Probabilistic Graphical Approaches for Learning, Modeling, and Sampling in Evolutionary Multi-objective Optimization

Vui Ann Shim and Kay Chen Tan

Department of Electrical and Computer Engineering,
National University of Singapore, 4 Engineering Drive 3, 117576, Singapore
{g0800438,eletankc}@nus.edu.sg

Abstract. Multi-objective optimization is widely found in many fields, such as logistics, economics, engineering, or whenever optimal decisions need to be made in the presence of tradeoff between two or more conflicting objectives. The synergy of probabilistic graphical approaches in evolutionary mechanism may enhance the iterative search process when interrelationships of the archived data has been learned, modeled, and used in the reproduction. This paper presents the implementation of probabilistic graphical approaches in solving multi-objective optimization problems under the evolutionary paradigm. First, the existing work on the synergy between probabilistic graphical models and evolutionary algorithms in the multi-objective framework will be presented. We will then show that the optimization problems can be solved using a restricted Boltzmann machine (RBM). The learning, modeling as well as sampling mechanisms of the RBM will be highlighted. Lastly, five studies that implement the RBM for solving multi-objective optimization problems will be discussed.

Keywords: Evolutionary algorithm, multi-objective optimization, probabilistic graphical model, restricted Boltzmann machine.

1 Introduction

Many real-world problems involve the simultaneous optimization of several competing objectives and constraints that are difficult, if not impossible, to solve without the aid of powerful optimization algorithms. In a multi-objective optimization problem (MOP) [1, 2], no one solution is optimal to all objectives. Therefore, in order to solve the MOP, search methods employed must be capable of finding a number of alternative solutions representing the tradeoff among the various conflicting objectives. An MOP in minimization case can be formulated as follows. Minimize:

$$F(X) = (f_1(X), ..., f_m(X)) \tag{1}$$

subject to $X = \{x_1, ..., x_n\} \in \theta$, $F \in \mathbf{R}^m$, where θ is the decision space and $\mathbf{R}^m$ is the objective space.

J. Liu et al. (Eds.): WCCI 2012 Plenary/Invited Lectures, LNCS 7311, pp. 122–144, 2012.

Evolutionary algorithms (EAs) are a class of stochastic search methods that have been found to be efficient and effective in solving MOPs. The advantage of EAs can be attributed to their capabilities of sampling multiple candidate solutions simultaneously, a task that most conventional optimization techniques fail to work well. Nonetheless, the stochastic recombination in standard EAs may disrupt the building of strong schemas of a population and thus movement towards optimal is extremely difficult to predict. Motivated by the idea of exploiting the linkage information among the decision variables, estimation of distribution algorithm (EDA) has been regarded as a new computing paradigm in the field of evolutionary computation [3–5].

In EDAs, the discovered knowledge of the data is used to predict the location or pattern of the Pareto front or to predict the favorable movement in the search space. By using the discovered correlations of the parameters of a cost function, the search can be regulated to follow the correlated patterns when generating an offspring solution. The correlations as well as the probability distribution of the cardinalities of the parameters can be learned and modeled by a probabilistic model. In order to effectively learn and model that information, the probabilistic graphical approach is one of the well-known and promising techniques [6–8]. In EDAs, the reproduction of children solutions is carried out by building a representative probabilistic model of the parent solutions, and new solutions are then generated through the sampling of the constructed probabilistic model. Therefore, the learning, modeling, and sampling mechanisms are important features of an EDA.

This paper focuses on the implementation of EDAs for solving MOPs. First, a literature review that studies the potential of EDAs in solving MOPs is given. Then, the focus is tailored to introduce an EDA that uses the restricted Boltzmann machine (RBM) as its modeling technique. An insight discussion on how and what information is learned and modeled by the RBM will be covered. Subsequently, five studies that implement the RBM-based EDA (REDA) in solving scalable problems, epistatic problems, noisy problems, combinatorial problems, and global unconstrained continuous optimization problems will be presented.

The rest of the paper is as follows. Section 2 presents a literature review on the implementation of probabilistic graphical models (PGMs) in solving MOPs. Section 3 describes the RBM as well as its training and modeling mechanisms. Five case studies that implement the REDA in solving MOPs are presented in Section 4. The conclusion is drawn in Section 5.

2 Probabilistic Graphical Models in Multi-objective Evolutionary Algorithms

EDAs draw its inspiration from the use of the probability distribution of promising solutions to predict the Pareto optimal front or the favorable movement in the search space. Based on this idea, the linkage information or the regularity patterns that appear quite often in a set of promising solutions can be captured

and used to predict the probability distribution of other superior solutions. In the literature, the probability information can be captured using at least three methods depends on how the interactions among the decision variables are taken into consideration. Those methods are the univariate modeling, bivariate modeling, and multivariate modeling [3]. Over the last decade, several attempts have been devoted to developing EDAs in the context of multi-objective optimization (MOEDAs) [9]. The main differences among the MOEDAs are the employment of different modeling and sampling mechanisms.

The first MOEDA was introduced in [10]. The authors proposed a mixture-based multi-objective iterated density-estimation evolutionary algorithm (MIDEA) with both discrete and continuous representations. The mixture probability distribution in MIDEA was constructed using the univariate factorization. MIDEA is well-known for its simplicity, speed, and effectiveness. Furthermore, it can also serve as a baseline algorithm for other MOEDAs. The simulation results indicated that MIDEA is able to generate a set of diverse solutions that is close to the Pareto optimal front.

In [11], Laumanns and Ocenasek examined the effect of incorporating mutual dependencies between the decision variables in approximating the set of Pareto optimal solutions. The mutual dependencies were captured using the Bayesian optimization algorithm with binary decision trees. The experimental results showed that the proposed Bayesian multi-objective optimization algorithm (BMOA) is effective in approximating the Pareto front for simple test instances. In order to deal with harder test instances, an additional computational time is required.

In [12], MOEDA based on the Parzen estimator was introduced. The Parzen estimator, a non-parametric technique, was used to estimate the kernel density through the learning of the multivariate dependencies among the decision variables. The Parzen estimator was also used in the objective space to enhance the diversity preservation of the algorithm. The empirical results indicated that the proposed algorithm has better convergence rate and is able to obtain a set of well spread solutions.

Li et al. [13] suggested a hybrid binary EDA with mixture distribution (MOHEDA) for solving the multi-objective 0/1 knapsack problems. One of the simplest EDA, the univariate marginal distribution algorithm (UMDA), was hybridized with a weighted sum local search method. This hybridization enable MOHEDA took advantage of both local and global information to guide the search towards optimality. Furthermore, the population was clustered into several groups using a proposed stochastic clustering algorithm before the mixture distribution was constructed. In [14], Okabe et al. proposed an EDA that uses Voronoi diagram (VEDA) as its probabilistic modeling method. The Voronoi diagram takes into account the problem structure in estimating the most appropriate probability distribution. Instead of determining the distribution from individuals in the selected population, the implementation also makes use of those that are not selected. The experimental results showed that VEDA performs better than NSGA-II [15] for a limited number of fitness evaluations.

