This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



# Multi-objective Estimation of Distribution Algorithm Based on Joint Modeling of Objectives and Variables

Hossein Karshenas, Roberto Santana, Concha Bielza, and Pedro Larrañaga

***Abstract*—This paper proposes a new multi-objective estimation of distribution algorithm (EDA) based on joint probabilistic modeling of objectives and variables. This EDA uses the multi-dimensional Bayesian network as its probabilistic model. In this way it can capture the dependencies between objectives, variables and objectives, as well as the dependencies learnt between variables in other Bayesian network-based EDAs. This model leads to a problem decomposition that helps the proposed algorithm to find better trade-off solutions to the multi-objective problem. In addition to Pareto set approximation, the algorithm is also able to estimate the structure of the multi-objective problem. To apply the algorithm to many-objective problems, the algorithm includes four different ranking methods proposed in the literature for this purpose. The algorithm is first applied to the set of walking fish group (WFG) problems, and its optimization performance is compared with a standard multi-objective evolutionary algorithm and another competitive multi-objective EDA. The experimental results show that on several of these problems and for different objective space dimensions the proposed algorithm performs significantly better and achieves comparable results on some other, when compared with the other two algorithms. The algorithm is then tested on the set of CEC09 problems, where the results show that multi-objective optimization based on joint model estimation is able to obtain considerably better fronts for some of the problems comparing with the search based on conventional genetic operators in the state-of-the-art multi-objective evolutionary algorithms.**

***Index Terms*—Estimation of distribution algorithm, Joint objective-variable modeling, Many-objective problem, Multi-objective optimization, Objectives relationship.**

## I. Introduction

MULTI-OBJECTIVE problems (MOPs) comprise several criteria that should be satisfied simultaneously, none of which can be preferred over others. Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be the set of objective functions. Then, given an MOP of the form

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{q} = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{D} \subseteq \mathbb{R}^n, \end{aligned} \tag{1}$$

the goal of a multi-objective optimization algorithm is to search for solutions that satisfy all or obtain an optimal trade-off between objectives. Note that here, without loss of generality, it is assumed that all objective functions should to be minimized.

Multi-objective evolutionary algorithms (MOEAs) [1]–[5] are considered as promising optimizers that have been successfully applied to a variety of MOPs. These algorithms use their nature-inspired operators to evolve a population of candidate solutions. This population-based optimization parallelizes the algorithm which can then simultaneously optimize several areas of the search space to arrive at several compromise solutions, as it is necessary when solving MOPs.

It is well known that in the presence of specific problem properties, traditional evolutionary algorithms (EAs) have difficulties in optimization [6]. Estimation of distribution algorithms (EDAs) [7]–[10] are a relatively new computational paradigm proposed to overcome these difficulties. EDAs have also been applied to solve MOPs [11]–[14]. Instead of genetic operators, these algorithms generate new candidate solutions from a probabilistic model, which is learnt from a set of promising solutions. The probabilistic model captures certain statistics about the values of problem variables and the important dependencies existing between these variables.

An important issue concerning MOEAs is how well they scale as the number of objectives in the MOP increases [15], [16]. This is especially important because real-world problems usually have many criteria that can be formulated as a many-objective problem. One way of accounting for this is to consider the relationships between objectives and explicitly reduce the number of objectives according to these relationships. Different methods, like correlation and principal component analysis [17]–[21], extending the definition of conflicting objectives [22], and linear programming [23] have been proposed for this purpose. These methods reduce optimization complexity by searching for a minimum subset of objectives.

In this study, we propose learning a joint probabilistic model of both objectives and variables within the context of EDAs. This allows the algorithm not only to capture the dependencies between variables, as in other EDAs, but also to learn the relationships between objectives and between objectives and

This work has been partially supported by Consolider Ingenio 2010-CSD2007-00018, TIN2010-20900-C04-04, TIN2010-14931 and Cajal Blue Brain projects (Spanish Ministry of Science and Innovation), and Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government).

H. Karshenas, C. Bielza and P. Larrañaga are with the Computational Intelligence Group, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain. E-mail: {hkarshenas, mcbielza, pedro.larranaga}@fi.upm.es

R. Santana is with Intelligent System Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Paseo Manuel de Lardizbal 1, 20080 San Sebastin–Donostia, Spain. E-mail: roberto.santana@ehu.es

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs_permisssions@ieee.org.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



variables. The learnt relationships can have more complex patterns of interaction than just linear correlation. These relationships are then implicitly used by the algorithm to generate new solutions in the search space. In addition to the approximated Pareto set, the joint probabilistic model learnt in this EDA provides the decision maker with an approximation of the MOP structure, i.e., the relationships among variables and objectives in MOP.

A preliminary study of this notion was presented in [24], discussing the incorporation of objectives into EDA model building. In this paper, we extend the study by using a specific probabilistic modeling adapted from a multi-dimensional Bayesian network (MBN), usually used for multi-dimensional classification tasks [25], [26]. In this type of problems, each instance or data point can have several class labels. The goal of learning a probabilistic model is then to predict the class labels of new unseen data points. On the other hand, in multi-objective optimization with EDAs, each solution in the search space has several objective values, and the goal is to generate new solutions from the probabilistic model that have better objective values. Clearly, there are similarities between the two problems that motivate the use of an MBN to capture the relationships between variables and objectives. For this purpose, the objectives are modeled as class variables in the probabilistic model and the dependencies learnt between objectives and variables in the model are exploited to generate new solutions. Using this type of model estimation, the structure of the MOP can be approximated systematically, and the proposed algorithm can be applied to many-objective problems. This paper examines how this algorithm performs on many-objective problems, using different ranking methods, and studies some of the MOP structures obtained by the algorithm.

The use of Bayesian network classifiers as the probabilistic model of an EDA has been previously reported for single-objective optimization in the evolutionary Bayesian classifier-based optimization algorithm (EBCOA) [27], [28]. However, there are several key differences between EBCOA and the algorithm presented in this paper. First, the presence of multiple objectives in an MOP increases the information about the quality of solutions (possibly contradictory) that should be addressed during modeling. Second, instead of classifying the solutions into disjoint classes, which may blur the differences in the quality of the solutions, here the continuous objective values are directly used in model learning. Third, in contrast to a fixed dependency between the objectives and variables, the algorithm presented here dynamically learns the relationships between the objectives and variables. In this way, the model can select a subset of variables that has more influence on each objective.

The rest of this paper is organized as follows. Section II briefly reviews some background required to follow the discussions in the paper. The proposed EDA based on joint modeling of objectives and variables is described in Section III. Section IV presents the numerical results of applying the algorithm on two sets of MOPs and analyzes the results. A study of the possible MOP structures learnt by the algorithm is presented in Section V. Finally, the paper is concluded in Section VI and some lines of future research are proposed.

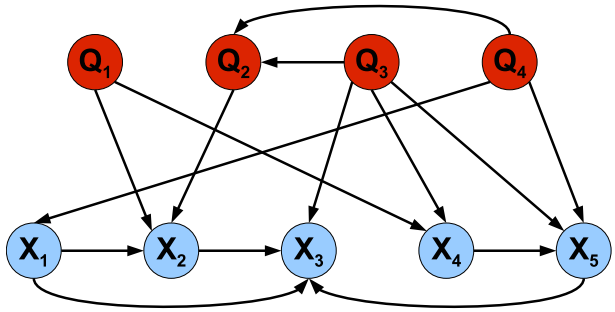



Fig. 1. An example of a multi-dimensional Bayesian network structure.

## II. Preliminaries

### A. Multi-dimensional Bayesian Network Classifiers

Bayesian networks [29] are multivariate probabilistic graphical models, consisting of two components:

- the structure, represented by a directed acyclic graph (DAG), where the nodes are the problem variables and the arcs are conditional (in)dependencies between triplets of variables, and
- the parameters, expressing for each variable $X_i$ the conditional probability of each of its values, given different value-settings of its parent variables ($\boldsymbol{Pa}(X_i)$) according to the structure, i.e.,

$$p(x_i \mid \boldsymbol{pa}(X_i)),$$

where $\boldsymbol{pa}(X_i)$ is a value-setting for the parent variables in $\boldsymbol{Pa}(X_i)$.

