*International Journal of Innovative Computing, Information and Control*
Volume **9**, Number **6**, June **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 2505–2519

# AN IMPROVED REAL-CODED BAYESIAN OPTIMIZATION ALGORITHM FOR CONTINUOUS GLOBAL OPTIMIZATION

Behnaz Moradabadi[1], Hamid Beigy[1] and Chang Wook Ahn[2,*]

[1]Department of Computer Engineering
Sharif University of Technology
Azadi Ave., Tehran, Iran
moradabadi@ce.sharif.edu; beigy@sharif.edu

[2]Department of Computer Engineering
Sungkyunkwan University
2066 Seobu-ro, Suwon 440-746, Korea
*Corresponding author: cwan@skku.edu

Received March 2012; revised October 2012

Abstract. *Bayesian optimization algorithm (BOA) utilizes a Bayesian network to estimate the probability distribution of candidate solutions and creates the next generation by sampling the constructed Bayesian network. This paper proposes an improved real-coded BOA (IrBOA) for continuous global optimization. In order to create a set of Bayesian networks, the candidate solutions are partitioned by an adaptive clustering method. Each Bayesian network has its own structure and parameters, and the next generation is produced from this set of networks. The adaptive clustering method automatically determines the correct number of clusters so that the probabilistic building-block crossover (PBBC) is effectively preserved. This leads to a better search when the diversity of population is high at the beginning of search. Moreover, it tunes the solutions by automatically decreasing the number of clusters as the diversity of population decreases during the search process. The experimental results demonstrate that the proposed algorithm achieves better performance on well-known benchmark functions in the continuous global optimization.*
**Keywords:** Evolutionary algorithms, Bayesian optimization algorithm, Adaptive clustering method, Probabilistic building-block crossover, Continuous global optimization

1. **Introduction.** Evolutionary algorithms (EAs) have been proposed as an efficient optimization technique, which were used to optimize general, constrained, multi-objective, real-valued optimization problems [1, 2, 3, 4, 5]. Genetic Algorithms (GAs) are stochastic search methods inspired by natural evolution and genetics [1, 2]. Once a set of promising solutions has been selected, new candidate solutions are sampled by applying recombination (i.e., crossover) and mutation to the promising solutions. Public, fixed, problem-independent mutation and recombination operators frequently lead to missing building blocks (BBs) (i.e., the knowledge of the relationship between variables), thereby resulting in converging to a local optimum [2]. Recently, *estimation of distribution algorithms* (EDAs) have been developed to resolve the disruption of BBs [6]. Instead of using crossover and mutation operators, they exploit probabilistic models to model the distribution of candidate solutions. This results in automatically decomposing the problem into several subproblems, thereby leading to better performance on a wide variety of problems. In particular, *Bayesian optimization algorithm* (BOA) employs Bayesian networks to model the distribution of candidate solutions [7]. In addition, real-coded Bayesian optimization algorithm (rBOA) makes use of the power of (discrete) BOA to

solve real-valued (i.e., continuous) optimization problems [8]. In rBOA, a Bayesian factorization is performed on the current population, the result of factorization that contains linkage information is utilized to identify subproblems, and new individuals are sampled from the independent subproblems.

This paper presents an improved rBOA (IrBOA) in order to more efficiently handle the continuous global optimization. To this end, we employ an adaptive clustering method, the $k'$-means algorithm [9], which partitions the problem space in terms of multiple Bayesian networks when the diversity of population is high. Moreover, we take advantage of *probabilistic building-block crossover* (PBBC) to improve the quality of solutions when the diversity of population decreases. The PBBC is a kind of population-wise building-block crossover that utilizes a probability distribution calculated from the results of suitable problem decomposition. Note that in the proposed IrBOA, each Bayesian network has its own structure and parameters. The use of different structures (based on the adaptive clustering method) leads to traversing the problem space efficiently from the beginning of search and improving the quality of the obtained solutions by promoting the PBBC in later generations. Realizing the PBBC results from the adaptive clustering method because the number of clusters decreases as the diversity of solutions is reduced. When a smaller number of Bayesian networks is constructed, more BBs tend to be aggregated in a Bayesian network. Therefore, the IrBOA can investigate the problem space in terms of a various dependency structures as well as realize the PBBC in an effective manner. As an adaptive clustering method, the $k'$-means algorithm can perform clustering on the input data without pre-assigning the correct number of clusters. It tries to minimize a cost function that is defined as the sum of mean-square-error and information uncertainty. When the cost function reaches the global minimum, the algorithm converges at the true cluster centers [9]. Finally, the next generation is produced from the Bayesian networks created at the previous generation. The experimental results show that the proposed algorithm is superior to several recent real-coded EDAs on both uni- and multi-modal optimization functions.

The rest of this paper is organized as follows. After briefly reviewing EDAs in the next section, the proposed algorithm is presented in Section 3. Section 4 shows the experimental results. Section 5 concludes the paper with a summary.

## 2. **Estimation of Distribution Algorithms.**

Traditional EAs randomly generate an initial population (i.e., a set of solutions) [1, 2]. At first, the promising solutions are selected as the parents for the next generation. Recombination exchanges the partial parts of the parents and mutation acts small agitation to the resulting solutions. New solutions replace some of the old ones and this process is repeated until the stopping criteria are satisfied. However, these conventional variation operators often destroy BBs discovered so far and thus often fail to blend them effectively. Meanwhile, EDAs have been developed for handling this BB disruption problem [6]. EDAs model the distribution of candidate solutions; but the correct probability distribution is unknown, and thus these algorithms try to estimate the true distribution [6, 10]. In EDAs, a part of promising solutions are selected from the first population. Then, a model to estimate the probability distribution of the promising solutions is created and the new generation is produced from this distribution. The new solutions are then merged with the primary population, replacing some of the old ones. This process is repeated until some termination criteria are met. Many EDAs have been suggested for solving optimization problems in discrete and continuous domains. For example, *extended compact genetic algorithm* (ECGA) [11], *factorized distribution algorithm* (FDA) [12], BOA [7, 13] in discrete-valued variables and *estimation of Gaussian networks algorithm* (EGNA) [14], *mixed Bayesian optimization algorithm*

(MBOA) [15], mIDEA [16], rBOA [8] in real-valued variables have been proposed. Some real-coded (i.e., continuous) EDAs are briefly reviewed below.

In MBOA, the structure of Bayesian network is formed by a set of decision trees with univariate normal-kernel leaves [15]. The MBOA has one decision tree for each variable of the optimization problem. The split nodes of the decision tree are defined to linearly split the domain of parent variables into segments. In EGNA, a Gaussian network is produced at each generation in terms of a scoring metric and new population is sampled from the learned network [14]. Since the EGNA only uses a single-peak Gaussian model, it is not suitable to solve complex problems. The mIDEA uses *Bayesian information criterion* (BIC) (i.e., a scoring metric) to evaluate a probabilistic model and generate a probabilistic model with a mixture of normal distributions to fit the model [16]. Moreover, the algorithm performs clustering on the entire problem space and operates in each cluster separately. Thus, the mIDEAs cannot realize the PBBC unless one cluster contains most BBs.