In [16], Pelikan et al. modified the Bayesian optimization algorithm and introduced a hierarchical Bayesian optimization algorithm (hBOA) to solve multi-objective decomposable problems. hBOA adapted the NSGA-II's domination-based framework and used k-mean clustering for modeling purpose. hBOA were used to solve scalable deceptive problems. The simulation results demonstrated that hBOA successes to obtain the optimal solutions and has faster convergence rate. They inferred that the clustering and linkage learning are the main criteria that contribute to the success of the algorithm in solving decomposable multi-objective problems. In [17], Sastry et al. proposed another MOEDA, called extended compact genetic algorithm (ECGA), to solve the scalable deceptive problems. The paper analyzed the characteristics of the algorithm on a class of bounding adversarial problems with scalable decision variables. m-k deceptive trap problems [16] were used to investigate the performance of the proposed algorithm.

In [18], Soh and Kirley proposed a parameter-less MOEA, which combines the ECGA with external ϵ-Pareto archive, clustering, and competent mutation to deal with scalable problems. Two types of scalable problems were studied, including deceptive problems with scalable decision variables and DTLZ problems with scalable objective functions. The proposed algorithm showed promising results in those test problems due to the incorporation of linkage learning, clustering, and local search. In [19], the authors examined the limitation of maximum-likelihood estimators in the problems which may lead to the prematurely vanishing variance. Using the framework of MIDEA, the authors combined the normal mixture distribution with adaptive variance scaling to remedy the vanishing variance problem. Under this scheme, the premature convergence was prevented in the condition that the estimated probability distribution is enlarged beyond its original maximum likelihood estimation. Zhong and Li [20] presented a decision-tree-based multi-objective estimation of distribution algorithm (DT-MEDA) for global optimization in the continuous-valued representation. The conditional dependencies among the decision variables are learned by the decision tree. The children solutions are generated through the sampling of the constructed conditional probability distribution.

Zhang et al. [21] proposed a regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) for solving continuous multi-objective optimization problems with linkage dependencies. A local principle component analysis (PCA) was used to model the probability distribution of promising individuals. The experimental results showed that RM-MEDA has good scalability in terms of decision variables and less sensitive to algorithmic parameter settings. In order to further improve the algorithm, Zhou et al. [22] generalized the idea of RM-MEDA and proposed a probabilistic model-based multi-objective evolutionary algorithm, named as MMEA, which is able to simultaneously approximate both the Pareto set and Pareto front of an MOP. Martí et al. [23] developed another MOEDA using growing neural gas (GNG) network - multi-objective neural EDA (MONEDA). GNG network is a self-organizing neural network based on

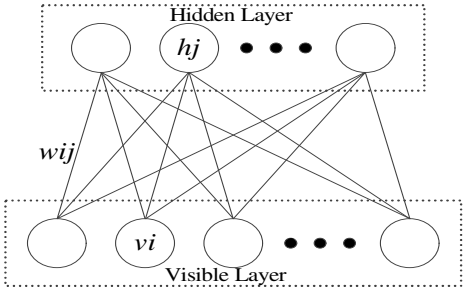

Fig. 1. Architecture of a RBM

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. WFG problems [24] were used to evaluate the search capability of the MONEDA.

In [25], an MOEDA, which uses Gaussian model as its modeling technique, was used to optimize the radio frequency identification (RFID) network design. In order to enhance the search capability of the algorithm, a particle swarm optimization (PSO) [26] algorithm was hybridized with the MOEDA. A number of children solutions are generated by the EDA while the rest of them are produced by the velocity-free PSO. The algorithm succeeded to obtain a set of tradeoff solutions in RFID network design.

3 Restricted Boltzmann Machine (RBM)

The RBM [27–29] is an energy-based binary stochastic neural network. The architecture of the network is illustrated in Fig. 1. The network consists of two layers of neurons - a visible layer and a hidden layer. The visible layer, denoted as v_i, is an input layer of the network. The hidden layer, denoted as h_j, is a latent layer that determines the capability of the network in modeling the probability distribution of the input stimuli. The network does not have the output layer. Instead, the output information is represented by the energy values of the network. w_{ij} is the weight that connecting visible unit i and hidden unit j. b_i is the bias of the visible unit i and d_j is the bias of hidden unit j. Both of the layers are fully connected to one another and the weights are symmetric. In this way, the information can flow from one layer to another, increasing the learning capability of the network. Furthermore, there is no interconnection among the neurons within the same layer. Thus, the hidden units are conditionally independent. Besides, the visible units can be updated in parallel given the hidden states. This behavior improves the training speed of the network. The weights and biases of an RBM define the energy function of the network. The energy function is presented as follows.

$$E(v,h) = -\sum_i \sum_j v_i h_j w_{ij} - \sum_i v_i b_i - \sum_j h_j d_j \tag{2}$$

Using the energy function of the network, the probability distribution of any global state can be derived as follows

$$P(v,h) = \frac{exp(-E(v,h))}{Z = \sum_{x,y} exp(-E(x,y))} \tag{3}$$

where Z is the normalizing constant, which is defined by the energy of all the global states while the numerator of the equation is the energy of a particular state. By summing all the configurations of the hidden units, the probability distribution of a visible state can be clamped to be

$$P(v) = \frac{\sum_h exp(-E(v,h))}{\sum_{x,y} exp(-E(x,y))} \tag{4}$$

3.1 Training

The training is one of the main issues in the RBM. In the literature, the contrastive divergence (CD) training method [30, 31] is the most well-known training mechanism for the RBM. In the CD training, two phases (positive phase and negative phase) are carried out. In the positive phase, the input stimuli or input data are rendered into the visible units of the network. Subsequently, the hidden states, given the visible states, are constructed by performing the Gibbs sampling as follows

$$P(h_j|v) = \varphi(\sum_i w_{ij} v_i - d_j) \tag{5}$$

where $\varphi(x) = \frac{1}{1+e^{-x}}$ is the logistic function. In the negative phase, given the hidden states, the visible states are reconstructed using the same logistic function. The process of these two phases is repeated S times. Next, the weights and biases of the network are updated as follows

$$w_{ij}^{'} = w_{ij} + \epsilon(< v_i h_j >_0 - < v_i h_j >_1) \tag{6}$$

$$b_i^{'} = b_i + \epsilon(< v_i >_0 - < v_i >_1) \tag{7}$$

$$d_j^{'} = d_j + \epsilon(< h_j >_0 - < h_j >_1) \tag{8}$$

where ϵ is the learning rate, $<>_0$ is the original states of the neurons, and $<>_1$ is the states of the neurons after a single step of reconstruction. The overall CD training is repeated until a stopping criterion is met. The process of CD training is further demonstrated in Fig. 2.

