in the proposed algorithm is the main operator that helps HREDA-E escape from trapping at the local optima, thus yielding the best performance in DTLZ1 and DTLZ3. DTLZ2 is one of the simple test problems. The IGD values indicate that all algorithms are able to approximate a near Pareto optimal front. However, the performance of MOEA/D-DE gives the best solution set since the solution set generated by this algorithm is the most evenly distributed one. This is due to the decomposition behavior of the algorithm that decomposes an MO optimization problem into multiple subproblems, in which the subproblems are constructed by using a set of uniformly generated weight vectors.

In UF test problems (CEC09 unconstrained test problems), the optimal solutions consist of a complicated Pareto set. According to [74], the ability to generate a wide set of solutions is one of the ways to efficiently solve the test problems. The results in Table III indicated that MOEA/D-DE obtains the best IGD values in UF1 and UF3, while HREDA-E yields the best IGD values in UF2 and UF4. By summing the rank of the algorithms, MOEA/D-DE has the best rank (sum of rank = 7), followed by HREDA-E (sum of rank = 9), and MMEA (sum of rank = 14).

In WFG test problems, HREDA-E yields the best performance in WFG1, WFG2, and WFG4, while MOEA/D-DE has the best performance in WFG3. The WFG are challenged by the different transformations between the decision variables. The diversity maintenance and the ability to generate a wide variety of solutions is also a main challenge of the test problems. The REDA, with an energy-based sampling mechanism, gives more emphasis on exploitation of the promising search regions while giving a smaller change of the solutions with large energy levels to survive to increase the exploration capability. The exploration and exploitation capabilities of the algorithm are further enhanced when hybridization is implemented. Thus, HREDA-E is able to obtain promising results in WFG test problems.

The convergence curve in terms of IGD values of the algorithms in seven of the test problems is shown in Fig. 8. The figure shows that REDA has a faster convergence rate at the early stages of evolution in ZDT1, ZDT3 Type-1, ZDT4 Type-2, and WFG1 test problems. However, HREDA-E outperforms REDA and other algorithms at the later stages of evolution. This is most probably due to the fact that the poor diversification in REDA prevents it from searching a wide set of optimal solutions. This is different in HREDA-E in which more efforts are paid to explore the search space to enhance the diversification of the algorithm. Thus, HREDA-E outperforms REDA at the later stages of evolution. In DTLZ3 and UF4, MOEA/D-DE shows a faster convergence at the early stages of evolution. HREDA-E outperforms MOEA/D-DE at the later stages of evolution. Even though MOEA/D-DE has a promising diversification feature, its convergence is poorer than HREDA-E. Thus, HREDA-E shows a good performance at the later stages of evolution.

### B. Comparison Results in Terms of NR

Table IV shows the results for six of the algorithms in terms of NR performance indicator. The mean and standard deviation of the IGD for the ten independent simulation runs are tabulated. The number in the parenthesis just beside the standard deviation is the rank of the algorithms.

In ZDT and its variant test problems, it is clear that almost every non-dominated solution is generated by HREDA-E, and a small number of non-dominated solutions are generated by REDA in ZDT3, ZDT6 Type-2, and ZDT4 Type-3. The other algorithms fail to generate any non-dominated solutions. Referring to the IGD values in Table III, HREDA-E has much smaller IGD values compared to other algorithms. This indicates that the convergence of HREDA-E in ZDT and its variants test problems is excellent. This is due to the strong

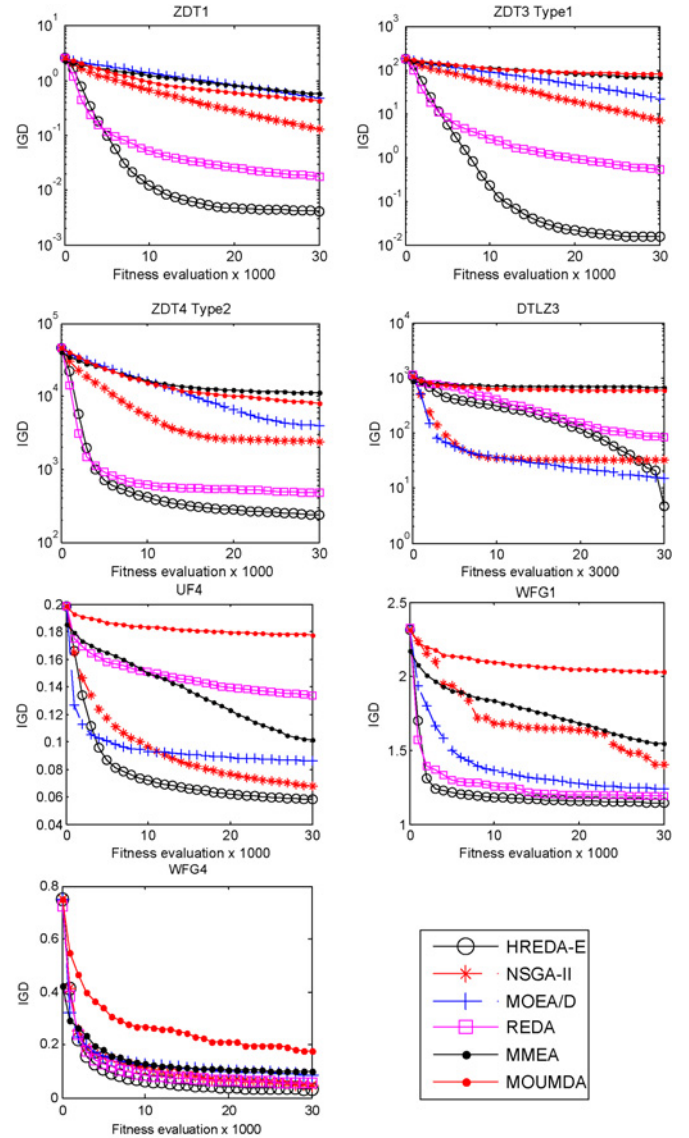


Fig. 8. Convergence trace of various algorithms in terms of IGD measurement.