In rBOA, the Bayesian network has a single structure with a mixture of normal distributions [8]. After selecting the promising solutions from the current population, a Bayesian network is built to model the distribution of the promising solutions. In the sampling phase, subtrees (i.e., BBs or subproblems) are extracted from the structure of the Bayesian network. For each subtree space, the clustering is performed on whole solutions and a mixture of normal distributions is fitted in each cluster. New individuals are created by assembling several partial solutions sampled from their corresponding mixture distributions. Finally, the new individuals are merged into the main population by a replacement method. The above procedures are repeated until some termination criteria are satisfied. Note that the rBOA can effectively actualize the PBBC since the clustering is performed at the level of subproblems. However, the rBOA employs a single Bayesian network; only one structure is used to produce the next generation [8]. It denotes that the problem space may not be traversed effectively.

Recently, some EDAs based on Bayesian networks for the continuous optimization have been proposed [17, 18, 19]. In *Bayesian networks and Gaussian mixture model* (BNGMM) [19], for instance, a Bayesian network is used to model the dependencies between variables, and thus the problem is decomposed into multiple subgraphs that represent subproblems. After calculating a GMM to estimate the probability distribution of each subproblem, the new generation is created from the subproblem-based GMM [19].

## 3. The Improved Real-Coded Bayesian Optimization Algorithm.

In this section, an *improved real-coded Bayesian optimization algorithm* (IrBOA) is presented. The IrBOA uses an adaptive clustering method to categorize the promising solutions into several clusters. As the adaptive clustering method, the $k'$-means algorithm performs clustering without pre-assigning the correct number of clusters [9]. This is achieved by minimizing a scoring function that is defined as the sum of mean-square-error and information uncertainty [9]. When this function reaches the global minimum, the true number of clusters is obtained and then a Bayesian network is constructed on each cluster. Each Bayesian network has its own structure and parameters and the next generation is sampled from these Bayesian networks. The number of individuals sampled from a Bayesian network corresponding to a cluster, depends on the number of individuals grouped in that cluster. After merging the new solutions with the old ones, the next population is chosen in accordance with their fitness. Figure 1 illustrates the overall procedures of IrBOA. At the early generations (when the diversity of population is high), the proposed algorithm can perform an effective search on the problem space in order to discover a variety of dependencies since the $k'$-mean algorithm leads to an efficient number of clusters. In later

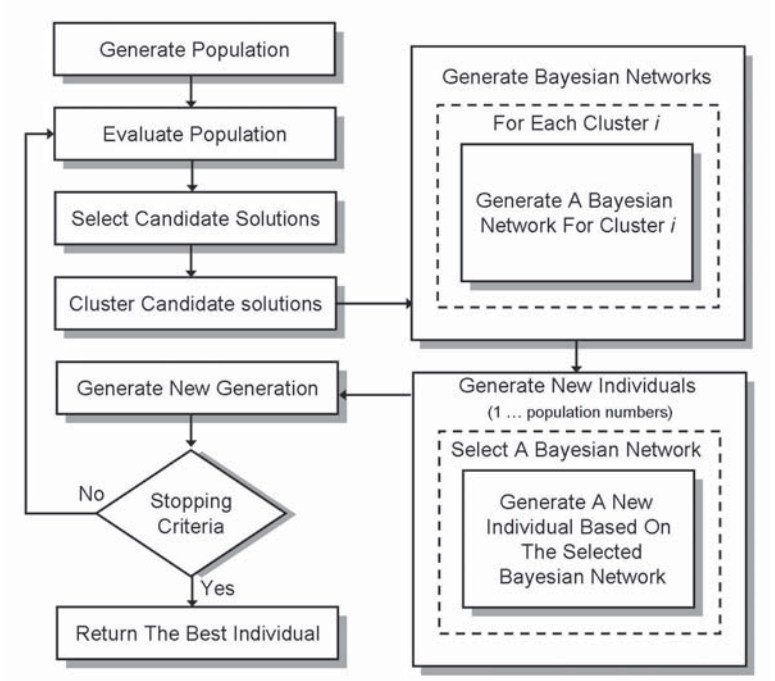



FIGURE 1. Overall procedures of IrBOA

generations (when the diversity of population is reduced), the number of clusters discovered by the $k'$-means algorithm also decreases. Thus, the PBBC can get stronger and more BBs can be brought together in a cluster; this results in a fine tuning of the solutions. The pseudo-code is shown in Algorithm 1. The IrBOA consists of three main steps: (1) Clustering, (2) Model building, and (3) Model sampling and replacement. These steps are described in the following subsections.

**Algorithm 1** Pseudo-code of IrBOA

Let $l$ be the problem dimension and $Np$ be the population size;
Let $k$ be the generation counter and $NFC$ be the allowable number of generations.
Create a random population $P$ of $Np$ individuals.
Evaluate the population $P$.
Set $k = 0$.
**while** $k < NFC$ **do**
  Select $\tau\%$ individuals $S$ from the population $P$ based on their fitness.
  Perform clustering on the selected solutions using the $k'$-means algorithm.
  Build a probabilistic model structure for each cluster.
  **for** $i = 1 \rightarrow Np$ **do**
    Select a cluster $j$ based on the coefficient score $\nu_{(j)}$.
    Generate a new individual from the joint pdf $f_{M(\zeta_k,\Theta_k)}(Y)$ on the basis of the subproblems corresponding to the selected Bayesian network.
  **end for**
  Merge the new individuals with the old ones.
  Evaluate the population $P$.
  Select the best $Np$ individuals as the new generation.
  $k \leftarrow k + 1$
**end while**