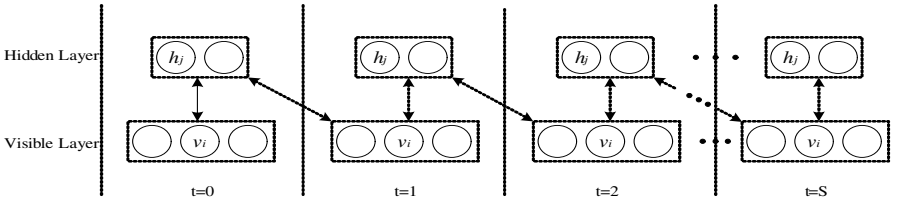

Fig. 2. The contrastive divergence (CD) training mechanism

3.2 Modeling

This paper will only study a restrcited Boltzmann machine-based estimation of distribution algorithm for multi-objective optimization (REDA). In the implementation stage, the alleles of the decision variables in the cost function are the input data that will be rendered to the visible layer of an RBM. Therefore, the RBM has n visible units if an MOP has n decision variables to be optimized. The setting of the number of hidden units is to be determined by users. The complexity of the network is directly proportional to the setting of the number of visible and hidden units. Since the probability distribution of the population needs to be constructed at every generation, it is essential for the model to be kept simple. Therefore, the number of hidden units is set to as small as possible as long as the probability model is representative.

After performing the CD training, a set of trained weights, biases, and hidden states are obtained. Subsequently, in binary representation, the joint probability distribution with n decision variables in generation g is formulated as follows.

$$P_g(v) = \prod_{i=1}^{n} p_g(v_i) \tag{9}$$

where $p_g(v_i)$ is the marginal probability of decision variable i (v_i) at generation g. The marginal probability of each decision variable is obtained through (4). Expanding the equation,

$$p_g(v_i = 1) = \frac{\sum_{l=1}^{N} \delta_l(v_i^+)}{\sum_{l=1}^{N} \delta_l(v_i^+) + \sum_{l=1}^{N} \delta_l(v_i^-)} \tag{10}$$

$$\delta_l(v_i^+) = \sum_{h=1}^{H} e^{-E(v_i=1,h)} \tag{11}$$

$$\delta_l(v_i^-) = \sum_{h=1}^{H} e^{-E(v_i=0,h)} \tag{12}$$

where $\delta_l(v_i^+)$ is the marginal cost of v_i when the cardinality of $v_i = 1$, $\delta_l(v^-)$ is the marginal cost of v_i when the cardinality cost of $v_i = 0$, N is the number of selected solutions or parent solutions, and H is the number of hidden units.

Direct sampling from the above probabilistic model reaches a limit in progress when the probability reaches a maximum value of 1.0 or a minimum value of 0.0. Therefore, the lower and upper bounds are added to the probability distribution based on the average cost of cardinality. The modified version of the marginal probability is given as below

$$p_g(v_i = 1) = \frac{\sum_{l=1}^{N} \delta_l(v_i^+) + avg(\sum_{l=1}^{N} \delta_l(v_i))}{\sum_{l=1}^{N} \delta_l(v_i^+) + \sum_{l=1}^{N} \delta_l(v_i^-) + r_i \times avg(\sum_{l=1}^{N} \delta_l(v_i))} \tag{13}$$

where $avg(\sum_{l=1}^{N} \delta_l(v_i)) = \frac{\sum_{l=1}^{N} \delta(v_i)}{N}$ and r_i is the number of different values that v_i may take. In binary case, r_i is 2.

3.3 Sampling

The children solutions are generated through the sampling of the constructed probabilistic model as follows

$$v_i = \begin{cases} 1 \text{ if random}\,(0,1) \leq p_g\,(v_i = 1) \\ 0 \qquad\quad \text{otherwise} \end{cases} \tag{14}$$

where random (0,1) is a randomly generated value between [0, 1].

3.4 Learning Capability of REDA

In this section, a detailed description of the behaviors of the RBM in the evolutionary perspective is presented. Three main issues will be covered: (1) How and what information is captured in an RBM (2) How to effectively train the RBM in the evolutionary perspective (3) What can be elucidated from the energy values of the RBM in the fitness landscape perspective.

How and What Information Is Captured in an RBM. In an RBM, the neurons between two layers are fully connected via weighted synaptic connections, and there is no intra-layer connection. These weight connections are used by the neurons to communicate their activations to one another. The quality of training of the network corresponds directly to the effectiveness at which the algorithm learns the probability distribution. Whenever the number of hidden units is sufficiently large, the network can represent any discrete distribution. This behavior that a sufficient number of hidden units in the network would guarantee improvement in the training error has been proven mathematically in [31]. During the training process, the aim is to minimize the energy equilibrium of the network such that the implicit correlations as well as the probability distribution of the input stimuli is captured and stored in the synaptic weights of the network. This distribution-based model allows the RBM to globally learn the probability distribution of the decision variables by considering the interdependencies of the data.

How to Effectively Train an RBM in the Evolutionary Perspective. 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 [31] method aims to 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, CD training will require a large number of training epochs to train the network well. When the RBM is modeled as EDA (REDA), 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 most likely due to the reduction in the size of more promising search space 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.

What Can Be Elucidated from the Energy Values of an RBM in the Fitness Landscape Perspective. In EDAs, the two main mechanisms that determine the success of the algorithms are probabilistic model construction and sampling technique. The core purpose of probabilistic modeling is to learn the probability distribution of the candidate solutions and to capture the dependencies among the decision variables. By using the linkage information of known solutions, the characteristics of the unknown solutions can be studied. In EDAs, the sampled solutions are the unknown solutions. If the characteristics of these solutions are known, this additional information can be taken into consideration during the optimization process. In an RBM, the energy-based model captures the dependencies of the decision variables by associating a scalar energy value from the network to each solution. Thus, it can be inferred that sampled solutions may have higher energy if the solutions are outside the boundary modeled by the RBM. 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. The choice of a solution outside the boundary of the modeled energy distribution may imply an increase in the exploration capability, while focusing on the solutions inside the boundary may imply an increase in the exploitation capability of the algorithm.

3.5 Algorithmic Process Flow

The general evolutionary process flow of REDA is presented in Fig. 3.