By introducing a special node $C$ into the network as the class node, Bayesian networks can be used for classification tasks to obtain the posterior probability of a class value $c$, given feature values $x_1, \ldots, x_n$, i.e., $p(c \mid x_1, \ldots, x_n)$. Several types of Bayesian network classifiers have been proposed in the literature: naïve Bayes, seminaïve Bayes, tree augmented naïve Bayes, etc.

If a data point can simultaneously belong to several (say $m$) classes, then a multi-dimensional Bayesian network (MBN) can be learnt to perform multi-dimensional classification, where the posterior probability is now given by

$$p(c_1, \ldots, c_m \mid x_1, \ldots, x_n).$$

Fig. 1 shows an example of the structure of an MBN used for multi-dimensional classification. In this type of model, the nodes are organized in two separate layers: the top layer comprises class variables and the bottom layer contains feature variables. The set of arcs in the structure is partitioned into three subsets, resulting in the following subgraphs:

- the class subgraph, containing the class nodes and the interactions between them,
- the feature subgraph, comprising the feature variables and their relations, and
- the bridge subgraph, depicting the *one-way* links from class nodes to feature nodes.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



This probabilistic model can answer several types of questions: the class labels of a given data point, the most probable feature values for a given combination of class labels, and the most probable values for a subset of features or classes given the value of the others. Considering the similarity between multi-dimensional classification and MOPs, the respective questions will be: what are the estimated objective values of a given solution, what is the most probable solution resulting in a specific value-setting for the objectives, and, having found the values of some objectives or variables, what will be the most probable values of the others. Also, worthy of note is that the existence of specific types of decomposability in the MBN structure can make these types of inference questions simpler to answer [26].

### B. Gaussian Bayesian Networks

In domains with continuous-valued variables, it is usually assumed that the variables follow a Gaussian distribution. The Bayesian network learnt for a set of variables, having a multivariate Gaussian distribution $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as their joint probability function, is called a Gaussian Bayesian network (GBN). Here, $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix of the distribution.

The structure of a GBN is similar to any other Bayesian network. However, for each node the conditional probability represented by the parameters is a univariate Gaussian distribution for the variable corresponding to that node, which is determined by the values of the parent variables [30], [31]

$$p\big(x_i \mid \boldsymbol{pa}(X_i)\big) = \mathcal{N}\big(\mu_i + \sum_{X_j \in \boldsymbol{Pa}(X_i)} w_{ij}(x_j - \mu_j), \nu_i^2\big), \quad (2)$$

where $\mu_i$ is the mean of variable $X_i$, $\nu_i$ is the conditional standard deviation of the distribution, and regression coefficients $w_{ij}$ specify the importance of each of the parents. These are the parameters stored in each node of a GBN. $x_j$ is the corresponding value of $X_j$ in $\boldsymbol{pa}(X_i)$.

### C. Estimation of Distribution Algorithms

Conventional genetic operators used for generating new solutions in evolutionary algorithms usually provide good and efficient exploration of the search space. However, it has been shown that these operators can disrupt the good sub-solutions found during evolution, affecting the effectiveness of the search for optimal solutions [32]. This disruption is more likely to occur as the correlation between problem variables increases, rendering the algorithm inefficient for such problems. *Estimation of distribution algorithms* (EDAs) make use of probabilistic models to replace the genetic operators in order to overcome this shortcoming of traditional evolutionary algorithms.

Fig. 2 shows the basic steps of a typical EDA. The set of selected solutions $S_t$ serves as a training dataset to estimate the probabilistic model and leads the search towards regions with better fitness values (represented by the selected solutions). The set of new solutions $U_t$ is generated using the probabilities encoded in the probabilistic model in accordance with the statistics collected from the solutions in $S_t$. The choice of probabilistic model can have a major influence on the performance and efficiency of EDAs. For example, some probabilistic models can also encode the dependencies between the variables, and use it to identify and preserve these dependencies in the sampling process. Bayesian networks are one of these probabilistic models that can encode dependencies between any number of variables. Thus EDAs using this probabilistic model can be applied to problems with high order of relationships between variables and a complex structure [10].

**Inputs:**
Representation of solutions
Objective function $f$

$P_0 \leftarrow$ Generate initial population according to the given representation
$F_0 \leftarrow$ Evaluate each individual $\boldsymbol{x}$ of $P_0$ using $f$
$t \leftarrow 0$
**while** termination criteria are not met **do**
  $S_t \leftarrow$ Select a subset of $P_t$ according to $F_t$
  $\hat{\rho}_t(\boldsymbol{x}) \leftarrow$ Estimate the probability density of solutions in $S_t$
  $U_t \leftarrow$ Sample from $\hat{\rho}_t(\boldsymbol{x})$ according to the given representation
  $H_t \leftarrow$ Evaluate $U_t$ using $f$
  $P_{t+1} \leftarrow$ Incorporate $U_t$ into $P_t$ according to $F_t$ and $H_t$
  $F_{t+1} \leftarrow$ Update $F_t$ according to the solutions in $P_{t+1}$
  $t \leftarrow t + 1$
**end while**

**Output:** The best solution(s) in $P_t$

Fig. 2. The basic steps of an estimation of distribution algorithm

In the context of multi-objective optimization, when there is more than one objective function in the problem, $F_t$ will be a matrix with $m$ columns instead of a vector. A successful strategy adopted by many MOEAs (including multi-objective EDAs) [33], [34] is to modify the solution selection and replacement mechanisms but use the same solution reproduction approach (model learning and sampling for EDAs) as in single objective optimization. However, as we discuss later, more objectives in the problem can also affect how the new solutions are being generated. In the remainder of this section we review some of methods proposed in literature for multi-objective optimization based on model estimation. A summary of multi-objective EDAs is given in TABLE I.

### D. A Survey of Multi-objective Optimization with Probabilistic Modeling

In several multi-objective EDAs proposed in the literature, a Bayesian network is estimated as the probabilistic model. Pareto Bayesian optimization algorithm [35] integrates the Pareto strength-based solution ranking method [33] into Bayesian optimization algorithm (BOA) [56] for multi-objective optimization. In a similar study, the non-dominated sorting algorithm [57] is used in BOA [36]. Bayesian multi-objective optimization algorithm (BMOA) [48] uses an $\epsilon$-Pareto dominance based ranking method to select a subset of solutions for estimating a Bayesian network model. Decision tree based multi-objective EDA (DT-MEDA) [49] uses regression decision tree with Gaussian kernels in its leaves as the probabilistic model. It uses a slightly modified version of non-dominated sorting algorithm for selecting a subset of solutions in each generation.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



TABLE I
SUMMARY OF THE MULTI-OBJECTIVE EDAS WITH THEIR RANKING METHODS AND PROBABILISTIC MODELS USED TO SEARCH IN THE SPACE OF CANDIDATE SOLUTION.