3.1. **Clustering.** The goal of this step is to partition the selected/promising solutions into a set of clusters. The IrBOA can investigate the problem space for a variety of dependency structures as well as realize the PBBC in an effective manner. For partitioning the promising solutions, the IrBOA utilizes the $k'$-means algorithm that is an extended version of the $k$-means algorithm [9]. There is no need to determine the number of clusters in advance; the correct number of clusters is automatically discovered during the operation [17]. To this end, the density of clusters is used to determine the correct number of clusters. If the $i$th cluster, $C_i$, contains $n_i$ solutions (out of $N$ individuals), then the amount of information of $C_i$, denoted by $I(C_i)$, is given as

$$I(C_i) = |\log(n_i/N)| \tag{1}$$

This information is used to approximate uncertainty about the cluster $C_i$. The $k'$-means algorithm consists of two phases. In the first phase, the traditional $k$-means algorithm is used to determine the initial cluster centers. The initial number of clusters should be sufficiently large [9]. In the second phase, a solution $x_t$ is randomly picked for each candidate solution $x_t$ and all clusters, and the cluster membership function $I(x_t, i)$ is calculated by

$$I(x_t, i) = \begin{cases} 1 & \text{if } i = \arg\min_j \{dm(x_t, C_j)\}, \text{ where } \ j = 1, \ldots, k' \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $k'$ is the current number of clusters and $dm(x_t, C_j)$ is

$$dm(x_t, C_j) = \|x_t - c_j\|^2 - E \log(I(C_j)),$$

and the parameter $E$ is equal to

$$E \in [a, 3a], \tag{3}$$

where $a$ is a constant [9]. Every candidate solution is grouped into the cluster whose centroid is closest to that solution according to the cluster membership $I(x_t, i)$. Thus, the center of the cluster $c_j$ is calculated as the average of its members, as follows:

$$c_j = \frac{1}{|C_j|} \sum_{x_t \in C_j} x_t, \quad \text{for } i = 1, 2, \ldots, k'. \tag{4}$$

The second phase is repeated until all the cluster centers remain [9].

After the clustering on the candidate solutions, a score is assigned to each cluster. This score depends on the ratio of the number of candidate solutions grouped into that cluster to the whole of candidate solutions [20]. The score is used as the probability of selecting a cluster in order to create new individuals sampled from the Bayesian network constructed on that cluster. If the candidate solutions are grouped in $k'$ clusters and the cluster $j$ has $n_j$ candidate solutions, then the score of the cluster $j$, $\nu_{(j)}$, is computed by

$$\nu_{(j)} = \frac{n_j}{\sum_{i=1}^{k'} n_i}. \tag{5}$$

3.2. **Model building.** After partitioning the promising solutions, the probabilistic models must be built. The goal of this step is to build a model structure $\zeta$ that better fits the current solutions grouped into a cluster. To find a good structure, there are two components: *scoring metric* and *search procedure* [21, 22]. The former measures how well the considering network models the candidate solutions; the latter explores the space of all possible networks to discover the best one [21, 22].

3.2.1. *Scoring metric.* The IrBOA uses the Bayesian networks in order to probabilistically model the promising individuals. Each Bayesian network has a single structure $M(\zeta)$ and a set of parameters $\Theta$. To construct a Bayesian network, a variety of metrics have been proposed. For instance, the minimum description length metric uses a hypothesis that the number of regularities in the data encoded by a model is somehow proportional to the amount of compression of the data allowed by the model [13, 23]. The Bayesian metric measures the quality of each structure by computing its marginal likelihood with respect to the given data. Especially, the *Bayesian information criterion* (BIC) is widely used due to its efficiency and simplicity [8, 24].

In this study, the evaluation of model structures is performed by the BIC principle [8, 24]. The BIC score consists of the model accuracy regarding the $k$th cluster and its complexity. The score of the model constructed on the $k$th cluster, denoted by $f_{M(\zeta_k,\Theta_k)}$, is computed by

$$f_{M(\zeta_k,\Theta_k)} = Accuracy(M(\zeta_k,\Theta_k)) + Complexity(M(\zeta_k,\Theta_k)), \tag{6}$$

where $Accuracy(M(\zeta_k,\Theta_k))$ represents the accuracy of a Bayesian network corresponding to the $k$th cluster. The accuracy is calculated by the sum of conditional probabilities of the solutions grouped into that cluster, as follows:

$$Accuracy(M(\zeta_k,\Theta_k)) = -\ln\left(\prod_{j=1}^{|S_k|} F_{M(\zeta_k,\Theta_k)}(Y^j)\right), \tag{7}$$

where $S_k$ is the set of promising solutions grouped into the $k$th cluster and $F_{(\zeta_k,\Theta_k)}(Y)$ denotes the conditional probability of solution $Y$. In addition, we have

$$F_{M(\zeta_k,\Theta_k)}(Y) = \prod_{i=0}^{l-1} f_{\Theta_k{}^i}(Y_i|\Pi_i), \tag{8}$$

where $Y = (Y_1, Y_2, \ldots, Y_l)$ is a vector of variables, $\Pi_i$ is a set of parents for $Y_i$, and $\Theta_k$ are the parameters of normal distribution, which describe the distribution of variables for the solutions belonging to the $k$th cluster. Moreover, the model complexity of the $k$th Bayesian network is computed by

$$Complexity(M(\zeta_k,\Theta_k)) = \lambda \ln(|S_k|)|\Theta_k|, \tag{9}$$

where $\lambda$ is the rate of complexity effect on the BIC score [8]. Thus, the final BIC score can be rewritten as

$$BIC(f_{M(\zeta_k,\Theta_k)}(Y), S) = -\ln\left(\prod_{j=1}^{|S_k|} f_{M(\zeta_k,\Theta_k)}(Y^j)\right) + \lambda \ln(|S_k|)|\Theta_k|. \tag{10}$$

The objective is to find a Bayesian network for each cluster that minimizes the BIC score of that cluster. Since the minimal negative log-likelihood is equivalent to the minimal entropy, the BIC score can be calculated by

$$BIC(f_{M(\zeta_k,\Theta_k)}(Y), S_k) = |S_k| H(f_{M(\zeta_k,\Theta_k)}(Y)) + \lambda \ln(|S_k|)|\Theta_k|, \tag{11}$$

where $H(f_{M(\zeta_k,\Theta_k)}(Y))$ represents the differential entropy of $f_{M(\zeta_k,\Theta_k)}(Y)$.

3.2.2. *Search procedure.* Learning the network structure with a given scoring metric is a hard combinatorial problem; in fact, the task of finding the best network with respect to most Bayesian and non-Bayesian metrics becomes a class of NP-complete problems [25]. Thus, there is no polynomial-time algorithm that finds the optimal structure of a Bayesian network with regard to most scoring metrics. However, a simple greedy algorithm often

performs well in a number of difficult machine learning tasks [3, 13, 23]. In this sense, the IrBOA employes a greedy-search heuristic in order to search for an optimal or near-optimal structure for the promising solutions. The search process starts with an empty structure and at each generation, adds an edge that leads to the greatest improvement in the BIC score corresponding to the considering cluster. This search is stopped when any edge that improves the scoring metric is not found.