Begin
Initialization: At generation g=0, randomly generate N solutions as the initial population, $Pop(g = 0)$
Evaluation: Evaluate all solutions
Do While ("maximum generation is not reached")
1. **Fitness Assignment**: Perform Pareto ranking and crowding distance over the population
2. **Selection**: Select N promising individuals based on the binary tournament selection
3. **Training**: Train the RBM by using CD training method to obtain the weights, biases, and hidden states
4. **Probabilistic model**: Compute the probability of the joint configuration $P(v)$ by using the trained weights, biases, and hidden states of the RBM
5. **Reproduction**: Generate new set of N solutions (P) from $P(v)$
6. **Evaluation**: Calculate the fitness values of all offspring
7. **Elitism**: Select N individuals from $P \cup Pop(g)$ to form $Pop(g + 1)$. $g = g + 1$

End Do
End

Fig. 3. Pseudo-code of REDA

4 Restricted Boltzmann Machine-Based Estimation of Distribution Algorithm for Solving Multi-objective Optimization Problems

4.1 REDA with Clustering for Solving High Dimensional Problems

Many real-world optimization problems are challenged by the different characteristics and difficulties. The problems may be non-linear, restricted to several constraints, has complex relationships within the decision variables, has a large number of variables, and even consists of several conflicting objectives [32, 33]. High dimensional problems with many decision variables and conflicting objective functions to be optimized simultaneously are hard problems which may challenge the algorithm in finding the global optimal solutions. In the problems with many decision variables, the complexity of the problems increase with an increase in the number of decision variables. This is due to the enlargement of the search space and an increase in the number of possible moves towards optimality. In the problems with many conflicting objective functions, the selection pressure in selecting fitter individuals is reduced when the number of conflicting objective functions is increased. This is due to the high rate of non-dominance among individuals during the evolutionary process. This may hinder the search towards optimality or result in the population getting trapped in a local optimal. One of the ideas to overcome these issues is to exploit extra information (e.g. linkage dependencies) from within the selected population. This information is hypothesized to provide valuable guidance in driving the search process.

In order to solve MOPs with scalable number of decision variables, the algorithm presented in Fig. 3 is implemented directly. The simulation results presented in [34] indicated that REDA performs equally well or slightly inferior

to NSGA-II in ZDT and DTLZ problems with a smaller number of decision variables. This may be attributed to the fact that, REDA does not directly use location information of the selected solutions in exploiting the new solutions. Furthermore, REDA, which models the global probability of the selected individuals, is not able to effectively escape from local optima. REDA shows superior convergence performances in problems with a larger number of decision variables. Besides, REDA also able to produce more non-dominated solutions compared to NSGA-II. The increase in number of decision variables increases the size of the search space, and thus increases the complexity of the problems. The stochastic behavior of NSGA-II prevents the algorithm from evolving a good set of solutions since the number of possible combinations towards optimality is increased. The incorporation of dependency information as well as the use of the global probability distribution of solutions enables REDA to effectively explore the huge search space.

In problems with many objective functions, clustering is incorporated into REDA in order to build a probabilistic model from different regions in the search space. For sake of simplicity, the number of cluster, k, is determined by the user. In the implementation stage, a probabilistic model is built for each cluster, $L^1, L^2, ..., L^k$. Subsequently, the new population $Pop(g+1)$ is generated by sampling the probabilistic model constructed from each cluster L, where N new solutions are generated and equal number of individuals is sampled from each cluster.

In problems with three objective functions, the performance of REDA is comparable to NSGA-II. In a higher number of objective functions, REDA gives the superior performance in converging to the Pareto optimal front as well as maintaining a set of diverse solutions. The superior performance of REDA may be due to the incorporation of linkage information in driving the search. This information is learned by the network and is clamped into the probability distribution before the sampling takes place. Some flexibility is given to the algorithm in exploring the search space by allowing the training to stop before the energy reaches the minimum. The good performance of REDA in these test instances supports the claim that REDA scales well with the number of objective functions compared to NSGA-II. Clustering is important for problems with solutions that are hard to represent by a single probabilistic model.

Most of the probabilistic modeling techniques for learning the linkage dependencies of the solutions incur additional computational cost and time. In an RBM, the most time consuming part is the network training. Training is conducted at each generation and stops when the maximum number of training epochs is reached. This training process is more complicated than the genetic operators in standard MOEAs, and thus incurs additional simulation time. Even though REDA may spend more simulation time, it has a faster convergence rate compared to NSGA-II. This is one of the strengths of REDA, especially when dealing with real-world problems where the fitness evaluations are computationally expensive. Detailed information of the implementation and experimental results can be referred to [34].

Even though REDA showed promising results in solving scalable problems, it still suffers several flaws which requires further investigation. Firstly, univariate sampling in REDA may limit its ability to generate new solutions. This is because univariate sampling does not consider the correlation between the decision variables when performing the sampling. A more sophisticated sampling mechanism that is able to take into account the multivariate dependencies between the decision variables may enhance the search capability of REDA. Secondly, REDA fails to converge to the global Pareto optimal front in problems with many local optima. This is because REDA in particular and MOEDAs in general will model the probability distribution of the solutions even though they are trapped at local optima, and subsequently use the constructed probabilistic information as a reference model to produce offspring. Hybridization with local search algorithms may be one of the approaches in dealing with problems with many local optima. Thirdly, REDA is sensitive to bias. This is because the modeling in EDAs only estimates the probability distribution of the current best solutions. In other words, only global information is used. Whenever the maintained solutions are biased towards certain search regions, EDAs may consider the maintained solutions are the promising one, thus, construct their probability distribution accordingly. Therefore, it is necessary to enhance the diversity preservation of REDA especially the ability to produce a set of diverse solutions. This can be achieved by combining EDAs with other search algorithms which use location information in producing offspring, including genetic algorithms, differential evolution, particle swarm optimization algorithms, or any other algorithms with similar features.

4.2 An Energy-Based Sampling Mechanism of REDA

In our another study, the sampling mechanism of REDA was investigated [35]. In [34], the simple probability sampling is employed in REDA. This sampling technique 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 of 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, energy value can be taken into consideration. Firstly, $N \times M$ solutions are generated. Then, the energy value serves 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 levels) in terms of energy equilibrium and are generally fit. On the other hand, the solutions outside the modeled topology (higher energy levels) may be considered unstable but not unfit. 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 in order to achieve a more explorative search.

By incorporating the above-mentioned approach (energy-based sampling approach) into REDA, the simulation results in [35] indicated that the modified REDA, which models the probability distribution of the solutions by applying the energy information to detect the dependencies, is able to perform well on epistatic problems. The results also showed that the incorporation of the energy-based sampling mechanism enhances the exploration and exploitation capability of REDA. However, the limitation of REDA in escaping from local optima has not been significantly improved through this mechanism. REDA with energy-based sampling mechanism is able to escape the local optima in some simulation runs, however, is trapped in local optima in most of the runs. Therefore, it can be concluded that the REDA with energy-based sampling mechanism is sensitive to different initializations for problem with multi-modality.

4.3 REDA in Noisy Environments

In [36], we have extended the study of REDA in dealing with MOPs with noisy objective functions. In noisy environments, the presence of noise in the cost functions may affect the ability of the algorithms to drive the search process towards optimality. Beyer [37] carried out an investigation and found that the presence of noise may reduce the convergence rate, resulting in suboptimal solutions. In another study by Goh and Tan [38], it was reported that the low level of noise helps an MOEA to produce better solutions for some problems, but a higher noise level may degenerate the optimization process into a random search. Darwen and Pollack [39] concluded that re-sampling can reduce the effect of noise for a small population, but may not be as helpful for a larger population.