exploration and exploitation capabilities of the REDA with an energy-based sampling scheme and a hybrid mechanism.

In DTLZ test problems, HREDA-E obtains the best results in DTLZ1 and DTLZ3, while NSGA-II has the best NR result in DTLZ2. In DTLZ2, all algorithms are able to generate a Pareto front that is close to the true Pareto front as indicated by a lower value of IGD, as shown in Table III. Thus, none of the algorithms has a high ratio of non-dominated solutions. As such, each algorithm contributes a particular number of non-dominated solutions.

In UF test problems, HREDA-E evolves more non-dominated solutions in UF1 and UF4, while MOEA/D-DE generates more non-dominated solutions in UF2 and UF4. In terms of the ranking, HREDA-E and MOEA/D-DE share the same rank (sum of rank = 9) and are followed by NSGA-II (sum of rank = 10), MMEA (sum of rank = 15), MOUMDA (sum of rank = 18), and REDA (sum of rank = 19). Even though REDA has a strong exploitation capability, its exploration ability is weak. This is due to the nature of

TABLE IV  
SIMULATION RESULTS IN TERMS OF NR VALUES OF VARIOUS ALGORITHMS ON ALL TEST INSTANCES

Instance	Algorithm					
	NSGA-II	MOEA/D-DE	MOUMDA	MMEA	REDA	HREDA-E
ZDT1	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0020±0.0042(2)	<b>0.9980±0.0042(1)</b>
ZDT2	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT3	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0058±0.0150(2)	<b>0.9942±0.0150(1)</b>
ZDT4	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT6	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT1 Type-1	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT2 Type-1	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT3 Type-1	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT4 Type-1	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT6 Type-1	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT1 Type-2	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT2 Type-2	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT3 Type-2	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT4 Type-2	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000(1)</b>
ZDT6 Type-2	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0109±0.0187(2)	<b>0.9891±0.0187(1)</b>
ZDT1 Type-3	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000 (1)</b>
ZDT2 Type-3	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000 (1)</b>
ZDT3 Type-3	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000 (1)</b>
ZDT4 Type-3	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0000±0.0000(3)	0.0000±0.0000(3)	0.1000±0.3162(2)	<b>0.9000±0.3162 (1)</b>
ZDT6 Type-3	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	0.0000±0.0000(2)	<b>1.0000±0.0000 (1)</b>
DTLZ1	0.2103±0.3961(3)	0.2218±0.3533(2)	0.0000±0.0000(4)	0.0000±0.0000(4)	0.0000±0.0000(4)	<b>0.5678±0.4687(1)</b>
DTLZ2	<b>0.2958±0.0206(1)</b>	0.2466±0.0199(2)	0.0838±0.0295(5)	0.0421±0.0155(6)	0.0975±0.0493(4)	0.2342±0.0344(3)
DTLZ3	0.0010±0.0032(4)	0.4742±0.3529(2)	0.0000±0.0000(5)	0.0265±0.0127(3)	0.0000±0.0000(5)	<b>0.4983±0.3581(1)</b>
UF1	0.2597±0.0750(2)	0.1996±0.0660(3)	0.0021±0.0045(6)	0.1477±0.1045(4)	0.0510±0.0455(5)	<b>0.3398±0.0759(1)</b>
UF2	0.1974±0.0372(2)	<b>0.6629±0.0278(1)</b>	0.0229±0.0160(4)	0.0007±0.0023(6)	0.0060±0.0118(5)	0.1100±0.0320(3)
UF3	0.0000±0.0000(4)	<b>0.7489±0.1696(1)</b>	0.0773±0.0225(3)	0.1737±0.1754(2)	0.0000±0.0000(4)	0.0000±0.0000(4)
UF4	0.2056±0.0669(2)	0.0358±0.0489(4)	0.0000±0.0000(5)	0.0437±0.0861(3)	0.0000±0.0000(5)	<b>0.7149±0.1162(1)</b>
WFG1	<b>0.5130±0.1667(1)</b>	0.0000±0.0000(4)	0.0000±0.0000(4)	0.0000±0.0000(4)	0.0080±0.0134(3)	0.4791±0.1663(2)
WFG2	0.0000±0.0000(6)	0.2674±0.2032(2)	0.2146±0.2307(3)	0.0962±0.1627(4)	0.0000±0.0000(5)	<b>0.4218±0.3253(1)</b>
WFG3	0.0066±0.0124(5)	<b>0.4268±0.0766(1)</b>	0.3343±0.1373(2)	0.1249±0.0482(3)	0.0000±0.0000(6)	0.1042±0.1004(4)
WFG4	<b>0.4860±0.1696(1)</b>	0.0096±0.0181(4)	0.0000±0.0000(6)	0.0011±0.0024(5)	0.0641±0.1140(3)	0.4392±0.1904(2)

the probabilistic modeling and sampling technique that only focuses the search on a specific modeled region.