3.3. **Model sampling and replacement method.** After constructing the probabilistic models, new individuals should be sampled from these models. The IrBOA utilizes the score of each cluster, $(\nu_j)$, as a probability of choosing the corresponding Bayesian network to create new solutions for the next generation. Then, all trees are extracted from the chosen Bayesian network. These trees encode subproblems for the selected structure. To create partial solutions corresponding to the subproblems, it employs the normal *probability distribution function* (pdf) due to its inherent advantages. In other words, new solutions are generated from the univariate conditional normal pdfs of the candidate solutions that are grouped into the selected cluster, as follows [8, 17]:

$$f(Y_0|Y_1, Y_2, \ldots, Y_n) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_0-\mu)^2}{2\sigma^2}}, \tag{12}$$

where $Y_0$ is a variable to be generated, $(Y_1, Y_2, \ldots, Y_n)$ are the parents of $Y_0$, and

$$\sigma = \frac{1}{\sqrt{(\sum^{-1})_{0,0}}}, \tag{13}$$

$$\mu = \mu_0 - \frac{\sum_{j=1}^{Y_n} (Y_j - \mu_j)((\sum^{-1})_{j,0})}{(\sum^{-1})_{0,0}}, \tag{14}$$

where $\mu_j$ is the mean of parent $j$, $\sum^{-1}$ is the inverse of covariance matrix for $(Y_0|Y_1, Y_2, \ldots, Y_n)$, and $\mu_0$ is the mean of candidate solutions grouped into the selected cluster [8, 17]. After generating the new solutions, their fitness is evaluated and then the worst population is replaced by the new best solutions.

4. **Experiments.** To evaluate the performance of IrBOA, computational experiments are conducted on various benchmark functions. In addition, the IrBOA is compared with the original rBOA [14], the mIDEA [16], BNGMM [19], samACO [26], and SGHS [27]. According to No Free Lunch theorem, "for any algorithm, any elevated performance over one class of problems is offset by performance over another class" [28]. Thus, the IrBOA tries to solve a class of benchmark functions in term of accuracy and efficiency. The test suite is chosen from the minimization benchmark functions introduced in [29]. This set of functions includes five uni-modal (from $F_1$ to $F_5$) and twenty multi-modal (from $F_6$ to $F_{25}$) functions (see Table 1). In the experiments, the quality of the obtained solution is taken as the performance measure. These benchmark functions have two important parameters: *dimensionality* $(D)$ and *shift value* $(F_{Bias})$. We set $D = 10$ and $D = 30$, and $F_{Bias} = 0$ as the default values.

4.1. **Experimental setting.** We describe the population initialization, the selection and replacement strategies, and the parameter setting of IrBOA.

4.1.1. *Population initialization.* In general, the random initialization can give a false impression on the performance especially in the functions whose optimum exists at or near the center of search space [30]. In this experiment, each individual is initialized with a random value; the values in every dimension are randomly chosen according to a uniform distribution over the search space. During the run of IrBOA, the position of an individual is not restricted to the defined boundaries. When an individual gets out of the defined boundaries, its position is set by a random value within the boundaries. Therefore, the population always exists in the feasible boundaries of the search space.

4.1.2. *Selection and replacement strategies.* The quality of the probabilistic model completely depends on the choice of selection method; but it does not depend on the replacement strategy [31]. It has been shown that in terms of the model learning, truncation selection is more effective than tournament selection [31]. Thus, the IrBOA employs the truncation selection (for selecting the parents) and the elitism replacement. In the truncation selection, the best $\tau\%$ individuals are selected as the candidate solutions. In the elitism replacement, the worst individuals in the current population are replaced by better individuals newly created.

4.1.3. *Parameters setting.* The parameters of IrBOA are chosen empirically as follows: (1) the population size, $Np = 2000$, (2) $\tau = 20$ in the truncation selection, (3) the complexity factor in the BIC score, $\lambda = 0.4$, (4) the parameter of the $k'$-means algorithm, $E = 0.3$, and (5) the number of initial clusters in the $k'$-means algorithm, $k = D$, where $D$ is the problem dimension. Moreover, the maximum number of function calls, $MAX_{NFC}$, is set to $1000 \times D$ in all the experiments.

4.2. **Effect of parameters.** We investigate the effect of the parameters of IrBOA.

TABLE 1. The CEC benchmark functions [29]

| No. | Function Name | Opt. | Range |
|---|---|---|---|
| $F_1$ | Shifted Sphere Function | 0.0 | [−100 100] |
| $F_2$ | Shifted Schwefel's Problem | 0.0 | [−100 100] |
| $F_3$ | Shifted Rotated High Conditioned Elliptic Function | 0.0 | [−100 100] |
| $F_4$ | Shifted Schwefel's Problem with Noise in Fitness | 0.0 | [−100 100] |
| $F_5$ | Schwefel's Problem with Global Optimum on Bounds | 0.0 | [−100 100] |
| $F_6$ | Shifted Rosenbrock's Function | 0.0 | [−100 100] |
| $F_7$ | Shifted Rotated Griewank's Function without Bounds | 0.0 | [0 600] |
| $F_8$ | Shifted Rotated Ackley's Function with Global Optimum on Bounds | 0.0 | [−32 32] |
| $F_9$ | Shifted Rastrigin's Function | 0.0 | [−5 5] |
| $F_{10}$ | Shifted Rotated Rastrigin's Function | 0.0 | [−5 5] |
| $F_{11}$ | Shifted Rotated Weierstrass Function | 0.0 | [−0.5 0.5] |
| $F_{12}$ | Schwefel's Problem 2.13 | 0.0 | [$\pi$ $\pi$] |
| $F_{13}$ | Expanded Extended Griewank's plus Rosenbrock's Function (F8F2) | 0.0 | [−3 1] |
| $F_{14}$ | Shifted Rotated Expanded Scaffer's $F6$ | 0.0 | [−100 100] |
| $F_{15}$ | Hybrid Composition Function | 0.0 | [−5 5] |
| $F_{16}$ | Rotated Version of Hybrid Composition Function | 0.0 | [−5 5] |
| $F_{17}$ | Rotated Hybrid Composition Function with Noise in Fitness | 0.0 | [−5 5] |
| $F_{18}$ | Rotated Hybrid Composition Function | 0.0 | [−5 5] |
| $F_{19}$ | Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum | 0.0 | [−5 5] |
| $F_{20}$ | Rotated Hybrid Composition Function with the Global Optimum on the Bounds | 0.0 | [−5 5] |
| $F_{21}$ | Rotated Hybrid Composition Function | 0.0 | [−5 5] |
| $F_{22}$ | Rotated Hybrid Composition Function with High Condition Number Matrix | 0.0 | [−5 5] |
| $F_{23}$ | Non-Continuous Rotated Hybrid Composition Function | 0.0 | [−5 5] |
| $F_{24}$ | Rotated Hybrid Composition Function | 0.0 | [−5 5] |
| $F_{25}$ | Rotated Hybrid Composition Function without Bounds | 0.0 | [2 5] |

4.2.1. *The effect of $\tau$:* We investigate the effect of the parameter $\tau$ (in the truncation selection) on the performance of IrBOA. For this purpose, the function $F_6$ ($D = 30$) is tested with a set of $\tau$ values; $\tau \in \{20, 40, 60\}$. The results are illustrated in Figure 2(a). It was shown that the performance is improved as the value of $\tau$ decreases; the best result was obtained when $\tau = 20$.