EDAs surpass the standard MOEAs in handling noisy information by constructing a noise handling feature in the built probabilistic model. In order to show this advantage, a likelihood correction feature was proposed in [36] in order to tune the error in the constructed probabilistic model. The previous studies showed that REDA has its limitations in exploiting the search space and may be trapped at any local optimum [34]. In order to overcome these limitations, REDA was hybridized with a particle swarm optimization (PSO) algorithm. This hybridization is expected to improve the performance since the particles may now move out of the regions modeled by REDA, and thus provide extra solutions that REDA alone was not able to tap on.

In the likelihood correction feature, the concept of probability dominance, proposed in [40], was employed. This concept is implemented to determine the

probability error of each selected individual. This probability error is then used to group the population into several clusters before a probabilistic model is built. Assume that $f_i(A)$ and $f_i(B)$ are two solutions in the objective space with m objective functions ($i = 1, ..., m$). In a noise free situation, $f_i(A)$ is said to strictly dominate $f_i(B)$ if $f_i(A)$ is smaller than $f_i(B)$ in all the objective values in minimization case. On the other hand, $f_i(A)$ and $f_i(B)$ are mutually non-dominated only if not all the objective values in one solution are lower than that of the other. In a noisy domain, the above statements may not correctly represent the true domination behavior. Even through $f_i(A)$ appears to strictly dominate $f_i(B)$, the noise may distort the actual fitness values where $f_i(B)$ is supposed to dominate $f_i(A)$. In the selection process, the selection error occurs when the less fit individual is chosen. Therefore, the probability to make an error in the selection process could be utilized to improve the decision making process.

The likelihood correction feature is based on the heuristic that if the distribution can be approximated as close to the real distribution of the best solutions, the detrimental effect of the noise can then reduced. To approximate the real distribution, the probability of making an error in the selection process is adapted in the probabilistic modeling. In binary tournament selection, if two solutions in the tournament have a huge distinction in their objective values, for example $f_i(A)$ dominates $f_i(B)$ by far, then the selection error for selecting $f_i(A)$ is small. On the other hand, if two individuals in the tournament are near to each other in the objective space, then the selection error is larger. Therefore, if the probabilistic model built by REDA is only based on individuals with small selection error, then, the model may avoid distortions caused by those solutions with a large selection error. However, the probability distribution may not come close to the real distribution if the number of individuals with smaller selection error is too little. Thus, a method to combine the distribution between solutions with small selection error and those with large selection error was designed. This combination was based on the penalty approach where individuals with smaller selection error will be penalized less while solutions with larger selection error will be more heavily penalized. This is because the real distribution is more likely to follow the distribution of the population with smaller selection error than those with larger selection error.

The simulation results in [36] demonstrated that the hybridization between REDA and PSO may slightly deteriorate the search ability of REDA in some noiseless circumstances. However, its performance was outstanding in noisy environments. The results also showed that REDA is more robust than NSGA-II because the performance of REDA is better than NSGA-II in noisy conditions. REDA, which performs the search by modeling the global probability distribution of the population, is more responsive since the reproduction is based on the global information and not individual solutions. Furthermore, likelihood correction is able to tune the probability distribution so that the distribution of the solutions is more likely to follow the one with a smaller selection error. The hybridization has further enhanced the ability of REDA in exploring the search space, especially in noisy conditions. This hybridization is utterly

important when REDA fails to model the promising regions in the search space. In that case, the hybridization could provide opportunities to explore those regions, thus, improving the search ability.

The scalability issue was also investigated. The results indicated that the hybrid REDA obtained the most promising results in the noisy test problems with different number of decision variables. This result demonstrated that the hybridization with PSO has enhanced the search ability of REDA. This is important as PSO provides a directional search which may explore the promising regions where probabilistic model may fail to explore. It can be concluded that the hybrid REDA scaled well with the number of decision variables compared to NSGA-II.

In order to study the potential of other hybridizations, REDA was hybridized with a genetic algorithm (GA) and a differential evolution (DE). The simplest and most common GA is applied, where single point crossover and bit-flip mutation are implemented. For DE, the standard recombination proposed in [41] is applied. The results indicated that all hybridizations are able to improve the performance of REDA, in most of the test problems, under both noiseless and noisy environments. Among them, hybridization with PSO gave the best results followed by DE and then GA. The function of hybridization is to provide extra search ability for REDA as REDA performs the search by using only global statistical information. This hybridization therefore enhances the ability for REDA in exploring the search space, especially in the early stage of evolutions where the search space is huge. The search using position information (GA, PSO, DE) is also essential and useful especially to explore and exploit certain promising regions. Thus, hybridization is an important mechanism to improve the search performance of EDAs.

4.4 REDA for Solving the Multi-objective Multiple Traveling Salesman Problem

In [42], REDA was implemented to solve the multi-objective multiple traveling salesman problem (MmTSP). The multiple travelling salesman problem (mTSP) is a generalization of the classical travelling salesman problem (TSP). In the mTSP, Ω salesmen are involved in a routing to visit Ψ cities ($\Omega < \Psi$) in order to achieve a common goal. In the routing, all the salesmen will start from and end at the single depot after visiting the ordered cities. Each city can only be visited once, and the total cost for all salesmen is required to be minimized. The cost can be defined as distance, time, expense, risk, etc. The complexity of the mTSP is higher than the TSP since it is required to allot a set of cities to each salesman in an optimal ordering while minimizing the total travelling cost for all salesmen. Furthermore, the mTSP is more appropriate for real life scheduling or logistic problems than the TSP because more than one salesman is usually involved. Over the past few decades, research on the TSP has attracted a great deal of attention. However, the mTSP has not received the same amount of research effort compared to the TSP.

In an MOP, no single point is an optimal solution. Instead, the optimal solution is a set of non-dominated solutions, which represents the tradeoff between the multiple objectives. In this case, fitness assignment to each solution in the evolutionary multi-objective evolutionary optimization is an important feature for the assurance of the survival of fitter and less crowded solutions to the next generation. Much research has been carried out over the past few decades to address this issue, and fitness assignment based on the domination approach is one of the most popular approaches. However, the fitness assignment in the domination approach is less efficient in solving many-objective problems. This is because the strength of the domination among the solutions in a population is weakened with the increase in the number of objective functions. This phenomenon results in poor decision making in the selection of promising solutions.

Recently, the classical method for multi-objective optimization based on decomposition has been re-formularized into a population-based approach [43, 44]. The decomposition approach decomposes an MOP into several subproblems and subsequently optimizes all the subproblems concurrently. Under this approach, it is not required to differentiate the domination behaviors of the solutions. Instead, the subproblems are constructed using any aggregation approach, and the superiority of the solutions is determined using the aggregated values.

Problem Formulation. The aim of the mTSP is to minimize the total traveling cost of all the salesmen under the condition that each city must be visited strictly once by any salesman, and all the salesmen must return to the starting depot after visiting their final ordered city. The traveling cost could be defined as the traveling distance, traveling time, traveling expense, traveling risk, etc incurred. Each salesman will have his own route and there should be no repeated visit on any city in the route of the salesman.