| Name | Probabilistic Model | Mixture | Hybridized | Ranking Method | External Archive |
|---|---|---|---|---|---|
| Pareto BOA [35] | Bayesian network | | | Pareto dominance, Pareto strength, Domination count | • |
| mBOA [36] | Bayesian network | | | Pareto dominance, Crowding distance | |
| MIDEA [37] | Univariate distribution, Tree-structured distribution | • | | Domination count, Spread in objective space | |
| SDR-AVS-MIDEA [38] | Multivariate Gaussian distribution | • | | Domination count, Spread in objective space | • |
| MOPEDA [39] | Gaussian and Cauchy kernel | • | | Pareto dominance, Crowding distance | |
| VEDA [40] | Geometric distribution with uniform noise | • | | Pareto dominance, Crowding distance | • |
| mohBOA [41] | Bayesian network | • | | Pareto dominance, Crowding distance | |
| meCGA [42] | Marginal product model | • | | Pareto dominance, Crowding distance | |
| moPGA [43] | Univariate marginal product model | • | | Pareto dominance, Crowding distance | • |
| dp-MrBOA [44] | Product of Gaussian Bayesian networks | • | | Pareto dominance, Domination count, Modified crowding distance | |
| RM-MEDA [13] | Linear combination of principal components with Gaussian noise | • | | Pareto dominance, Crowding distance | |
| MORBM [45] | RBM network | • | | Pareto dominance, Crowding distance | |
| MB-GNG [46] | Univariate Gaussian distribution | • | | Hypervolume indicator | |
| MARTEAD [47] | Univariate Gaussian distribution | • | | Hypervolume indicator estimation | |
| BMOA [48] | Bayesian network | | | $\epsilon$-Pareto dominance, Domination count | |
| DT-MEDA [49] | Gaussian Bayesian network | | | Pareto dominance, Crowding distance | |
| Tabu-BOA [50] | Bayesian network | | • | Pareto dominance | • |
| MOHEDA [51] | Univariate distribution | • | • | Weighted sum | |
| EDA-PSO [52] | Univariate Gaussian distribution | | • | Pareto dominance, Crowding distance | |
| PLREDA [53] | RBM network | • | • | Pareto dominance, Crowding distance, Probabilistic dominance | |
| hNSEA, hMOEA/D [54] | RBM network | | • | Pareto dominance, Crowding distance Decomposition with Tchebycheff method | |
| UMEGS, UMSA, UMHC [55] | Univariate distribution | | • | Decomposition with Tchebycheff method | |

Some of the proposed EDAs explicitly estimate a mixture of probability distributions by means of a clustering method to obtain a well-spread approximation of the Pareto solutions. Multi-objective mixture-based iterated density estimation evolutionary algorithm (MIDEA) [37] clusters the selected solutions into several groups in the objective space and learns a separate component for each group of solutions. Probabilistic models with different orders of complexity (e.g. encoding univariate, bivariate or multi-variate dependencies) are tested as the components of the mixture. The proposed algorithm is also further improved by maintaining an $\epsilon$-Pareto archive and introducing adaptive variance scaling to prevent premature convergence in continuous MOPs [38].

Multi-objective Parzen-based EDA (MOPEDA) [39] applies a Parzen estimator to learn a mixture of kernel functions in order to reduce the variance of the probability distribution estimation. Both Gaussian and Cauchy kernels are used alternatively during evolution. Voronoi-based EDA (VEDA) [40] constructs a Voronoi diagram by considering the inferior solutions in addition to good solutions in each generation. It also uses principal component analysis to reduce the dimensionality of the objective space.

In multi-objective hierarchical BOA (mohBOA) [41] each component of the mixture is a Bayesian network. To obtain a well-distributed approximation of Pareto front, approximately equal shares are allocated to the mixture components during model sampling. In the multi-objective extended compact genetic algorithm (meCGA) [42] a marginal product model is used for each component of the mixture. The algorithm is later improved by using an $\epsilon$-Pareto dominance based clustering and algorithm parameters are dynamically computed during evolution [43].

In addition to clustering the solutions, the diversity preserving multi-objective real-coded Bayesian optimization algorithm (dp-MrBOA) [58] decomposes the problem variables by estimating a Gaussian Bayesian network. It employs a diver-

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



sity preserving selection method which uses adaptive sharing and dynamic crowding methods [44]. Regularity model-based multi-objective EDA (RM-MEDA) [13] estimates a piecewise continuous manifold using the local principal component analysis algorithm. Each mixture component is an affine plus a Gaussian noise.

In restricted Boltzmann machine (RBM)-based EDA [45] each mixture component is an RBM, a stochastic neural network with hidden neurons. Model building growing neural gas (MB-GNG) algorithm [46] uses a specific single-layer neural network, called growing neural gas, to determine the location of mixture components which are Gaussian distributions. The approach is further extended in [47] by using the adaptive resonance theory and employing a hypervolume indicator-based selection method [59].

There are also some approaches which have combined the EDA search method with other search heuristics. In multi-objective hybrid EDA (MOHEDA) [51] an EDA based on a mixture of univariate models is hybridized with a local search method that is applied to the solutions generated from the probabilistic model of EDA. Tabu-BOA [50] uses the tabu lists maintained in tabu search to improve multi-objective BOA. Gao et al. [52] proposed an algorithm which hybridizes an EDA based on univariate distributions with a particle swarm optimization (PSO) algorithm. Very recently, the RBM-based EDA has also been hybridized with PSO for noisy multi-objective optimization [53].

While most of the multi-objective EDAs use selection methods based on Pareto dominance, there are also some works which have studied EDAs in other multi-objective optimization frameworks. Shim et al. [54] proposed a multi-objective optimization algorithm with dynamic combination of operators in genetic algorithms, differential evolution and EDAs during search. The proposed algorithm is tested when using both Pareto dominance-based and decomposition-based selection methods. They have also used decomposition-based selection in an algorithm which hybridizes EDA based on univariate distributions with several local search methods [55].

Although in EDAs probabilistic models are used to estimate the values for problem variables and to generate new solutions based on these estimations, probabilistic modeling has also been used for estimating the values of objective functions. Zhang et al. [60] proposed a decomposition-based MOEA which uses Gaussian processes to estimate surrogate models of the objective functions in MOPs with computationally expensive (cost or time) objectives.

## III. MBN-EDA

### A. Joint Modeling of Variables and Objectives

It is common practice in most of EDAs to estimate a probabilistic model of only the problem variables encoding the characteristics of the selected solutions $S_t$ (see Algorithm 2). The sampling algorithm is then expected to generate a new set of solutions $U_t$ from this model according to the statistics collected from the solutions in $S_t$. Apart from this, there is no requirement for the solutions in $U_t$ to have better or comparable objective values to those in $S_t$.

Using this solution generation scheme, exploration of the search space, driven by the characteristics encoded in the probabilistic model, is usually good. To extend the scheme in order to account for objective values, the objectives can also be encoded in the model. In this way, preferences concerning objective values (obtained from the selected solutions) can be encoded in the model. This applies especially to MOPs, where because of having several objectives more information about the quality of the solutions is available. In this study, we show that this type of information, handled by expressive probabilistic models, turns out to be useful for solving multi-objective problems.

The joint learning of objectives and variables also suggests a new way for estimating the relationships between MOP variables and objectives. These relationships can be exploited by the optimization algorithm to facilitate the search, focusing only on variables that influence the values of an objective. Thus, an implicit variable selection is taking place for each of the objectives. Moreover, the relationships between the objectives are also captured, helping to identify how the values of some objectives might change against the values of some others, using the relationships encoded in the model.

### B. An EDA based on MBN Estimation

The probabilistic model used in this paper for joint model learning is an MBN. The variables are modeled as feature nodes and objectives as continuous-valued class nodes. The feature subgraph encodes the dependencies between problem variables like the models learnt by other EDAs that use Bayesian networks as their probabilistic model [35], [36], [41], [48], [50], [58]. The bridge and class subgraphs, however, encode new types of dependencies. The bridge subgraph shows the relationships between each objective and the variables, and the class subgraph represents the direct interactions between objectives.

Let $(\boldsymbol{X}, \boldsymbol{Q}) = (X_1, \ldots, X_n, Q_1, \ldots, Q_m)$ denote the joint vector of problem variables and objectives respectively (of size $n + m$). Then, like any other Bayesian network, the learnt MBN encodes a factorization of the joint probability distribution of its constituent variables. This will give an implicit decomposition of the MOP corresponding to this joint vector. The joint probability distribution of this MBN is given by

$$p\left(x_1, \ldots, x_n, q_1, \ldots, q_m\right) = \prod_{i=1}^{n} p(x_i|\boldsymbol{pa}(X_i)) \cdot \prod_{j=1}^{m} p(q_j|\boldsymbol{pa}(Q_j)), \quad (3)$$

where $\boldsymbol{Pa}(X_i) \subseteq \{\mathbf{X} \cup \mathbf{Q} \setminus X_i\}$ and $\boldsymbol{Pa}(Q_j) \subseteq \{\mathbf{Q} \setminus Q_j\}$ respectively are the parents of each variable and objective node, and $\boldsymbol{pa}(X_i)$ and $\boldsymbol{pa}(Q_j)$ represent one of their possible value-settings. $\boldsymbol{q} = (q_1, \ldots, q_m)$ denotes a possible value-setting for the objective variables $\boldsymbol{Q} = (Q_1, \ldots, Q_m)$.