4.2.2. *The effect of $\lambda$:* In this experiment, the effect of the BIC parameter $\lambda$ is considered on the function $F_9$ ($D = 30$). As shown in Figure 2(b), the parameter $\lambda$ should be carefully set for getting a better solution; the best performance was achieved when $\lambda = 0.4$.

4.2.3. *The effect of the number of initial clusters, $k$:* The influence of the number of initial clusters, $k$, is studied on the function $F_9$ ($D = 30$). The results are given in Figure 2(c). A similar trend could be observed here as well; the best outcome was attained when $k = 30$.

4.3. **Comparison with existing algorithms.** The performance of IrBOA is compared with that of original rBOA [8] and mIDEA [16].

4.3.1. *Uni-modal test functions.* We compare the IrBOA with the original rBOA and the mIDEA on the uni-modal functions. In this experiment, the parameter setting on the original rBOA and the mIDEA are chosen from [17]. The results obtained by the algorithms on the uni-modal functions (i.e., $F_1 \sim F_5$) in the dimension of 10 and 30 are summarized in Tables 2 and 3, respectively. In the tables, the mean error value $\mu$ (between the obtained fitness and the global optimum) and the standard deviation $\sigma$ (of the fitness values) were averaged over 50 independent runs, and the best results were presented in boldface. The results showed that the IrBOA reaches the global optimum with 100% success rate for the functions $F_1$ and $F_2$ and obtains a better solution for the function $F_3$ in comparison to the mIDEA and the original rBOA in the dimension of 10. Also, the IrBOA performed better than the two references on the functions $F_1$, $F_2$ and $F_3$ in the dimension of 30. For the functions $F_4$ and $F_5$, similar trends were observed as well; the rBOA was superior to the original rBOA and the mIDEA.

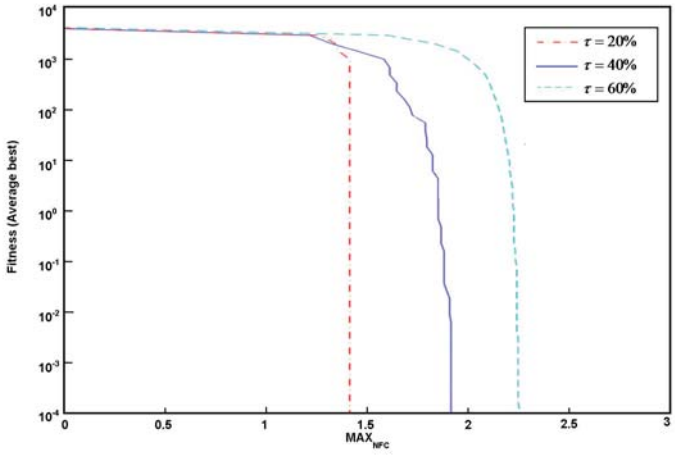
In addition, a two-tailed $t$-test to check whether the two algorithms are statistically different or not is performed by

$$t\text{-}test(alg.1, alg.2) = \frac{\hat{X}_1 - \hat{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}, \tag{15}$$

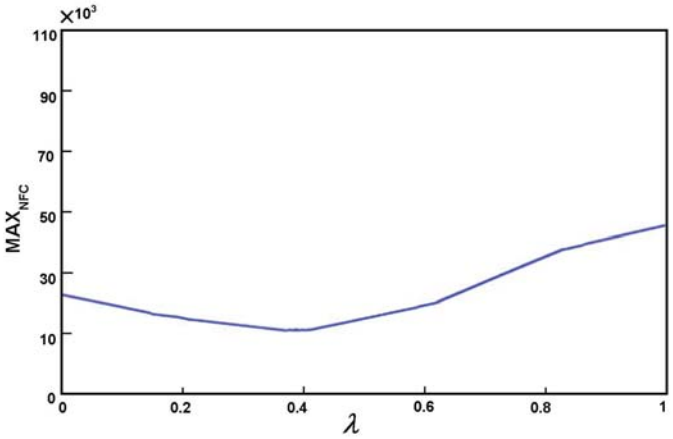
where $\hat{X}_i$, $S_i^2$, and $n_i$ are the mean, the standard deviation, and the sample size of the algorithm $i$, respectively. In this test, the IrBOA is always put to Algorithm 1; thus, a negative value denotes the superiority of the IrBOA over the reference. The statistical test results are given in Table 4. The results demonstrated that the proposed IrBOA clearly outperforms the original rBOA and the mIDEA in all the uni-modal test functions.

TABLE 2. Experimental results of the algorithms as applied to the uni-modal test functions with 10 dimension

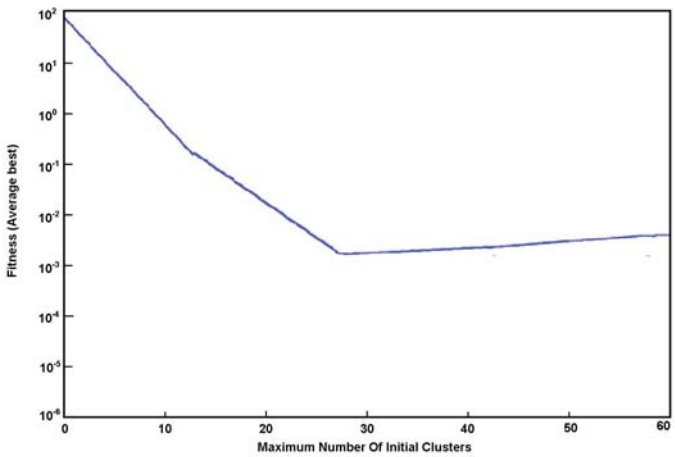
| Function | IrBOA | | Original rBOA | | mIDEA | |
|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| $F_1$ | **0.0** | 0.0 | **0.0** | 0.0 | **0.0** | 0.0 |
| $F_2$ | **0.0** | 0.0 | **0.0** | 0.0 | **0.0** | 0.0 |
| $F_3$ | **1.13e–25** | 3.03e–25 | 6.92e–24 | 1.60e–26 | 6.87e–22 | 1.40e–22 |
| $F_4$ | **0.0** | 0.0 | **0.0** | 0.0 | **0.0** | 0.0 |
| $F_5$ | **3.75e–15** | 1.54e–8 | 1.43e–9 | 6.28e–14 | 1.40e–7 | 4.35e–14 |




(a) Effect of the parameter $\tau$ on $F_6$ ($D = 30$)




(b) Effect of the parameter $\lambda$ on $F_9$ ($D = 30$)