In the literature, the aim of the mTSP is specified to be either minimizing the total traveling cost of all salesmen or the highest traveling cost incurred by any single salesman [45]. In [42], the focus is tailored specifically for the mTSP with single depot; considering the minimization of the total traveling cost and the balancing of the workload among all salesmen. This is achieved by formularizing the objective function to be the weighted sum of the total traveling cost of all salesmen and the highest traveling cost of any single salesman. In the context of multi-objective optimization (MmTSP), more than one objective is subject to be minimized, which is formulated as follows.

Minimize:

$$F(x) = (F_1(x), ..., F_m(x))$$
$$F_1(x) = \omega_1 \times TC^1 + \omega_2 \times MC^1$$
$$\vdots$$
$$F_m(x) = \omega_1 \times TC^m + \omega_2 \times MC^m$$

where

$$TC^k(x) = \sum_{j=1}^{\Omega} IC_j^k(x)$$

$$MC^k(x) = \max_{1 \le j \le \Omega} (IC_j^k(x))$$

$$IC_j^k(x) = \sum_{i=1}^{n_j - 1} D^k(a_{i,j}, a_{i+1,j}) + D^k(a_{n_j,j}, a_{1,j})$$

In the above formulation, $x \in \phi$, ϕ is the decision space, $a_{i,j}$ is the i^{th} visiting city by salesman j, m is the number of objective functions, ω_1 and ω_2 are the weights to balance between total cost and highest cost ($\omega_1 + \omega_2 = 1.0$), TC is the total traveling cost of all salesmen, MC is the highest traveling cost of any single salesman, IC is the individual traveling cost, Ω is the number of salesmen, n_j is the number of cities traveled by salesman j, $D^k(a_{i,j}, a_{i+1,j})$ is the traveling cost (for the k^{th} objective function) between cities at locations i and $i+1$ for salesman j. In a chromosome, two conditions should be met, which are all the cities must be visited exactly once and each salesman must be assigned at least one city in his traveling route.

REDA in this section was developed in the decomposition-based framework of multi-objective optimization. Furthermore, REDA was also hybridized with the evolutionary gradient search [46] (hREDA). In the mTSP, integer number representation is used to represent the permutation of the cities. The modeling and sampling steps of REDA or hREDA is illustrated as follows.

1. Modeling. Decode the integer representation of the cities into the binary representation. Train the network. Compute the $\delta(x_i^j)$ as (11) and (12). Encode the binary representation of $\delta(x_i^j)$ into integer representation. Construct the probabilistic model $P_g(x)$ by computing the marginal probability of each city $(c_1, ..., c_\beta)$, where $\beta = \Psi + \Omega - 1$, in each permutation location as follows.

$$p_g(x) = \begin{bmatrix} p_g(x_1 = c_1) & \dots & p_g(x_\beta = c_1) \\ \vdots & \ddots & \vdots \\ p_g(x_1 = c_\beta) & \dots & p_g(x_\beta = c_\beta) \end{bmatrix}$$

$$p_g(x_i = c_j) = \frac{\sum_{l=1}^{N} \delta_l(x_i = c_j) + \frac{Z_i}{\beta \times N}}{Z_i + \frac{Z_i^2}{\beta \times N}}$$

where $p_g(x_i)$ is the probability distribution of the cities at the position x_i of the chromosomes at generation g, $p_g(x_i = c_j)$ is the probability of city j to be located at the i^{th} position of the chromosomes, c_j is the city j ($c_1 = 2 - \Omega, ..., c_\beta = \Psi$) and Z_i is the normalizing constant.

2. Sampling. Sample the constructed probabilistic model, $p_g(x)$, to generate N children solutions as follows.

$$y_i = \begin{cases} c_1 & \text{if random}\,(0,1) \leq p_g\,(x_i = c_1) \\ c_2 & \text{if } p_g(x_i = c_1) < \text{random}\,(0,1) \leq \sum_{j=1}^{2} p_g(x_i = c_j)) \\ & \vdots \\ c_\beta & \text{if } \sum_{j=1}^{\beta-1} p_g(x_i = c_j) < \text{random}(0,1) \leq \sum_{j=1}^{\beta} p_g(x_i = c_j) \end{cases}$$

where y_i is a newly generated city at i^{th} position of a chromosome.

The formulation of the MmTSP takes into account the weighted sum of total traveling cost of all salesmen and the highest traveling cost of any single salesman. The weight setting is dependent on the preference of the manager whether he wants to achieve the lowest total traveling cost of all salesmen or he wants to achieve the balancing of workload of all salesmen. If the aim is to obtain the lowest total traveling cost, the weights will be set to $\omega_1 = 1.0$, $\omega_2 = 0.0$. On the other hand, if the final objective is to balance the workload of all salesmen, the weights will then be set to $\omega_1 = 0.0$, $\omega_2 = 1.0$. However, if the aim is to achieve tradeoff between the two aims, then different weight settings should be employed.

The simulation results reported in [42] indicated that the hybrid REDA with decomposition (hREDA) is able to produce a set of diverse solutions but it is slightly inferior in terms of proximity to MOEA/D [44] in problems with small number of cities. In problems with a large number of cities, the simulation results showed that the decomposition algorithms (hREDA, REDA, and MOEA/D) achieve better Pareto front than the domination algorithms (NSGA-II). For the decomposition algorithms, hREDA generates a better set of diverse solutions than REDA and MOEA/D. However, the solutions generated by REDA have a better proximity than hREDA. Shim et al. [42] concluded that the decomposition algorithms scale well with the increase in the number of decision variables compared to the algorithms using the concept of domination. REDA uses global distribution of the parent solutions to guide the search process. The results showed that REDA have good proximity results, but poor solution diversity. Introducing local information into the evolutionary process, which helps the algorithm to further explore and exploit the search space, rectifies this limitation of REDA.

The findings also revealed that the total traveling cost increases with the increase in the number of salesmen. This is because when more salesmen are involved, the task gets more difficult since the algorithms need to determine the route for each salesman while maintaining the minimum total traveling cost at the same time. Since all salesmen need to return to the home city and the final assigned city could be far from the depot, additional traveling cost may be incurred. For hREDA, the gradient information weakens with the increase in the number of salesmen, resulting in the algorithm not being able to exploit the information as effectively. However, its performance was the best compared to the other algorithms.

Overall, the performances of algorithms using the decomposition framework (hREDA, REDA and MOEA/D) were superior to those of the algorithms based on the concept of domination (NSGA-II) in most of the problem settings. The superiority of the decomposition algorithms is attributed to the aggregation principle used for fitness assignment. The tournament could be carried out by simply comparing the aggregated fitness values of solutions. Solutions with higher fitness values will always be selected to survive and reproduce. On the other hand, the concept of domination requires that fitness be assigned to each solution based on their rank of domination. In many-objective problems, most of the solutions are non-dominated and are given lower ranks. This may prevent the tournament process from selecting promising solutions to survive. Thus, NSGA-II performed poorly compared to the decomposition algorithms.