The proposed algorithm, which is called MBN-EDA, uses this probabilistic model to capture the characteristics of selected solutions and their objective values, and generates new candidate solutions to the MOP at hand in search for

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



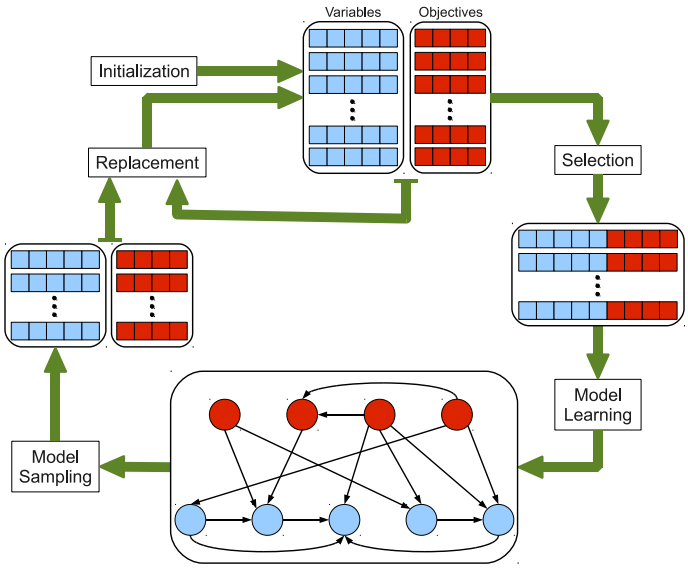



Fig. 3. An overview of the proposed MBN-EDA

the Pareto optimal solutions. Fig. 3 shows the algorithm layout. After selecting a subset of solutions according to a selection mechanism, e.g., non-dominated sorting + truncation selection, the solutions are joined with their objective values to form extended solutions. These extended solutions, comprising values for both variables and objectives, are used to serve as a dataset for estimating an MBN. The model sampler generates new candidate solutions from the learnt MBN, taking into consideration the values of both objectives and variables. Finally, these new solutions are added to the population based on a replacement strategy. The following sections provide more details about the algorithm.

### C. Solution Ranking for Elitist Selection

In contrast to single objective optimization, where the objective values can be used directly to rank solutions, the existence of multiple objectives in MOPs necessitates the application of an intermediate function of the form

$$G : \mathcal{Q} \subseteq \mathbb{R}^m \mapsto \mathcal{T} \subseteq \mathbb{R},$$

whose output will be used to rank the solutions. One of the most commonly used techniques in multi-objective optimization is the non-dominated sorting algorithm [57], which sorts the solutions into non-dominated Pareto fronts and then sorts the solutions within each front according to their crowding distances in the objective space. However, it has been shown that the effectiveness of this ranking method decreases as the number of objectives increases [61]–[63].

Finding efficient ranking methods for many-objective optimization (i.e. when there are more than three objectives) is the topic of ongoing research, and several methods have been proposed in the literature. In this study, we adopt four methods, which have been reported to show better performance for evolutionary many-objective optimization comparing with several other methods [64]–[67], to rank the solutions in the MBN-EDA selection step.

Let $\boldsymbol{q} = (q_1, \ldots, q_m) = \big(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\big)$ and $\boldsymbol{r} = (r_1, \ldots, r_m) = \big(f_1(\boldsymbol{y}), \ldots, f_m(\boldsymbol{y})\big)$ be the objective values obtained for two solutions $\boldsymbol{x}$ and $\boldsymbol{y}$, where $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D} \subseteq \mathbb{R}^n$, and assume all objectives are to be minimized. Then the employed ranking methods are as follows:

- Weighted sum of the objectives, using a weight vector $\boldsymbol{w} = (w_1, \ldots, w_m)$ showing the importance of each objective:
$$G_{\mathrm{WS}}(\boldsymbol{q}) = \sum_{i=1}^{m} w_i q_i. \tag{4}$$
- Distance to the best objective values $\boldsymbol{b} = (b_1, \ldots, b_m)$, using some distance measure $\mathrm{d}(\cdot,\cdot)$ in the objective space (e.g., Euclidean distance):
$$G_{\mathrm{DB}}(\boldsymbol{q}) = \mathrm{d}(\boldsymbol{b}, \boldsymbol{q}). \tag{5}$$
When the best objective values are not known beforehand (which is usually the case), the best objective values achieved so far (considering each objective individually) in the current population ($F_t$ in Algorithm 2) can be used, i.e., the best value $b_i$ for objective $f_i$ is
$$b_i = \min_{\boldsymbol{q} \in F_t} \{q_i\}.$$
- Global detriment or the total gain lost by each solution against other solutions in the population:
$$G_{\mathrm{GD}}(\boldsymbol{q}) = \sum_{\forall \boldsymbol{r} \in F_t, \boldsymbol{r} \neq \boldsymbol{q}} \mathrm{gain}(\boldsymbol{r}, \boldsymbol{q}), \tag{6}$$
where the function $\mathrm{gain}(\cdot,\cdot)$ computes the gain obtained in the objective values by a solution $\boldsymbol{q}$ compared to another solution $\boldsymbol{r}$:
$$\mathrm{gain}(\boldsymbol{q}, \boldsymbol{r}) = \sum_{i=1}^{m} \max\{0, r_i - q_i\}. \tag{7}$$
- Profit of the gain obtained from each solution against other solutions in the population:
$$G_{\mathrm{PG}}(\boldsymbol{q}) = \max_{\boldsymbol{r} \in F_t, \boldsymbol{r} \neq \boldsymbol{q}} \mathrm{gain}(\boldsymbol{q}, \boldsymbol{r}) - \max_{\boldsymbol{r} \in F_t, \boldsymbol{r} \neq \boldsymbol{q}} \mathrm{gain}(\boldsymbol{r}, \boldsymbol{q}), \tag{8}$$
where the definition of $\mathrm{gain}(\cdot,\cdot)$ is equal to (7) above.

Applied to the objective values obtained for the MOP candidate solutions, these ranking methods will result in an ordered set, which is then used to select a subset of solutions. Any selection mechanism can be simply applied on the ordered set. MBN-EDA uses *truncation* selection where the best $\tau \cdot N$ solutions (according to the ranking method) of the population are selected for a given $\tau \in (0, 1)$, where $N$ is the number of solutions in the population.

### D. Solution Reproduction based on Probabilistic Modeling

*1) Estimating the Probabilistic Model:* A search+score strategy [68]–[70] is used in MBN-EDA to learn the MBN from the data. In this strategy, a search algorithm is employed to explore the space of possible MBN structures to find a structure that closely matches the data. The quality of different

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



MBN structures obtained in this search process is measured using a scoring metric, usually computed from data. Although using a search algorithm (structure search) within another search algorithm (solution search) may appear to be circular, note that the aim of the structure search algorithm is to find a structure that adequately represents data characteristics rather than finding the optimum structure. For further discussion refer to [71].

A greedy local search algorithm is used to learn the structure of MBN (Fig. 4). In each iteration, this algorithm weighs all possible arc addition, removal and reversal operations that will map the current network structure to a new valid structure (according to the MBN structural constrains) in a single step and then applies the operation that will result in the highest increase in the network score [72]. The structure search is restarted from a new random structure after reaching a local optimum of the scoring function up to a maximum number of node score evaluations. The algorithm finally returns the highest scoring network in all these sub-searches. The Bayesian information criterion (BIC) [73] is used to score possible MBN structures. This score is based on a penalized log-likelihood measure

$$\begin{aligned}\sum_{k=1}^{N}\Bigg(&\sum_{i=1}^{n}\log\Big(p\big(x_{ki}\mid \boldsymbol{pa}_k(X_i)\big)\Big)\\&+\sum_{j=1}^{m}\log\Big(p\big(q_{kj}\mid \boldsymbol{pa}_k(Q_j)\big)\Big)\Bigg)\\-\frac{1}{2}\log(N)\Big(&\sum_{i=1}^{n}|\boldsymbol{Pa}(X_i)|+\sum_{j=1}^{m}|\boldsymbol{Pa}(Q_j)|\\&+2(n+m)\Big),\end{aligned}\tag{9}$$

where $x_{ki}$ and $q_{kj}$ are the values of variables $X_i$ and $Q_j$ in the $k$th extended solution, respectively. Similarly, $\boldsymbol{pa}_k(X_i)$ and $\boldsymbol{pa}_k(Q_j)$ are respectively the value-settings for the parents of variables $X_i$ and $Q_j$ in the $k$th extended solution. $|\boldsymbol{Pa}(X_i)| \leq n+m-1$ and $|\boldsymbol{Pa}(Q_j)| \leq m-1$ respectively show the number of parents of $X_i$ and $Q_j$, according to the MBN structure.