In WFG test problems, NSGA-II obtains the best ratio of non-dominated solutions in WFG1 and WFG4, while MOEA/D-DE and HREDA-E obtain the best NR results in WFG3 and WFG2, respectively. In terms of ranking, HREDA-E has the best rank (sum of rank = 9), followed by MOEA/D-DE (sum of rank = 11), NSGA-II (sum of rank = 13), MOUMDA (sum of rank = 15), MMEA (sum of rank = 16), and REDA (sum of rank = 17).

The convergence curve, in terms of NR values, of the algorithms in seven of the test problems is shown in Fig. 9. In ZDT1, REDA has the highest NR at the early stages of evolution. However, its ratio declines over generations due to the increment of NR of HREDA-E over generations. At the end of the simulations, all individuals evolved by HREDA-E dominate solutions evolved by the other five algorithms. A near-similar trend is observed in ZDT3 Type-1, ZDT4 Type-2, and ZDT6 Type-3, with the exception that some of the non-dominated solutions are generated by REDA at the later stages of evolution in ZDT4 Type-2 and some solutions are generated by MOEA/D-DE at the early stages of evolution in ZDT6 Type-3.

In DTLZ3, NSGA-II maintains a good number of solutions at the early stages of evolution, but the solutions were eventu-

ally dominated by those of MOEA/D-DE and HREDA-E. It is also observed that HREDA-E outperforms MOEA/D-DE at the final stage of evolution. This is due to the slow convergence of HREDA-E in this test problem. The hybrid mechanism in HREDA-E provides a strong search mechanism that increases the chance to escape from the local optima. In UF4, a large portion of non-dominated solutions is generated by MOEA/D-DE, while a smaller number of non-dominated solutions are generated by HREDA-E and NSGA-II at the early stages of evolution. As the search progresses, the NR for MOEA/D-DE declines while the NR for HREDA-E increases and HREDA-E finally outperforms other algorithms. In WFG1, REDA seems to have more non-dominated solutions at the early stages of evolution. However, HREDA-E and NSGA-II are able to eventually evolve a set of fitter solutions that is able to dominate most of the solutions produced by the other algorithms. At the later stages of evolution, the NR evolved by NSGA-II slightly outperforms HREDA-E.

### C. Effects of Different Mechanisms

HREDA-E is an improved version of REDA that includes an energy-based sampling mechanism and hybridization with a GA and an EGS. In this section, we would like to observe the effects of the hybridization and the energy-based sampling mechanism. Table V shows the IGD values



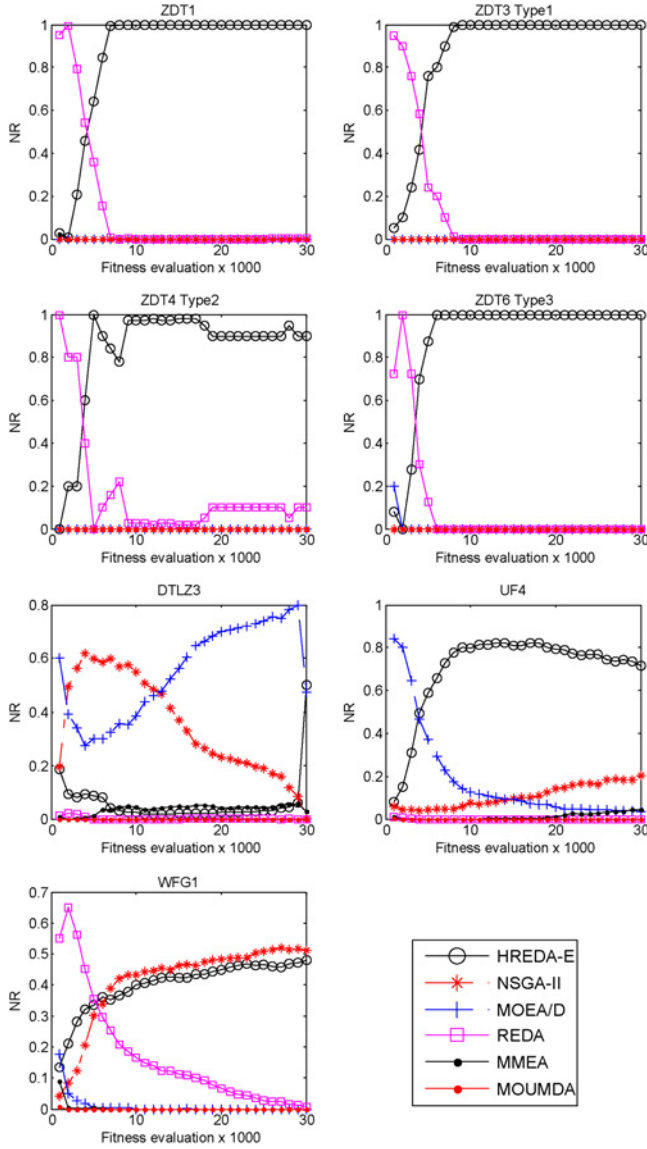


Fig. 9. Convergence trace of various algorithms in terms of NR measurement.

in 14 test instances for REDA, HREDA, and HREDA-E. Note that REDA is the original algorithm concerned; HREDA is the hybridization algorithm between REDA, GA, and EGS; HREDA-E is HREDA plus the energy-based sampling mechanism.