(c) Effect of the number of initial clusters, $k$, on $F_9$ ($D = 30$)

FIGURE 2. Performance analysis of the parameters of IrBOA

TABLE 3. Experimental results of the algorithms as applied to the uni-modal test functions with 30 dimension

| Function | IrBOA | | Original rBOA | | mIDEA | |
|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| $F_1$ | **9.53e–10** | 4.63e–10 | 4.44e–8 | 1.23e–7 | 9.47e–9 | 4.45e–9 |
| $F_2$ | **6.94e–10** | 1.49e–10 | 6.94e–7 | 1.49e–6 | 9.85e–9 | 1.53e–10 |
| $F_3$ | **11.00e–30** | 42.01e–23 | 8.13e–7 | 1.74e–6 | 6.87e–25 | 1.40e–19 |
| $F_4$ | **8.13e–7** | 1.74e–6 | 4.20e–2 | 2.43e–2 | 1.11e–3 | 4.35e–5 |
| $F_5$ | **13.00e–2** | 0.05e+0 | 1.43e–1 | 6.28e–2 | 1.40e–01 | 4.35e–2 |

TABLE 4. Statistical results of the algorithms on the uni-modal test functions

| Function | IrBOA vs. Original rBOA | | IrBOA vs. mIDEA | |
|---|---|---|---|---|
| | $D = 10$ | $D = 30$ | $D = 10$ | $D = 30$ |
| $F_1$ | – | −2.49 | – | −13.46 |
| $F_2$ | – | −3.29 | – | −303.15 |
| $F_3$ | −130.63 | −3.30 | −34.69 | −0.01 |
| $F_4$ | – | −12.22 | – | −180.15 |
| $F_5$ | −0.66 | −1.14 | −64.28 | −1.06 |

TABLE 5. Experimental results of the algorithms when applied to the multi-modal test functions with 10 dimension

| Function | IrBOA | | Original rBOA | | mIDEA | |
|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| $F_6$ | **7.19e–14** | 2.04e–10 | 7.33e–9 | 0.11e–6 | 2.43e–6 | 0.03e–2 |
| $F_7$ | 7.32e–13 | 2.32e–11 | 2.28e–10 | 3.11e–12 | **6.23e–13** | 2.47e–9 |
| $F_8$ | **1.15e–5** | 5.32e–10 | 0.74e–3 | 1.46e–8 | 5.64e–2 | 3.54e–8 |
| $F_9$ | **0.0** | 0.0 | **0.0** | 0.0 | **0.0** | 0.0 |
| $F_{10}$ | **6.45e–5** | 4.35e–4 | 2.3e–3 | 3.48e–2 | 5.3e–2 | 3.33e–3 |
| $F_{11}$ | **3.48e–8** | 1.55e–4 | 5.52e–7 | 4.32e–4 | 8.45e–4 | 4.67e–3 |
| $F_{12}$ | **2.30e–15** | 0.72e–9 | 4.3e–6 | 2.3e–4 | 4.22e–6 | 1.26e–9 |
| $F_{13}$ | **2.48e–4** | 5.33e–1 | 1.22e–3 | 2.18e–2 | 3.67e–3 | 1.1e–2 |
| $F_{14}$ | **2.12e–7** | 0.14e–4 | 4.56e–4 | 0.03e–2 | 3.0e–2 | 0.43e–2 |
| $F_{15}$ | 7.12e–2 | 1.41e–1 | 2.34e–2 | 3.60e–2 | **4.23e–3** | 7.56e–4 |
| $F_{16}$ | **9.8e–4** | 7.6e–4 | 4.54e–2 | 3.68e–2 | 3.7e–2 | 6.21e–1 |
| $F_{17}$ | **1.31e–3** | 5.36e–1 | 3.78e–1 | 5.36e–1 | 3.46e–2 | 4.6e–2 |
| $F_{18}$ | **3.83e–3** | 3.25e–2 | 2.47e–2 | 2.42e–2 | 5.36e–2 | 5.9e–2 |
| $F_{19}$ | **4.7e–3** | 2.46e–2 | 3.67e–2 | 2.34e–2 | 7.55e–2 | 4.7e–2 |
| $F_{20}$ | 6.38e–1 | 1.35e–2 | 3.29e–2 | 3.58e–2 | **2.58e–3** | 4.81e–2 |
| $F_{21}$ | **3.73e–4** | 6.79e–2 | 4.36e–2 | 3.18e–2 | 7.83e–2 | 7.77e–2 |
| $F_{22}$ | **2.41e–3** | 3.4e–2 | 8.83e–2 | 1.34e–2 | 8.34e–2 | 1.4e–2 |
| $F_{23}$ | **6.34e–3** | 5.24e–2 | 5.34e–2 | 4.52e–1 | 2.34e–1 | 3.4e–1 |
| $F_{24}$ | **3.47e–3** | 1.9e–2 | 3.18e–2 | 3.8e–1 | 2.21e–2 | 1.72e–2 |
| $F_{25}$ | 5.23e–2 | 2.11e–2 | **5.21e–3** | 2.47e–1 | 1.49e–2 | 2.59e–2 |

TABLE 6. Experimental results of the algorithms when applied to the multi-modal test functions with 30 dimension