The second term in (9), which is the penalizing term, is computed assuming that MBN is implemented in continuous domains as a GBN. The parameters of this type of MBN are computed from the mean vector and covariance matrix of the multivariate Gaussian distribution (MGD) estimated for the joint vector of variables and objectives: $\mathcal{N}(\hat{\boldsymbol{\mu}}_{<1\times(n+m)>}, \hat{\boldsymbol{\Sigma}}_{<(n+m)\times(n+m)>})$. Usually the maximum likelihood (ML) estimation is used to estimate the parameters of MGD (the mean vector and covariance matrix) from the data. However, when the dataset is small comparing to the number of parameters that should be estimated, this method cannot obtain a robust estimation of the parameters and especially the covariance matrix which should be symmetric and positive-definite. In our case, since the solutions are extended by appending the objective values, this problem becomes even worse.

Regularization techniques [74], [75] are one of the methods that can be used to overcome this problem. MBN-EDA uses the covariance shrinkage method [76] to obtain a better estimation of MGD for the joint vector of variables and objectives. In this method, the ML estimation of the covariance matrix is linearly combined with a simpler target matrix, which has a smaller number of parameters and thus can be estimated more accurately. More specifically, a diagonal matrix with zeros in all off-diagonal entries is used as the target to enforce shrinkage towards sparser matrices while leaving the diagonal elements (variances) intact, preventing early loss of diversity:

$$\boldsymbol{\Sigma}^* = (1-\lambda)\hat{\boldsymbol{\Sigma}} + \lambda \boldsymbol{T}. \tag{10}$$

Here, $\boldsymbol{T}$ represents the target matrix and $\lambda$ is the shrinkage intensity (also called regularization parameter), which can be computed analytically. In practice, the ML estimation of the correlation matrix is used to compute the shrinkage intensity for the specified diagonal target matrix in a data-driven manner, minimizing a mean square error loss function. The regularized estimation in (10) leads to a statistically more efficient covariance matrix that is well-conditioned and positive-definite, which is necessary for computing the parameters of MBN. For more details on applying regularization techniques to the model learning of continuous EDAs, see [77].

*2) Generating New Solutions:* New candidate solutions to the MOP can be sampled from the probability distribution encoded in the MBN. Probabilistic logic sampling [78], also known as forward sampling, is the method frequently used for sampling Bayesian networks. This method first obtains an ancestral or topological ordering of the network nodes. In this ordering, each node appears after its parent nodes

**Input:** A joint dataset of variables and objectives $A$

$\mathcal{M} \leftarrow$ Regularized ML estimation of MGD from $A$
$g \leftarrow -\infty$
**while** termination criteria are not met **do**
  $\mathcal{S} \leftarrow$ Randomly generate a valid MBN structure
  $b_0 \leftarrow$ Compute the score of $\mathcal{S}$ from $\mathcal{M}$ using (9)
  $t \leftarrow 0$
  **repeat**
    $T \leftarrow$ All possible *valid** edge addition, removal and reversal operations on $\mathcal{S}$
    $S \leftarrow \emptyset$
    $B \leftarrow \emptyset$
    **for all** operations $u \in T$ **do**
      $\mathcal{S}' \leftarrow$ Apply operation $u$ on $\mathcal{S}$
      $b' \leftarrow$ Compute the score of $\mathcal{S}'$ from $\mathcal{M}$ using (9)
      $B \leftarrow B \cup \{b'\}$
      $S \leftarrow S \cup \{\mathcal{S}'\}$
    **end for**
    $t \leftarrow t+1$
    $b_t \leftarrow \max_{b' \in B} b'$
    $\mathcal{S} \leftarrow$ The structure in $S$ with score $b_t$
  **until** $b_t$ is equal to $b_{t-1}$ or termination criteria are met
  **if** $b_t > g$ **then**
    $\mathcal{S}_b \leftarrow \mathcal{S}$
    $g \leftarrow b_t$
  **end if**
**end while**
$\Theta \leftarrow$ Compute the parameters of the conditional probabilities of every node in $\mathcal{S}_b$ from $\mathcal{M}$ // The triple $(\mu_i, \boldsymbol{w}_i, \nu_i)$ for node $i$

**Output:** Best found MBN $(\mathcal{S}_b, \Theta)$

* A valid operation on an MBN structure is the one that respects both the acyclicity condition and the edge restrictions imposed by the bridge subgraph of MBN.

Fig. 4. The greedy MBN learning algorithm

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



**Inputs:**
An MBN $(\mathcal{S}, \Theta)$
Number of new solutions to generate $N$
Problem variables $\boldsymbol{X}$ and their domain $\mathcal{D}$

1 $\boldsymbol{\pi} \leftarrow$ Topological ordering of the nodes in $\mathcal{S}$
2 $A \leftarrow \emptyset$
3 **for** $k \leftarrow 1 \ldots N$ **do**
4 $\quad \boldsymbol{y} \leftarrow \mathbf{0}$
5 $\quad \boldsymbol{\mu} \leftarrow \mathbf{0}$
6 $\quad i \leftarrow 1$
7 $\quad$ **repeat**
8 $\quad\quad B \leftarrow$ Parents of node $\pi_i$ in $\mathcal{S}$ // $\boldsymbol{Pa}(\pi_i)$
9 $\quad\quad \boldsymbol{w} \leftarrow$ Parents weight vector of node $\pi_i$ in $\Theta$
10 $\quad\quad \mu_i \leftarrow$ Mean value of node $\pi_i$ in $\Theta$
11 $\quad\quad m \leftarrow \mu_i + \boldsymbol{w}^T(\boldsymbol{y}_B - \boldsymbol{\mu}_B)$
12 $\quad\quad \nu \leftarrow$ Standard deviation of node $\pi_i$ in $\Theta$
13 $\quad\quad y_i \leftarrow$ Sample a random value from $\mathcal{N}(m, \nu^2)$
14 $\quad\quad i \leftarrow i + 1$
15 $\quad$ **until** all of the nodes in $\boldsymbol{\pi}$ are sampled
16 $\quad \boldsymbol{y}_k \leftarrow \boldsymbol{y}_{\boldsymbol{X}}$ // Remove the sampled objective values
17 $\quad \boldsymbol{y}_k \leftarrow$ Fix the values in $\boldsymbol{y}_k$ which are out of the domain $\mathcal{D}$
18 $\quad A \leftarrow A \cup \{\boldsymbol{y}_k\}$
19 **end for**

**Output:** Set of sampled solutions $A$

Fig. 5. The sampling algorithm used in MBN-EDA for continuous domains.

according to the Bayesian network structure. Consequently, all objective nodes appear before variable nodes in the topological ordering obtained for an MBN, due to the restrictions imposed by the bridge subgraph in model learning. New solutions are generated by sampling the conditional probability distributions estimated for each node in the MBN according to the computed ancestral ordering. Since all parents of node $i$ appear before node $i$ in the ordering, all of its parent nodes will be already sampled at the time of sampling node $i$ and therefore the parameters of the conditional distribution of this node can be computed.