The results show that there is great improvement when REDA is hybridized with GA and EGS. This is most probably due to the exploration enhancement provided by the hybridization. REDA, which builds the probabilistic model of the selected solutions and then generates the solutions by sampling the constructed model, is weak in exploring search regions which are not modeled by the RBM. Furthermore, REDA only utilizes global information of the solutions (without using any location information) when producing the offspring. This may prevent the algorithm from exploiting the near-optimal regions. The hybridization with GA provides a level of exploration and exploitation while the incorporation of EGS may help REDA seek neighboring solutions that may

TABLE V  
SIMULATION RESULTS OF VARIOUS MECHANISMS IN TERMS OF IGD MEASUREMENT

Instance	Algorithm		
	REDA	HREDA	HREDA-E
ZDT2	0.0332±0.0061	0.0049±0.0003	0.0045±0.0002
ZDT6	2.4995±0.1165	0.0182±0.0068	0.0100±0.0023
ZDT1 Type-1	0.6243±0.1239	0.0071±0.0001	0.0063±0.0010
ZDT4 Type-1	541.70±113.29	365.43±147.48	264.68±23.173
ZDT2 Type-2	1.1474±0.1396	0.0128±0.0044	0.0088±0.0019
ZDT3 Type-2	0.5114±0.1071	0.0154±0.0135	0.0144±0.0083
ZDT4 Type-3	697.32±109.52	411.39±121.75	398.28±230.41
ZDT6 Type-3	2.3780±0.6626	0.3564±0.1181	0.3217±0.0795
DTLZ2	0.0467±0.0051	0.0487±0.0027	0.0418±0.0025
DTLZ3	81.909±27.143	12.000±13.907	4.7336±7.4685
UF2	0.1141±0.0074	0.0582±0.0098	0.0536±0.0045
UF3	0.4716±0.0532	0.3281±0.0403	0.3016±0.0359
WFG2	0.2991±0.0362	0.1668±0.0329	0.1460±0.0623
WFG3	0.3313±0.0343	0.1795±0.0078	0.1862±0.0093

enhance the diversification and exploitation of the algorithm. The performance of HREDA is further improved when the energy-based sampling mechanism with IESS is employed. The implementation of the IESS (HREDA-E) means that an allele of a solution with a lower energy value stands a higher chance of being chosen. This may enhance the exploitation capability of the algorithm. On the other hand, a smaller selection probability is given to the alleles of an individual with a higher energy value, thus increasing the exploratory capability of the algorithm.

#### D. Further Analysis

There are four parameters that will influence the optimization performance of REDA: 1) number of hidden units; 2) number of training epochs; 3) decay factor ( $\alpha$ ); and 4) multiplier ( $M$ ). The setting of the number of hidden units and training epochs have been studied in [16], which concluded that hidden units between 5 to 20 and training epochs of 10 would give a better optimization performance. In this paper, we follow the suggested settings. In this section, further studies are carried out to investigate the influence of  $\alpha$  and  $M$  on optimization performance. It is certain that the incorporation of the energy-based sampling mechanism in REDA will increase the computational time. A computational time analysis is also performed to show how costly the energy-based sampling mechanism is.

1) *Effect of  $\alpha$* :  $\alpha$  determines the decay factor of an exponential function ( $e^{\alpha x}$ ). If  $x$  is kept constant,  $\alpha$  will be the only parameter that determines the probability of selection as shown in Algorithm 4. In this section, the effect of different values of  $\alpha$  on optimization performance is examined. Fig. 10 shows the convergence traces of REDA-E using different values of  $\alpha$  to solve ZDT1 and DTLZ1 test problems. It is observed that an  $\alpha$  value that is too low (1.0) or too high (9.0) would not yield good results, while an  $\alpha$  value ranging from 3.0 to 7.0 gives acceptable performance. Recall that a lower value of  $\alpha$  will give a more even probability distribution function used for selecting the alleles of solutions, while a

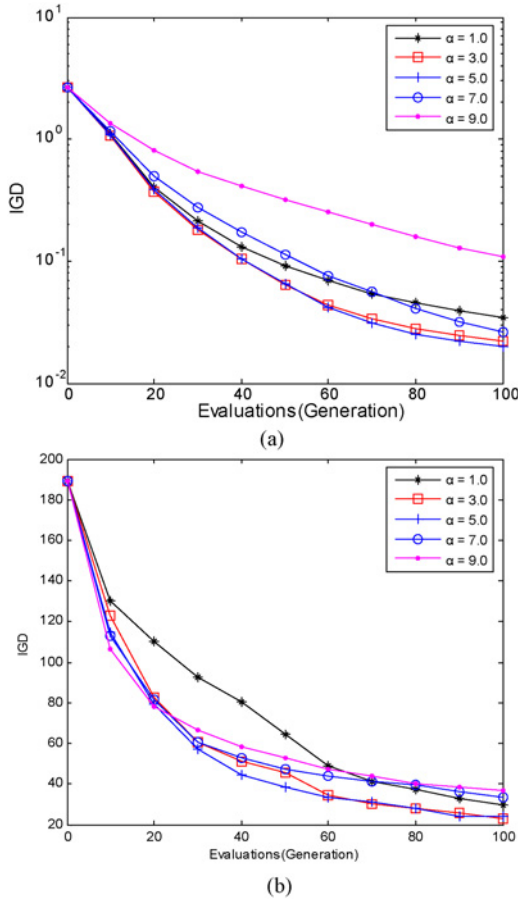


Fig. 10. Convergence traces of REDA-E for solving (a) ZDT1 and (b) DTL Z1 problems under different settings of  $\alpha$ .