4.5 Hybrid Adaptive MOEAs for Solving Continuous MOPs

In our recent work [47], we introduced two versions of hybrid adaptive MOEAs for solving continuous MOPs. The motivation of this study is as follows. Many multi-objective evolutionary algorithms (MOEAs) have been designed to solve MOPs. For example, MOEAs that use genetic algorithms (GAs) as the search technique are NSGA-II [15] and MOEA/D [44], among others. MOEAs that use differential evolutions (DE) as the search technique are Pareto differential evolution (PDE) [48], generalized differential evolution3 (GDE3) [49], and MOEA/D with DE [50], among others. Next, MOEAs that use estimation of distribution algorithms (EDAs) as the search approach are as discussed in Section II. Each of the above-mentioned algorithms is efficient in solving certain MOPs and has their own strengths and weaknesses. Furthermore, no evidence indicates that one of the EAs is superior to the others. Thus, it is possible that the synergy among the EAs can complement their weaknesses while maintaining their strengths.

In [47], an adaptive feature, which determines the proportion of the number of solutions to be produced by each EA in a generation, was proposed. Initially, each EA is given an equal chance to produce the initial solutions. After the reproduction processes, a number of promising solutions are selected and stored in an archive. Then, the proportion of the number of solutions to be generated by each optimizer in the next generation is calculated according to the proposed adaptive mechanism as illustrated in Fig. 4. Let ψ as the solutions in an archive. First, calculate the number of solutions in ψ that are generated by each EA. Afterward, the adaptive proportion rate ($Ar_g^{EA_i}$) at generation g for each EA is calculated according to Step 2. A learning rate ($\epsilon < 0$) is incorporated to the updating rule in Step 2 in order to moderate the influences of the proportion of the number of selected solutions in generation g to the whole evolutionary processes. This is because the optimizers that are able to generate a more number of promising solutions in the current generation may not be the superior optimizers in the next generation. In Step 3, a lower bound is set to the adaptive proportion rate. This is necessary since an optimizer may dominate other EAs and finally the adaptive proportion rate of this optimizer will become 1.0 while the adaptive proportion rate of other EAs will become 0.0. When this happens, all children solutions will only be generated by

%%Given a set of selected solutions that are stored in an archive (ψ)

1. Calculate the number of solutions in ψ that are generated by each EA, denoted as $D_g^{EA_i}$ where $i = 1, \ldots, M$, M is the number of EAs that are involved in the hybridization. In this paper, three EAs are involved. Thus, the number of solutions in ψ that are generated by each EA are denoted as $D_g^{EA\_1} = D_g^{GA}$, $D_g^{EA\_2} = D_g^{DE}$ and $D_g^{EA\_3} = D_g^{EDA}$.
2. Calculate the adaptive proportion rate for each EA as follows.
   For $\boldsymbol{i = 1 : M}$
   $Ar_g^{EA\_i} = Ar_{g-1}^{EA\_i} + \epsilon \times Pr_g^{EA\_i}$, where $Pr_g^{EA\_i} = D_g^{EA\_i}/N$
   End For
   where $Ar_g^{EA\_i}$ is the adaptive proportion rate at generation g for i^{th} EA, ϵ is the learning rate, $Pr_g^{EA\_i}$ is the current proportion rate and N is the archive size or the number of solutions in an archive.
3. Check for the lower bound of the adaptive proportion rate
   For $\boldsymbol{i = 1 : M}$
   $If\ Ar_g^{EA\_i} < l\_bound$
   $Ar_g^{EA\_i} = l\_bound$
   End For
4. Normalize the adaptive proportion rate so that the sum of the adaptive proportion rates is equal to 1.0
   For $\boldsymbol{i = 1 : M}$
   $Ar_g^{EA\_i} = Ar_g^{EA\_i} / (\sum_{i=1}^{M} Ar_g^{EA\_i})$
   End For

Fig. 4. Pseudo-code of the adaptive mechanism

this optimizer till the end of the evolutionary processes. Thus, it is necessary to set a lower bound to the adaptive proportion rate to guarantee that the problem would not exist. Since the summation of all the adaptive proportion rates should be equal to 1.0, the final adaptive proportion rates should be normalized especially when Step 3 is applied (Step 4). Afterward, a typical evolutionary process is continued. Through this hybridization, the hybrid algorithms showed the best results in most of the MOPs.

5 Conclusion

This paper presented our recent works on synergy between PMGs and MOEAs. More specifically, a literature review focused on the existing work on MOEDAs has been outlined. Next, the possibility of constructing an EDA based on RBM has also been demonstrated. Next, five studies that implement REDA for solving different MOPs were given. The case studies are:(1) REDA with clustering in solving scalable MOPs (2) sampling study of REDA and its implementation in solving epistatic MOPs (3) synergy between REDA and PSO in tackling noisy MOPs (4) hybrid REDA with the EGS in evolving a set of permutation of cities in MmTSP problems and (5) hybrid adaptive MOEAs in solving various types of MOPs. The simulation results indicated that REDA is weak in addressing multimodality problems, inferior to NSGA-II in exploiting the near optimal solutions, and mediocre in generating a set of diverse solutions. The positive aspects are that REDA can perform well in high dimensional problems and many objective

problems, REDA is more robust than NSGA-II in noisy MOPs, and REDA has a faster convergence rate than NSGA-II even though it takes a higher computational time. The performance of REDA was improved through the hybridization with local search algorithms and the synergy with other evolutionary algorithms.

References

1. Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
2. Tan, K.C., Khor, E.F., Lee, L.H.: Multiobjective Evolutionary Algorithms and Applications. Springer (2005)
3. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer, Norwell (2001)
4. Lozano, J.A., Larrañaga, P., Bengoetxea, E.: Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms. STUDFUZZ. Springer, New York (2006)
5. Neri, F., Mininno, E.: Memetic Compact Differential Evolution for Cartesian Robot Control. IEEE Computational Intelligence Magazine 5(2), 54–65 (2010)
6. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
7. Larrañaga, P.: Probabilistic Graphical Models and Evolutionary Computation. In: Plenary Lecturer in 2010 IEEE World Congress on Computational Intelligence (2010)
8. Ting, C.K., Zeng, W.M., Lin, T.C.: Linkage Discovery Through Data Mining. IEEE Computational Intelligence Magazine 5(1), 10–13 (2010)
9. Pelikan, M., Sastry, K., Goldberg, D.E.: Multiobjective Estimation of Distribution Algorithm. In: Scalable Optimization via Probabilistic Modeling, pp. 223–248. Springer (2006)
10. Bosman, P.A.N., Thierens, D.: Multi-objective Optimization with Diversity Preserving Mixture-based Iterated Density Estimation Evolutionary Algorithms. International Journal of Approximate Reasoning 31(3), 259–289 (2002)
11. Laumanns, M., Ocenasek, J.: Bayesian Optimization Algorithms for Multi-objective Optimization. In: Proceedings of the 7th International Conference on Parallel Problem Solving from Nature, pp. 298–307 (2002)
12. Costa, M., Minisci, E.: MOPED: A Multi-objective Parzen-Based Estimation of Distribution Algorithm for Continuous Problems. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 282–294. Springer, Heidelberg (2003)
13. Li, H., Zhang, Q., Tsang, E., Ford, J.A.: Hybrid Estimation of Distribution Algorithm for Multiobjective Knapsack Problem. In: Gottlieb, J., Raidl, G.R. (eds.) EvoCOP 2004. LNCS, vol. 3004, pp. 145–154. Springer, Heidelberg (2004)
14. Okabe, T., Jin, Y., Sendhoff, B., Olhofer, M.: Voronoi-based Estimation of Distribution Algorithm for Multi-objective Optimization. In: IEEE Congress on Evolutionary Computation, pp. 1594–1601 (2004)
15. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
16. Pelikan, M., Sastry, K., Goldberg, D.E.: Multiobjective hBOA, Clustering, and Scalability. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 663–670 (2005)