Thanks to the joint modeling of objectives and variables in the MBN that has encoded the dependencies between the variables and objectives, any information about good objective values can be inserted and propagated in the network in the sampling process, increasing the probability of generating variable values that will result in similar objective values. Moreover, the restrictions imposed by the direction of the arcs in the MBN bridge subgraph decrease the number of generated solutions that are inconsistent with the inserted evidence [79].

The approach adopted in this paper treats the objective nodes as normal nodes, and new dummy values (since they are not computed from the objective functions) are generated for these nodes using the probabilities encoded in the MBN. In this way the interactions that are captured during model learning between objectives are also taken into account in the sampling process. When a variable node that has some objective nodes as its parents is being sampled, these dummy objective values are used to compute the parameters of the conditional distribution. The values generated for the objectives are an approximation of the characteristics encoded in the model for the objective values of the selected solutions. Therefore, this method can increase the conformity of the sampled solutions with the learnt MBN. Fig. 5 shows the details of the sampling method adopted in MBN-EDA.

*3) Example:* Consider the MBN structure shown in Fig.1. Such an MBN can be learnt in MBN-EDA for an MOP with five variables and four objectives. For each node of this probabilistic model, the learning algorithm estimates the mean and variance of the conditional probability densities in Equation (2). In addition to these two parameters, a regression coefficient is also estimated for each of the parents of every node (last line of Fig. 4). For example, five parameters are stored in the node corresponding to variable $X_2$: its mean and variance, and the coefficients corresponding to variable $X_1$ and objectives $Q_1$ and $Q_2$.

At the time of sampling, first an ancestral ordering of the nodes is computed, which for example can be: $(Q_1, Q_3, Q_4, Q_2, X_1, X_2, X_4, X_5, X_3)$. To generate a new extended solution, the variable or objective corresponding to each node is sampled in turn according to this ordering. When it is the turn to generate a value for variable $X_2$, the values of variable $X_1$ and objectives $Q_1$ and $Q_2$ are already generated, and thus a new value for variable $X_2$ can be obtained by sampling the Gaussian distribution when replacing the sampled values of the parent nodes in Equation (2) (lines 11–13 of Fig. 5). This procedure is repeated as many times as new offspring solutions are needed.

*4) Discussion:* By estimating a probability distribution from a set of selected solutions, EDAs try to model the regions of the search space that are represented by these solutions. In doing so, these algorithms assume that the selected solutions represent promising regions of the search space, and that further exploration and exploitation of these regions (which in EDAs are interchanged automatically based on the distribution of the solutions) will guide the search toward optimal solutions. Using probabilistic models will allow EDAs to both discover and take advantage of the useful regularities in the set of selected solutions for better optimization.

In a typical EDA, all of the solutions used for estimating the probabilistic model are equally weighted and are treated in the same way. In other words, the quality of solutions are not taken into consideration in the probabilistic modeling of the search space. Instead, it is the density of the selected solutions in the search space that drives model estimation and therefore the algorithm will be completely blind to the quality of the new candidate solutions sampled from the estimated model.

When the quality information of the solutions is integrated into model estimation, it adds another extent to the probabilistic model concerning the regions of the objective space that correspond to the objective values of the selected solutions. Since the solutions are often selected for model learning according to their objective values, the estimated model encodes the best found regions of the objective space. Hence, the new candidate solutions sampled from the joint probabilistic model are more likely to fall in these regions of the objective space, essentially helping the algorithm to find better candidate solutions in each generation. Moreover, isolated solutions with good objective values have a better chance of reproduction when estimating a joint probabilistic model. This is because such a model can encode the relationships between the promising regions in the objective and search spaces, and thus sampling different regions of the search

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



space will be influenced (i.e. controlled) by their approximated qualities.

The employment of similar approaches in a few single-objective EDAs like EBCOA [27], [28] and DEUM [80] is already shown to help achieving better optimization results. However, with the existence of more than one objective in multi-objective optimization, the joint probabilistic model estimation should account for several, possibly conflicting quality values of the solutions. Using an expressive probabilistic model like MBN allows to capture the relationships between different objectives at a manageable complexity when modeling promising regions of the objective space. Therefore MBN-EDA can adaptively combine the distinct quality information available and use them for a more effective generation of new candidate solutions at the time of sampling the estimated probabilistic model. Furthermore, regularized model estimation employed in MBN-EDA helps to discard irrelevant information about the role of, and the relationships among, variables and objectives.

## IV. Experiments

To study the performance of MBN-EDA in multi-objective optimization and to see how the proposed scheme for solution reproduction works, this algorithm is tested on two sets of MOPs, namely the WFG problems and the CEC09 benchmark. The following sections give more details of these MOPs, algorithm implementations, experimental design and the results of applying the algorithms on these MOPs with discussion of the results

### A. WFG Problems

In [81], Huband et al. reviewed many of the available MOP benchmarks in the literature, based on which they proposed a set of MOPs called the walking fish group (WFG) problems. These problems encompass a diverse set of properties that can be found in real-world MOPs and, therefore, raise substantial obstacles for any multi-objective optimization algorithm. Each of the objective functions $f_j$ of an MOP in this benchmark takes the following form

$$\min_{\boldsymbol{z}} \quad f_j(\boldsymbol{z}) = D \cdot z_m + S_j \cdot h_j(z_1, \ldots, z_{m-1}), \qquad (11)$$

where $D$ and $S_j$ are scaling factors and the functions $h_j(\cdot)$ together determine the shape of the Pareto optimal front (e.g., concave, convex, etc.) for that MOP. $\boldsymbol{z}$ is an $m$-dimensional vector obtained by applying a number of transformation functions, like shifting, biasing or reduction, to the $n$-dimensional input solution $\boldsymbol{x} \in \mathcal{D}$ and is composed of two parts: the first $m-1$ parameters, $z_1, \ldots, z_{m-1}$, are obtained from the first $k$ variables of the input solution, and the last parameter ($z_m$) is obtained from the last $l$ variables of the input solution, where $n = k+l$. To simplify the application of transformation functions in the input solution, $k$ is assumed to be a multiple of $m-1$ and $l$ should be an even number.

The number of both objectives and variables can be scaled in this benchmark, which consists of nine MOPs. All WFG problems, except the first three, have a concave Pareto optimal front. WFG1 has a mixed convex-concave optimal front, WFG2 has a disconnected convex front, and WFG3 has a degenerated one-dimensional linear front. For most of these MOPs, the optimal solution of objectives is shifted away from zero to neutralize any bias of the optimization algorithm towards smaller values of the variables. Moreover, in many of the WFG problems, the objective functions are inseparable, requiring the optimization algorithm to consider the relationships between variables.

The number of objectives considered in the experiments with these WFG problems are 3, 5, 7, 10, 15 and 20, whereas the number of variables is set to 16 (with some exceptions). In this way, we will be able to investigate the performance of the algorithms against an increasing number of objectives with an unchanged solution space size.

*1) Implementation Details and Experimental Design:* MBN-EDA is implemented with the help of Matlab toolbox for EDAs (Mateda-2.0) [82], and the implementation of the MBN learning algorithm is adapted from the code provided for GBN learning [83]. Before learning the MBN, training data (extended solutions) are first standardized to have a mean of zero and a standard deviation of one, in order to simplify the learning process by reducing the number of parameters in each node.

To get a better assessment of the optimization performance of MBN-EDA, the results are compared against two other algorithms: a multi-objective evolutionary algorithm (MOEA) and a multi-objective EDA. The MOEA uses simulated binary crossover [84] and polynomial mutation [85] in continuous domains as its genetic operators to generate new solutions, and is used as a standard reference algorithm in many evolutionary multi-objective optimization studies [5], [86]. The multi-objective EDA, namely the regularity-model based multi-objective EDA (RM-MEDA) [13] has been demonstrated to outperform many MOEAs on several benchmark functions. RM-MEDA assumes a certain type of smoothness for the optimal Pareto set and iteratively applies local principal component analysis to build a piece-wise continuous manifold of dimension $m-1$ ($m$ is the number of objectives). This is then used with Gaussian noise to generate new solutions.