higher value of  $\alpha$  will give a higher chance of selecting the alleles of solutions with lower energy values. A set of offspring will contain more unfit individuals if all the alleles of solutions are given almost equal chances of being selected. On the other hand, if the selection scheme only selects the alleles of individuals with lower energy, less exploration of the search space is performed by the algorithm. A set of offspring with too many alleles of solutions having low or high energies is not an ideal case in an evolutionary process since exploration and exploitation of the search space must be balanced. Therefore, an  $\alpha$  value ranging from 3.0 to 7.0 is the ideal setting for implementation.

2) *Effect of  $M$* : The fundamental idea behind the proposed energy-based sampling mechanism is that sampling the population for an infinitely large number of solutions may increase the chance of producing fitter candidate solutions. However, such a sampling scheme is impractical. Thus, a multiplier  $M$  is used to determine the number of sampled individuals. Note that the population size will also affect the influence of  $M$ . Thus, we fix the population size to be 100 and only varying the  $M$ . It may be argued that  $M$  can affect the final performance of the algorithm. Thus, this section carries out an investigation to examine the effect of  $M$  on optimization performance and then suggests a possible range of values for  $M$ . Fig. 11 shows the convergence traces of REDA-E using different values of  $M$  to solve ZDT1 and DTLZ1 test problems. It is observed

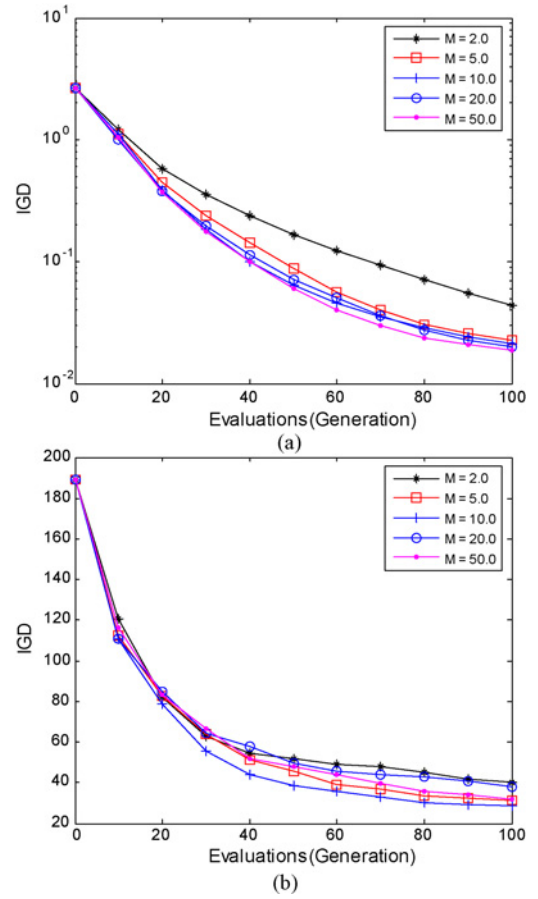


Fig. 11. Convergence traces of REDA-E for solving (a) ZDT1 and (b) DTL Z1 test problems under different settings of  $M$ .

that a larger value of  $M$  (20 to 50) gives better performance in solving ZDT1 but not DTLZ1 problems. A smaller value of  $M$  (2) leads to poorer performance while an  $M$  value around 5 to 10 generally gives good results for both the test instances. Therefore, it is possible to conclude that large values of  $M$  are unnecessary and an  $M$  value of around 5 to 10 is enough to evolve a good Pareto front. In fact, the  $M$  and  $\alpha$  settings are correlated. The purpose of these two parameters in the energy-based sampling mechanism is to assign a higher chance of selecting the alleles of solutions with lower energy values while still allowing some chance for the alleles of solutions with higher energy values to be selected.

3) *Computational Time Analysis*: From Section V, it is clear that the energy-based sampling mechanism will incur additional computational time due to the sampling of  $N \times M$  candidate solutions rather than  $N$  solutions, and the recomputation of the hidden states. A computational time analysis is carried out to determine how costly the proposed sampling technique is. Table VI presents the computational times required by one generation of REDA-E using different settings of  $M$  in solving ZDT1 and DTLZ1 test problems. From the table, the computational time is increased approximately three times when  $M$  is 50. However, only a slight increment in computational time is incurred for smaller values of  $M$  (2–10).