| Function | IrBOA | | Original rBOA | | mIDEA | |
|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| $F_6$ | **6.31e–9** | 1.14e–9 | 4.32e–5 | 2.25e–7 | 1.75e–1 | 2.20e–1 |
| $F_7$ | **6.48e–9** | 1.46e–9 | 9.07e–9 | 7.84e–10 | 1.50e–8 | 1.15e–6 |
| $F_8$ | **2.00e+1** | 9.62e–5 | 2.94e+1 | 5.76e–2 | 2.00e+1 | 4.97e–4 |
| $F_9$ | **8.25e–2** | 1.43e+0 | 1.51e–1 | 5.03e+0 | 2.79e+1 | 4.56e–1 |
| $F_{10}$ | **4.60e–1** | 2.73e+0 | 3.51e+2 | 1.03e+1 | 1.17e+2 | 7.11e+1 |
| $F_{11}$ | 4.44e–1 | 7.21e+0 | **2.74e–6** | 3.53e+0 | 3.29e–3 | 3.65e+0 |
| $F_{12}$ | **6.8e–11** | 5.66e–8 | 8.07e–2 | 9.52e–1 | 4.15e+1 | 2.49e+1 |
| $F_{13}$ | **0.54e+0** | 0.98e–1 | 5.15e+0 | 4.02e+0 | 0.52e+1 | 1.32e+0 |
| $F_{14}$ | **1.05e+1** | 8.45e–1 | 1.12e+1 | 6.68e–1 | 1.38e+1 | 1.90e–1 |
| $F_{15}$ | 2.44e+2 | 2.12e+1 | **2.18e+2** | 7.34e+1 | 8.76e+2 | 1.95e+1 |
| $F_{16}$ | **1.35e+1** | 4.06e+1 | 8.86e+1 | 9.69e+1 | 7.15e+1 | 8.10e+1 |
| $F_{17}$ | 2.92e+3 | 1.44e+2 | **1.34e+2** | 1.12e+2 | 1.56e+2 | 1.58e+2 |
| $F_{18}$ | **7.23e+1** | 2.22e–2 | 9.03e+2 | 5.95e–1 | 8.30e+2 | 1.61e+0 |
| $F_{19}$ | 5.35e+3 | 2.12e+0 | 9.03e+2 | 5.12e–1 | **8.31e+2** | 1.47e+0 |
| $F_{20}$ | **7.32e+0** | 2.98e–1 | 9.93e+2 | 2.30e–1 | 8.31e+2 | 1.32e+0 |
| $F_{21}$ | **1.40e+2** | 1.11e–10 | 2.30e+2 | 8.76e–14 | 1.59e+2 | 5.44e–11 |
| $F_{22}$ | **3.32e+1** | 1.57e+1 | 8.73e+2 | 1.82e+1 | 1.56e+3 | 4.83e+2 |
| $F_{23}$ | **2.44e+1** | 5.42e–5 | 5.87e+2 | 7.70e+1 | 8.66e+2 | 8.07e–1 |
| $F_{24}$ | **1.53e+1** | 1.42e+2 | 8.76e+2 | 2.55e+2 | 9.12e+2 | 3.89e–1 |
| $F_{25}$ | **2.43e+1** | 3.26e–1 | 2.32e+2 | 2.14e–1 | 2.13e+2 | 5.50e–1 |

4.3.2. *Multi-modal test functions.* The aim of this experiment is to evaluate the performance of IrBOA on the multi-modal optimization functions. The results of the algorithms on the multi-modal functions (i.e., $F_6 \sim F_{25}$) in the dimension of 10 and 30 are given in Tables 5 and 6, respectively. Moreover, the statistical test results are presented in Table 7. Note that the uni-modal and the multi-modal nature appears in the function $F_6$, the difficulty of finding the global optimum of the function $F_7$ increases as its dimension is reduced, and the function $F_8$ has a very narrow global basin and half of the dimensions are on the boundies, and the functions $F_9$ and $F_{10}$ have a huge number of local optima. Furthermore, the functions $F_{15} \sim F_{25}$ are novel composite functions built up with basic ones. From the results, it was observed that the IrBOA achieves better performance in most multi-modal test functions.

4.4. **Comparison with other state-of-the-art algorithms.** At this juncture, the proposed IrBOA is compared with some advanced algorithms: BNGMM [19], SamACO [26], and SGHS [27]. The BNGMM employs Bayesian network and Gaussian mixture model to solve continuous optimization problems [19]. The SamACO handles continuous variables by employing a sampling method that converts ACO from discrete optimization to continuous optimization [26]. The SGHA is a self-adaptive search method that maintains good information by the current best solution, which is used to create new harmonies [27]. The comparison results are shown in Tables 8-10, in which the performance of BNGMM, SamACO, and SGHS has been consulted from literature [19, 26, 27]. The results supported the claim that the IrBOA generally achieves better performance than BNGMN, SamACO, and SGHS on the considered test functions.

TABLE 7. Statistical results of the algorithms on the multi-modal test functions

| Function | Original rBOA | | mIDEA | |
|---|---|---|---|---|
| | $D = 10$ | $D = 30$ | $D = 10$ | $D = 30$ |
| $F_6$ | **−0.47** | –13.57e+2 | –5.72e–2 | **−5.62** |
| $F_7$ | **−68.65** | –11.05 | 0.03e–2 | **−0.05** |
| $F_8$ | –35.25e+4 | –11.53e+2 | –11.26e+6 | 0.0 |
| $F_9$ | **–** | **−0.09** | **–** | –13.1e+1 |
| $F_{10}$ | –0.45 | –2.32e+2 | –11.14e+1 | –11.58 |
| $F_{11}$ | –0.07e–1 | 0.39 | –1.27 | 0.38 |
| $F_{12}$ | –0.13 | –0.59 | –20.52e+3 | –11.78 |
| $F_{13}$ | –0.01 | –8.10 | –0.04 | –24.89 |
| $F_{14}$ | –10.73 | –4.59 | –49.33 | –26.94 |
| $F_{15}$ | 2.32 | 2.40 | 3.35 | –15.51e+1 |
| $F_{16}$ | –8.53 | –5.05 | –0.41 | –4.52 |
| $F_{17}$ | –3.49 | 10.79e+1 | –0.43 | 91.42 |
| $F_{18}$ | –3.64 | –98.65e+2 | –5.22 | –33.27e+2 |
| $F_{19}$ | –6.66 | 14.41e+3 | –9.43 | 2.38e+3 |
| $F_{20}$ | 11.18e+1 | –18.51e+3 | 8.99 | –43.04e+2 |
| $F_{21}$ | –4.07 | –5.73e+12 | –5.34 | –1.08e+12 |
| $F_{22}$ | –16.61 | –24.70e+1 | –15.57 | –22.34 |
| $F_{23}$ | –0.73 | –51.66 | –4.67 | –73.74e+2 |
| $F_{24}$ | –0.52 | –20.85 | –5.14 | –44.65 |
| $F_{25}$ | 1.34 | –37.66e+2 | 7.91 | –20.86 |

TABLE 8. Comparison of IrBOA and BNGMM on some test functions ($D = 4$)

| Function | IrBOA | BNGMM |
|---|---|---|
| | fitness | fitness |
| $F_1$ | **0.0** | 1.7e–4 |
| $F_2$ | **0.0** | 5.11e–5 |
| $F_6$ | **0.0** | 3.43e–7 |
| $F_7$ | **0.0** | 3.03e+0 |

TABLE 9. Comparison of IrBOA and SamACO on some test functions ($D = 30$)

| Function | IrBOA | | SamACO | |
|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| $F_6$ | **6.31e–9** | 1.14e–9 | 1.26e+2 | 2.94e+2 |
| $F_7$ | **6.48e–9** | 1.46e–9 | 0.02e+0 | 0.01e+0 |
| $F_8$ | 2.00e+1 | 9.62e–5 | **20.0e+0** | 4.3e–3 |
| $F_9$ | 8.25e–2 | 1.43e+0 | **1.59e–14** | 2.6e–14 |
| $F_{10}$ | **4.60e–1** | 2.73e+0 | 270e+0 | 86.9e+0 |