The four ranking methods mentioned in Section III-C (Equations (4)–(6) and (8)) are implemented within an individual selector engine which is plugged into each of these algorithms. Since in a black-box optimization scenario, none of the objectives takes precedence over others, equal weights are used for all of the objectives in the weighted sum ranking method ($G_{WS}$). To allow combining the values of different objectives with possibly different ranges, all objective values are normalized before applying a ranking method. In the replacement step of MBN-EDA and MOEA, first an aggregation of both population and offspring solutions is formed and then they are ranked with the same ranking method used for selection to select the best $N$ solutions, where $N$ is the population size. The replacement step of RM-MEDA, on the other hand, does not need any raking of the solutions as the newly generated offspring solutions completely replace the whole population.

Each algorithm is applied with each ranking method separately to each WFG problem with different numbers of

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



objectives. Therefore, there will be $3 \times 4 \times 9 \times 6$ possible combinations in the experiments. The additive epsilon indicator [86], [87] is used to measure the quality of the results obtained by each of the algorithms because of its tractable computational complexity for many-objective problems. This indicator is based on the notion of epsilon efficiency [88], and the corresponding relation of *epsilon dominance* that is defined as

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}, \quad \boldsymbol{x} \preceq_{\epsilon+} \boldsymbol{y} \iff \forall f_i \in \mathcal{F} \quad f_i(\boldsymbol{x}) \leq \epsilon + f_i(\boldsymbol{y}). \tag{12}$$

The additive epsilon indicator between two approximations $A$ and $B$ of the Pareto set is defined as the smallest epsilon value that allows all the solutions in $B$ to be '$\epsilon+$'-dominated by at least one solution in $A$:

$$I_{\epsilon+}(A, B) = \max_{\epsilon \in \mathbb{R}^+} \{A \preceq_{\epsilon+} B\}, \tag{13}$$

where

$$A \preceq_{\epsilon+} B \iff \forall \boldsymbol{y} \in B, \exists \boldsymbol{x} \in A \mid \boldsymbol{x} \preceq_{\epsilon+} \boldsymbol{y}.$$

According to this definition, the additive epsilon indicator for an approximation $A$ of the Pareto set is obtained using a reference set $R$

$$I_{\epsilon+}(A) = I_{\epsilon+}(A, R). \tag{14}$$

This definition implies that smaller values of the epsilon indicator are better. A good choice for the reference set $R$ is an approximation of the Pareto optimal set. However, the size of a good approximation of the Pareto optimal set should increase exponentially with the number of objectives in the MOP to offer a good coverage of the Pareto optimal front. Therefore, this choice of reference set is impractical for many-objective problems. The reference set considered in this paper is composed of the endpoint solutions, obtained by setting one of the objectives to its minimum value and the others to their maximum value, plus the solution representing an approximate compromise between the values of all objectives (e.g., the mean value in the objectives range). The size of this reference set grows only linearly with the number of objectives, and the inclusion of endpoints favors those Pareto set approximations that result in a more scattered Pareto front.

*2) Results:* Fig. 6–8 show the epsilon indicator value obtained for the Pareto set approximations of each of the algorithms, averaged over 20 independent runs. All of the algorithms stop after reaching a maximum number of generations, which is set to 300. The population size is equal for all algorithms and is gradually incremented as the number of objectives increases according to TABLE II. In each generation, $50\%$ of the solutions in the population are selected for reproduction (i.e., $\tau = 0.5$).

Table III shows the statistical analysis of the results for the algorithms with different ranking methods on each of the MOPs with different numbers of objectives. The non-parametric Friedman test [89] is used to check for the statistical differences of the algorithm performance. When the null hypothesis that all the algorithms have an equal average rank is rejected for a specific problem configuration with a p-value less than $0.05$, the entry related to the algorithm with the best Friedman rank is shown in bold. The numbers in parentheses show the results of pairwise comparisons using Bergmann-Hommel's post-hoc test with a significance level of $\alpha = 0.05$. The first number shows how many algorithms are significantly worse than the algorithm listed in this column, and the second number shows how many algorithms are significantly better.

TABLE II
THE POPULATION SIZE USED FOR DIFFERENT NUMBER OF OBJECTIVES AND VARIABLES.

| No. Objectives | 3 | 5 | 7 | 10 | 15 | 20 |
|---|---|---|---|---|---|---|
| No. Variables | 16 | 16 | 16 | 15 | 16 | 21 |
| Population Size | 50 | 100 | 150 | 200 | 250 | 300 |

The objectives in WFG1 are unimodal and biased for specific regions of their input. For this problem, MBN-EDA is able to obtain significantly better Pareto set approximations than the other two algorithms. The performance of the algorithm is very similar when using the different ranking methods tested in these experiments (Fig. 6, left column). Even though there are more interdependencies between the variables in WFG2, MBN-EDA is able to obtain significantly better results for this problem, evidencing the advantage of its probabilistic model for guiding the solution space search. The difference in the optimal front of the MOP problems does not significantly affect MBN-EDA's optimization ability as observed for WFG3, which is very similar to WFG2 except for the shape of the Pareto optimal front. Moreover, approximating the degenerated front in this problem requires good search process exploitation, which, according to the results for this problem, MBN-EDA is better able to do than the other two algorithms.

For WFG4 (Fig. 7, left column), where all of the objectives are multi-modal, the optimization results obtained by MBN-EDA with different ranking methods are significantly better in most of objective space dimensions. For some of the ranking methods, MOEA and MBN-EDA performances are comparable as the number of objectives increase, suggesting the usefulness of genetic operators if there are a large number of optima in a problem. When objective multi-modality is combined with deception, as in WFG5, MBN-EDA performance significantly deteriorates. In fact this algorithm has the worsted optimization performance compared with the other two algorithms for this problem, where it obtains significantly worse Pareto set approximations with all the tested ranking methods, and specially for larger objective space dimensions. A possible explanation for this behavior is that the relationships between deceptive objectives do not provide sufficient information to help the algorithm generate good trade-off solutions in the search space.

The interdependencies between variables in WFG6 are more complex than in the WFG2 and WFG3 problems. Therefore, we find that the choice of the ranking method used in solution selection, which provides the training data for model estimation, will play a major role. In this MOP, the results obtained by MBN-EDA with the $G_{PG}$ and $G_{GD}$ ranking