TABLE VI

COMPUTATIONAL TIME (IN SECONDS) USED BY REDA-E FOR SOLVING ZDT1 AND DTLZ1 TEST PROBLEMS UNDER DIFFERENT SETTINGS OF  $M$

Instance	Multiplier $M$					
	1	2	5	10	20	50
F1	2.0772	2.1510	2.3882	2.7551	3.4463	5.5789
F6	0.1670	0.1787	0.1958	0.2365	0.2981	0.5007

### VIII. CONCLUSION

In this paper, an energy-based sampling mechanism for REDA was proposed. The sampling approach took advantage of the energy information of solutions. Energy was used to determine a set of offspring to undergo the evolution process in the next generation. The set of selected offspring is a combination of the alleles of solutions with low and high energy values. The IESS assigned a larger probability of selecting the alleles of solutions with lower energy values and a smaller probability of selecting the alleles of individuals with higher energy values. The algorithm is then hybridized with a GA and EGS to further enhance the exploration and exploitation capabilities of the algorithm. The experimental results showed that the hybrid mechanism greatly improved the performance of REDA in terms of IGD and NR. The energy-based sampling mechanism with IESS further improved the performance of hybrid REDA in terms of IGD and NR, but at the expense of a longer computational time. Further analyses were also performed to examine the parameter settings of the energy-based sampling mechanism. Even though the proposed algorithm performs well in most of the test problems, the experimental results showed that the algorithm is still unable to converge to the global Pareto optimal front for problems with many local optima. Further investigations are currently being performed to overcome this limitation of the algorithm.

### REFERENCES

- [1] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Dordrecht, The Netherlands: Kluwer Academic, 2001.
- [2] J. A. Lozano, P. Larrañaga, and E. Bengoetxea, *Toward a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms* (Studies in Fuzziness and Soft Computing). Berlin, Germany: Springer, 2006.
- [3] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multi-objective estimation of distribution algorithm," in *Scalable Optimization via Probabilistic Modeling*. Berlin, Germany: Springer, 2006, pp. 223–248.
- [4] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions I: Binary parameters," in *Proc. 4th Int. Conf. Parallel Problem Solving Nature*, 1996, pp. 178–187.
- [5] S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning," Tech. Rep. CS-94-163, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.
- [6] P. A. N. Bosman and D. Thierens, "Continuous iterated density estimation evolutionary algorithm within the IDEA framework," in *Proc. Optimization Building Using Probabilistic Models OBUPM Workshop Genet. Evol. Comput. Conf.*, 2000, pp. 197–200.
- [7] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A survey of optimization by building and using probabilistic models," *Comput. Optimization Appl.*, vol. 21, no. 1, pp. 5–20, 2002.
- [8] C. Gueret, S. Schlobach, K. Dentler, M. Schut, and G. Eiben, "Evolutionary and swarm computing for the semantic web," *IEEE Comput. Intell. Mag.*, vol. 7, no. 2, pp. 16–31, May 2012.
- [9] S. Damas, O. Cordón, and J. Santamaría, "Medical image registration using evolutionary computation: An experimental survey," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 26–42, Nov. 2011.
- [10] A. E. I. Brownlee, M. Pelikan, J. A. W. McCall, and A. Petrovski, "An application of a multivariate estimation of distribution algorithm to cancer chemotherapy," in *Proc. 10th Annu. Conf. Genet. Evol. Comput.*, 2008, pp. 463–464.
- [11] E. Bengoetxea, "Inexact graph matching using estimation of distribution algorithms," Ph.D. dissertation, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2003.
- [12] R. Santana, P. Larrañaga, and J. A. Lozano, "Protein folding in simplified models with estimation of distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 12, no. 4, pp. 418–438, Aug. 2008.
- [13] M. Pelikan and D. E. Goldberg, "Hierarchical BOA solves ising spin glasses and MAXSAT," in *Proc. Int. Conf. Genet. Evol. Comput.*, 2003, pp. 1271–1282.
- [14] S. K. Shukla, "DEUME: A framework for an estimation of distribution algorithm based on Markov random fields," Ph.D. dissertation, School of Computing Robert Gordon Univ., Aberdeen, Scotland, 2006.
- [15] C. H. Chen, T. K. Liu, I. M. Huang, and J. H. Chou, "Multi-objective synthesis of six-bar mechanisms under manufacturing and collision-free constraints," *IEEE Comput. Intell. Mag.*, vol. 7, no. 1, pp. 36–48, Feb. 2012.
- [16] H. J. Tang, V. A. Shim, K. C. Tan, and J. Y. Chia, "Restricted Boltzmann machine based algorithm for multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 3958–3965.
- [17] G. E. Hinton, "What kind of a graphical model is the brain?" in *Proc. Int. Joint Conf. Artif. Intell.*, 2005, pp. 1765–1775.
- [18] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proc. 24th Int. Conf. Mach. Learning*, 2007, pp. 791–798.
- [19] T. Tieleman, "Training restricted Boltzmann machines using approximations to the likelihood gradient," in *Proc. 25th Int. Conf. Mach. Learning*, 2008, pp. 1064–1071.
- [20] M. Yasuda and K. Tanaka, "Approximate learning algorithm for restricted Boltzmann machines," in *Proc. Int. Conf. Comput. Intell. Modelling Control Autom.*, 2008, pp. 692–697.
- [21] Y. W. Teh and G. E. Hinton, "Rate-coded restricted Boltzmann machines for face recognition," in *Proc. 14th Annu. Neural Inform. Process. Syst. Conf.*, 2000, pp. 908–914.
- [22] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multi-objective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.
- [23] M. A. Villalobos-arias, G. T. Pulido, and C. A. Coello Coello, "A proposal to use stripes to maintain diversity in a multi-objective particle swarm optimizer," in *Proc. IEEE Swarm Intell. Symp.*, Jun. 2005, pp. 22–29.
- [24] C. K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.
- [25] P. A. N. Bosman and D. Thierens, "Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms," *Int. J. Approximate Reasoning*, vol. 31, no. 3, pp. 259–289, 2002.
- [26] M. Laumanns and J. Ocenasek, "Bayesian optimization algorithms for multi-objective optimization," in *Proc. 7th Int. Conf. Parallel Problem Solving Nature*, 2002, pp. 298–307.
- [27] H. Li, Q. Zhang, E. Tsang, and J. A. Ford, "Hybrid estimation of distribution algorithm for multi-objective knapsack problem," in *Proc. 4th Eur. Conf. Evol. Comput. Combinatorial Optimization*, 2004, pp. 145–154.
- [28] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multi-objective hBOA, clustering, and scalability," in *Proc. Int. Conf. Genet. Evol. Comput.*, 2005, pp. 663–670.
- [29] K. Sastry, D. E. Goldberg, and M. Pelikan, "Limits of scalability of multi-objective estimation of distribution algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2005, pp. 2217–2224.
- [30] H. Soh and M. Kirley, "moPGA: Toward a new generation of multi-objective genetic algorithms," in *Proc. IEEE Congr. Evol. Comput.*, July 2006, pp. 1702–1709.
- [31] M. Costa and E. Minisci, "MOPED: A multi-objective Parzen-based estimation of distribution algorithm for continuous problems," in *Proc. 2nd Int. Conf. Evol. Multi-Criterion Optimization*, 2003, pp. 282–294.