5. **Conclusions.** This paper has presented an improved rBOA (IrBOA) by employing an adaptive clustering method in the model selection phase. The clustering method is used to create a set of Bayesian networks that model promising solutions with respect to the dependencies of variables. The size of this set of Bayesian networks depends on the diversity of the selected solutions. This method can adaptively traverse the search

TABLE 10. Comparison of IrBOA and SGHS on several test functions ($D = 30$)

| Function | IrBOA | | SGHS | |
|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| $F_1$ | 5.42e–9 | 9.53e–10 | **0.0** | 0.0 |
| $F_6$ | **6.31e–9** | 1.14e–9 | 21.21e+2 | 39.66e–2 |
| $F_7$ | **6.48e–9** | 1.46e–9 | 13.3e+1 | 13.3e+0 |
| $F_9$ | **8.25e–2** | 1.43e+0 | 1.86e+0 | 0.3e+0 |
| $F_{10}$ | **4.60e-1** | 2.73e+0 | 98.7e+0 | 3.0e+1 |

space based on the population diversity. Due to the nature of evolutionary algorithms, the diversity of solutions decreases over time. Thus, the number of Bayesian networks in the IrBOA decreases over time and thus the PBBC can appear stronger. The experimental results showed that the IrBOA is capable to obtain more sophisticated solutions in comparison to the mIDEA, the original rBOA, and several up-to-date algorithms in optimizing various uni-modal and multi-modal optimization functions.

**Acknowledgment.** This research of the third author was supported by MKE, Korea under ITRC NIPA-2012-(H0301-12-3001).

## REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[2] D. E. Goldberg and K. Sastry, *Genetic Algorithms: The Design of Innovation*, 2nd Edition, Springer, 2010.

[3] C. W. Ahn, *Advances in Evolutionary Algorithms: Theory, Design and Practice*, Springer-Verlag, 2006.

[4] S. Oh, C. W. Ahn and M. Jeon, Effective constraints based evolutionary algorithm for constrained optimization problems, *International Journal of Innovative Computing, Information and Control*, vol.8, no.6, pp.3997-4014, 2012.

[5] F.-T. Lin and T.-R. Tsai, A Two-stage genetic algorithm for solving the transportation problem with fuzzy demands and fuzzy supplies, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.4775-4785, 2009.

[6] J. A. Lozano, P. Larrañaga, I. Inza and E. Bengoetxea, *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, Springer, Berlin, Germany, 2006.

[7] M. Pelikan, D. Goldberg and E. Cantu-Paz, BOA: The Bayesian optimization algorithm, *Proc. of GECCO'99*, pp.525-532, 1999.

[8] C. W. Ahn, R. S. Ramakrishna and D. E. Goldberg, Real-coded Bayesian optimization algorithm: Bringing the strength of boa into the continuous world, *Lecture Notes in Computer Science (GECCO'04)*, vol.3102, pp.840-851, 2004.

[9] K. R. Žalik, An efficient $k'$-means clustering algorithm, *Pattern Recognition Letters*, vol.29, pp.1385-1391, 2008.

[10] M. Pelikan, K. Sastry and E. Cantu-Paz, *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, Springer, 2006.

[11] G. Harik, Linkage learning in via probabilistic modeling in the ECGA, *Technical Report IlliGAL-99010*, University of Illinois at Urbana-Champaign, Urbana, IL, USA, 1999.

[12] H. Muhlenbein and T. Mahnig, FDA – A scalable evolutionary algorithm for the optimization of additively decomposed functions, *Evolutionary Computation*, vol.7, no.4, pp.353-376, 1999.

[13] M. Pelikan, *Bayesian Optimization Algorithm: From Single Level to Hierarchy*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2002.

[14] P. Larrañaga, R. Etxeberria, J. A. Lozano and J. M. Peña, Optimization in continuous domains by learning and simulation of Gaussian networks, *Proc. of GECCO Workshop*, pp.201-204, 2000.

[15] J. Ocenasek and J. Schwarz, Estimation of distribution algorithm for mixed continuous discrete optimization problems, *Proc. of the 2nd Euro-International Symposium on Computational Intelligence*, pp.227-232, 2002.
[16] P. A. N. Bosman, *Design and Application of Iterated Density-Estimation Evolutionary Algorithms*, Ph.D. Thesis, Utrecht University, Utrecht, The Netherlands, 2003.
[17] B. Moradabadi and H. Beigy, An improved bayes optimization algorithm for continuous environments using clustering techniques, *Proc. of the 16th National Computer Science*, Iran, pp.497-501, 2011.
[18] B. Moradabadi and H. Beigy, A new estimation of distribution algorithm for optimization problems, *Journal of Computer Science and Engineering*, Iran, 2012 (in press).
[19] X. Wei, Evolutionary continuous optimization by Bayesian networks and Gaussian mixture model, *Proc. of IEEE ICSP'10*, pp.1437-1440, 2010.
[20] M. Li, D. E. Goldberg, K. Sastry and T. L. Yu, Real-coded ECGA for solving decomposable real-valued optimization problems, *Linkage in Evolutionary Computation*, pp.61-86, 2008.
[21] D. Heckerman, D. Geiger and D. M. Chickering, A tutorial on learning with Bayesian networks, *Innovations in Bayesian Networks*, pp.33-82, 2008.
[22] D. Heckerman, D. Geiger and D. M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, *Technical Report MSR-TR-94-09*, Microsoft Research, Redmond, WA, 1994.
[23] M. Gallagher, I. Wood and J. Keith, Bayesian inference in estimation of distribution algorithms, *Proc. of IEEE Congress on Evolutionary Computation*, pp.127-133, 2007.
[24] C. W. Ahn and R. S. Ramakrishna, On the scalability of real-coded Bayesian optimization algorithm, *IEEE Transactions on Evolutionary Computation*, vol.12, no.3, pp.307-322, 2008.
[25] D. M. Chickering, D. Geiger and D. Heckerman, Learning Bayesian network is NP-hard, *Technical Report MSR-TR-94-17*, 1994.
[26] X. M. Hu, J. Zhang, H. Chung, Y. Li and O. Liu, Samaco: Variable sampling ant colony optimization algorithm, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol.40, no.6, pp.1555-1566, 2010.
[27] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren and J. J. Liang, A self-adaptive global best harmony search algorithm for continuous, *Applied Mathematics and Computation*, vol.216, pp.830-848, 2010.
[28] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation*, vol.1, no.1, pp.67-82, 1997.
[29] *Function Definitions and Performance Criteria for the Special Session on Real-Parameter Optimization at CEC-05*, http://www.ntu.edu.sg/home/EPNSugan, 2005.
[30] D. B. Fogel and H. G. Beyer, A note on the empirical evaluation of intermediate recombination, *Evolutionary Computation*, vol.3, no.4, pp.491-495, 1995.
[31] 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, *Proc. of Congress on Evolutionary Computation*, pp.1083-1090, 2007.