17. Sastry, K., Goldberg, D.E., Pelikan, M.: Limits of Scalability of Multiobjective Estimation of Distribution Algorithms. In: IEEE Congress on Evolutionary Computation, pp. 2217–2224 (2005)
18. Soh, H., Kirley, M.: moPGA: Towards a New Generation of Multi-objective Genetic Algorithms. In: IEEE Congress on Evolutionary Computation, pp. 1702–1709 (2006)
19. Bosman, P.A.N., Thierens, D.: Adaptive Variance Scaling in Continuous Multi-objective Estimation of Distribution Algorithms. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 500–507 (2007)
20. Zhong, X., Li, W.: A Decision-tree-based Multi-objective Estimation of Distribution Algorithm. In: Proceedings of the 2007 International Conference on Computational Intelligence and Security, pp. 114–118 (2007)
21. Zhang, Q., Zhou, Z., Jin, Y.: RM-MEDA: A Regularity Model-based Multiobjective Estimation of Distribution Algorithm. IEEE Transactions on Evolutionary Computation 12(1), 41–63 (2008)
22. Zhou, A., Zhang, Q., Jin, Y.: Approximating the Set of Pareto-optimal Solutions in Both the Decision and Objective Spaces by an Estimation of Distribution Algorithm. IEEE Transactions on Evolutionary Computation 13(5), 1167–1189 (2009)
23. Martí, L., García, J., Berlanga, A., Molina, J.: Solving Complex High-dimensional Problems with the Multi-objective Neural Estimation of Distribution Algorithm. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 619–626 (2009)
24. Huband, S., Hingston, P., Barone, L., While, L.: A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. IEEE Transactions on Evolutionary Computation 10(5), 477–506 (2006)
25. Gao, Y., Hu, X., Liu, H., Feng, Y.: Multiobjective Estimation of Distribution Algorithm Combined with PSO for RFID Network Optimization. In: International Conference on Measuring Technology and Mechatronics Automation, pp. 736–739 (2010)
26. Li, X.: Niching Without Niching Parameters: Particle Swarm Optimization Using a Ring Topology. IEEE Transactions on Evolutionary Computation 14(1), 150–169 (2010)
27. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted Boltzmann Machines for Collaborative Filtering. In: Proceedings of the 24th International Conference on Machine Learning, pp. 791–798 (2007)
28. Hinton, G.E.: Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation 14(8), 1771–1800 (2002)
29. Hinton, G.E., Salakhutdinov, R.: Reducing the Dimensionality of Data with Neural Networks. Science 313(5786), 504–507 (2006)
30. Tieleman, T.: Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1064–1071 (2008)
31. Carreira-Perpiñán, M., Hinton, G.: On Contrastive Divergence Learning. In: 10th International Workshop on Artificial Intelligence and Statistics, pp. 59–66 (2005)
32. Lu, K., Yen, G.G.: Rank-density-based multiobjective genetic algorithm and benchmark test function study. IEEE Transactions on Evolutionary Computation 7(4), 325–343 (2003)
33. Deb, K., Sinha, A., Kukkonen, S.: Multi-objective Test Problems, Linkages, and Evolutionary Methodologies. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 1141–1148 (2006)

34. Tang, H., Shim, V.A., Tan, K.C., Chia, J.Y.: Restricted Boltzmann Machine Based Algorithm for Multi-objective Optimization. In: IEEE Congress on Evolutionary Computation, pp. 3958–3965 (2010)
35. Shim, V.A., Tan, K.C., Chia, J.Y.: An Investigation on Sampling Technique for Multi-objective Restricted Boltzmann Machine. In: IEEE Congress on Evolutionary Computation, pp. 1081–1088 (2010)
36. Shim, V.A., Tan, K.C., Chia, J.Y., Al Mamun, A.: Multi-objective Optimization with Estimation of Distribution Algorithm in a Noisy Environment. Evolutionary Computation (accepted)
37. Beyer, H.G.: Evolutionary Algorithms in Noisy Environments: Theoretical Issues and Guidelines for Practice. Computer Methods in Applied Mechanics and Engineering 186(2), 239–267 (2000)
38. Goh, C.K., Tan, K.C.: An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization. IEEE Transactions on Evolutionary Computation 11(3), 354–381
39. Darwen, P., Pollack, J.: Coevolutionary learning on Noisy Tasks. In: IEEE Congress on Evolutionary Computation, pp. 1724–1731 (1999)
40. Hughes, E.J.: Evolutionary Multi-objective Ranking with Uncertainy and Noise. In: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, pp. 329–343 (2001)
41. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. Journal of Global Optimization 11(4), 341–359 (1997)
42. Shim, V.A., Tan, K.C., Cheong, C.Y.: A Hybrid Estimation of Distribution Algorithm with Decomposition for Solving the Multi-objective Multiple Traveling Salesmen Problem. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews (accepted)
43. Ishibuchi, H., Yoshida, T., Murata, T.: A Multi-objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 28(3), 204–223 (1998)
44. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutinoary Algorithm Based On Decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007)
45. Okonjo, C.: An Effective Method of Balancing the Workload Amongst Salesmen. Omega 16(2), 159–163 (1998)
46. Goh, C.K., Ong, Y.S., Tan, K.C., Teoh, E.J.: An Investigation on Evolutionary Gradient Search for Multi-objective Optimization. In: IEEE Congress on Evolutionary Computation, pp. 3741–3746 (2008)
47. Shim, V.A., Tan, K.C.: A Hybrid Adaptive Evolutionary Algorithm in the Domination-based and Decomposition-based Frameworks of Multi-objective Optimization. In: IEEE Congress on Evolutionary Computation (accepted, 2012)
48. Abbass, H.A., Sarker, R.: The Pareto Differential Evolution Algorithm. International Journal of Artificial Intelligence Tools 11(4), 531–552 (2002)
49. Kukkonen, S., Lampinen, J.: GDE3: The Third Evolution Step of Generalized Differential Evolution. In: IEEE Congress on Evolutionary Computation, pp. 443–450 (2005)
50. Li, H., Zhang, Q.: Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 13(2), 284–302 (2009)