- [32] T. Okabe, Y. Jin, B. Sendhoff, and M. Olhofer, "Voronoi-based estimation of distribution algorithm for multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2004, pp. 1594–1601.
- [33] X. Zhong and W. Li, "A decision-tree-based multi-objective estimation of distribution algorithm," in *Proc. Int. Conf. Comput. Intell. Security*, 2007, pp. 114–118.
- [34] L. Marti, J. Garcia, A. Berlanga, and J. M. Molina, "Solving complex high-dimensional problems with the multi-objective neural estimation of distribution algorithm," in *Proc. 11th Annu. Conf. Genet. Evol. Comput.*, 2009, pp. 619–626.
- [35] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 1167–1189, Oct. 2009.
- [36] L. Zinchenko, M. Radecker, and F. Bisogno, "Multi-objective univariate marginal distribution optimisation of mixed analogue-digital signal circuits," in *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, 2007, pp. 2242–2251.
- [37] Y. Gao, X. Hu, H. Liu, and Y. Feng, "Multi-objective estimation of distribution algorithm combined with PSO for RFID network optimization," in *Proc. Int. Conf. Measuring Technol. Mechatronics Autom.*, 2010, pp. 736–739.
- [38] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Norwell, MA, USA: Kluwer, 2002.
- [39] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
- [40] K. C. Tan, E. F. Khor, and T. H. Lee, *Multi-Objective Evolutionary Algorithms and Applications*. Berlin, Germany: Springer, 2005.
- [41] X. Shen, M. Zhang, and T. Li, "A multi-objective optimization evolutionary algorithm addressing diversity maintenance," in *Proc. Int. Joint Conf. Comput. Sci. Optimization*, vol. 1, Apr. 2009, pp. 524–527.
- [42] E. Zitzler, "Evolutionary algorithms for multi-objective optimization: Methods and applications," Ph.D. dissertation, Computer Engineering and Networks Laboratory, Swiss Federal Institut. Technol., Zurich, 1999.
- [43] N. Le Roux and Y. Bengio, "Representational power of restricted Boltzmann machines and deep belief networks," *Neural Comput.*, vol. 20, no. 6, pp. 1631–1649, 2008.
- [44] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Sci.*, vol. 9, no. 1, pp. 147–169, 1985.
- [45] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in Boltzmann machines," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 282–317.
- [46] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Artificial Intelligence and Statistics*. Society for Artificial Intelligence and Statistics, Barbados, 2005, pp. 17–22.
- [47] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [48] Y. Hong, Q. Ren, and J. Zeng, "Genetic drift in univariate marginal distribution algorithm," in *Proc. Conf. Genet. Evol. Comput.*, 2005, pp. 745–746.
- [49] Y. Hong, Q. Ren, and J. Zeng, "Adaptive population size for univariate marginal distribution algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2005, pp. 1396–1402.
- [50] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [51] M. Chakraborty and U. Chakraborty, "An analysis of linear ranking and binary tournament selection in genetic algorithms," in *Proc. 1st Int. Conf. Inform. Commun. Signal Process.*, 1997, pp. 407–411.
- [52] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Syst.*, vol. 9, no. 3, pp. 193–212, 1995.
- [53] J. Zhang, Z. Zhan, Y. Lin, N. Chen, Y. Gong, J. Zhong, H. Chung, Y. Li, and Y. Shi, "Evolutionary computation meets machine learning: A survey," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 68–75, Nov. 2011.
- [54] M. Er and R. Oentaryo, "Computational intelligence: Methods and techniques," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 76–78, Nov. 2011.
- [55] T. K. Paul and H. Iba, "Reinforcement learning estimation of distribution algorithm," in *Proc. Int. Conf. Genet. Evol. Comput.*, 2003, pp. 1259–1270.
- [56] L. R. Emmendorfer and A. Pozo, "Effective linkage learning using low-order statistics and clustering," *IEEE Trans. Evol. Comput.*, vol. 13, no. 6, pp. 1233–1246, Dec. 2009.
- [57] M. D. Platel, S. Schliebs, and N. Kasabov, "Quantum-inspired evolutionary algorithm: A multimodel EDA," *IEEE Trans. Evol. Comput.*, vol. 13, no. 6, pp. 1218–1232, Dec. 2009.
- [58] L. Qiang and Y. Xin, "Clustering and learning gaussian distribution for continuous optimization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 2, pp. 195–204, May 2005.
- [59] Q. Zhang, "On stability of fixed points of limit models of univariate marginal distribution algorithm and factorized distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 8, no. 1, pp. 80–93, Jan. 2004.
- [60] E. S. Correa and J. L. Shapiro, "Model complexity vs. performance in the Bayesian optimization algorithm," in *Proc. 9th Int. Conf. Parallel Problem Solving Nature*, 2006, pp. 998–1007.
- [61] C. F. Lima, M. Pelikan, D. E. Goldberg, F. G. Lobo, K. Sastry, and M. Hauschild, "Influence of selection and replacement strategies on linkage learning in BOA," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 1083–1090.
- [62] H. Wu and J. L. Shapiro, "Does over-fitting affect performance in estimation of distribution algorithms," in *Proc. 8th Annu. Conf. Genet. Evol. Comput.*, 2006, pp. 433–434.
- [63] M. Hauschild, M. Pelikan, K. Sastry, and C. Lima, "Analyzing probabilistic models in hierarchical BOA," *IEEE Trans. Evol. Comput.*, vol. 13, no. 6, pp. 1199–1217, Dec. 2009.
- [64] G. Winter, J. Périaux, M. Galan, P. Cuesta, C. Poloni, "Hybrid GA for multi-objective aerodynamic shape optimization," in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Périaux, M. Galan, P. Cuesta, Eds. Chichester, U.K.: Wiley, 1997, pp. 397–414.
- [65] F. Kursawe, "A variant of evolution strategies for vector optimization," in *Proc. 1st Workshop Parallel Problem Solving Nature*, 1991, pp. 193–197.
- [66] C. K. Goh, Y. S. Ong, K. C. Tan, and E. J. Teoh, "An investigation on evolutionary gradient search for multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, June 2008, pp. 3741–3746.
- [67] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *Proc. 8th Annu. Conf. Genet. Evol. Comput.*, 2006, pp. 1141–1148.
- [68] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. IEEE Congr. Evol. Comput.*, May 2002, pp. 825–830.
- [69] Q. Zhang, A. Zhou, P. N. Zhao, S. Suganthan, W. Liu, and S. Tiwari, "Multi-objective optimization test instances for the CEC 2009 special session and competition," School of Computer Science Electron. Engineering, Trondheim, Norway, Tech. Rep. CES-487, 2009.
- [70] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multi-objective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.
- [71] K. Deb and A. Agrawal, "Understanding interactions among genetic algorithm parameters," in *Proc. 5th Workshop Found. Genet. Algorithms*, 1998, pp. 265–286.
- [72] K. Deb, "Multi-objective genetic algorithm: Problem difficulties and construction of test problems," *Evol. Comput.*, vol. 7, no. 3, pp. 205–230, 1999.
- [73] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a Pareto front," in *Proc. Late Breaking Papers Genetic Programming Conf.*, 1998, pp. 221–228.
- [74] H. Li and Q. Zhang, "Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.



**Vui Ann Shim** received the Ph.D. degree in electrical engineering from the National University of Singapore, Singapore, in 2012.

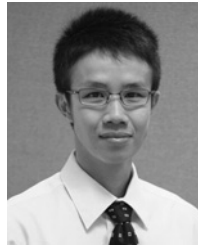
He is currently a Scientist with the Robotics Program, Institute for Infocomm Research, A\*STAR, Singapore. His current research interests include computational intelligence, computational neuroscience, multi-objective optimization, and robotics.



**Kay Chen Tan** is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore. He has published over 200 journal and conference papers and co-authored five books.

Prof. Tan has been invited to be a Keynote/Invited Speaker for 30 international conferences. He has served on international program committees for over 100 conferences and has been involved in organizing committees for over 40 international conferences. He is currently a Distinguished Lecturer of the IEEE

Computational Intelligence Society and the Editor-in-Chief of the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE. He also serves as an Associate Editor/Editorial Board Member of over 15 international journals, such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, *Evolutionary Computation* (MIT Press), the *European Journal of Operational Research*, the *Journal of Scheduling*, and the *International Journal of Systems Science*. He was a fellow of the NUS Teaching Academy.



**Chun Yew Cheong** received the Ph.D. degree in electrical engineering from the National University of Singapore in 2010.

Since 2010, he has been a Scientist with the Computing Science Department, Institute of High Performance Computing, A\*STAR, Singapore. His current research interests include high-performance computing, evolutionary computation, and multi-objective optimization.