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



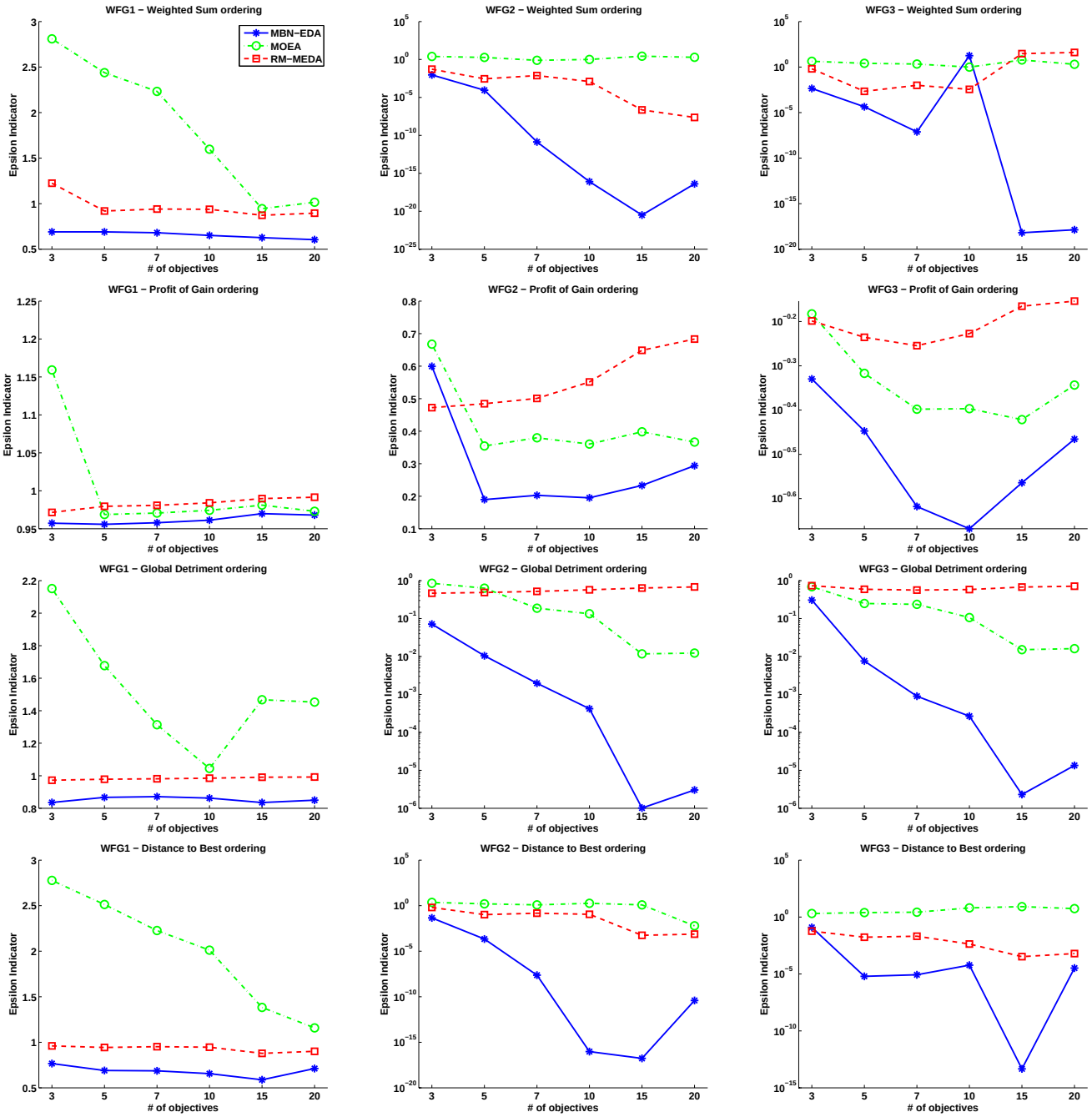



Fig. 6. The average epsilon indicator values for WFG1 (left column), WFG2 (middle column) and WFG3 (right column) problems.

methods are significantly better than for the other algorithms, whereas the results are comparable or significantly worse with the other two ranking methods when the number of objectives is increased. This also shows that the gain function defined in (7) can be a good measure of solutions superiority in this type of MOP.

In WFG7 and WFG8, the optimum value of each variable is biased based on the values of other variables. All of the algorithms find it very difficult to deal with this property of the problem. Again, we see (Fig. 8, left and middle columns) that the choice of ranking method has a significant influence on algorithm performance. With some of the ranking methods (e.g., $G_{PG}$ and $G_{GD}$), MBN-EDA is able to obtain significantly better approximations of the Pareto optimal set for these two problems according to the quality indicator values. The last problem (WFG9) combines many of the properties found in the previous WFG problems. Specifically, apart from variable optimal values being biased, many of the objectives are deceptive as in WFG5. As described previously for WFG5, this prevents MBN-EDA from being able to perform considerably better than the other two algorithms, despite the additional information it collects from data. Nevertheless, the performance of MBN-EDA for this problem is comparable to the other two algorithms, and, with some ranking methods

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



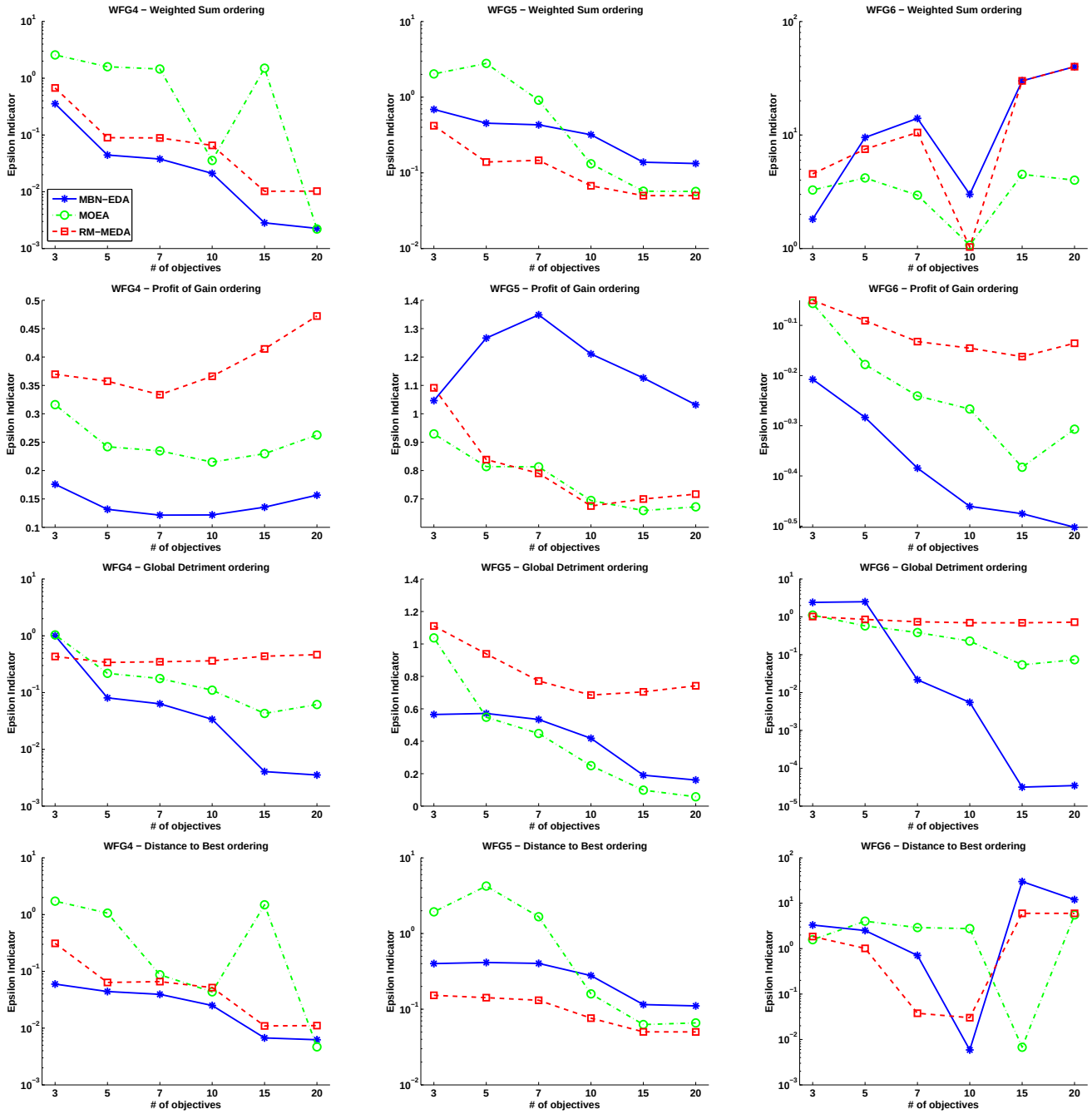



Fig. 7. The average epsilon indicator values for WFG4 (left column), WFG5 (middle column) and WFG6 (right column) problems.

($G_{WS}$ and $G_{DB}$), is significantly better.

In general, the results suggest that there are several factors affecting the optimization performance of the tested algorithms on the selected set of MOP problems. According to the selected quality indicator, MBN-EDA is able to obtain better approximations of the Pareto set than the other two algorithms for many of the tested MOPs featuring different properties and on different objective space dimensions. MBN-EDA finds some specific MOP properties, like deception in the variable values, difficult to deal with.

For some of the tested MOPs, the choice of the ranking method plays a crucial role in algorithm performance and some algorithms tend to be more compatible with specific ranking methods. For example, MBN-EDA performed better than the other algorithms for most MOPs using $G_{<PG>}$ as the ranking method. Since the solution selection mechanism is similar in all algorithms (as they use similar ranking methods), a significant difference in the performance between one algorithm and the others can be attributed to its solution reproduction mechanism. Therefore, the better optimization results for MBN-EDA may be related to model estimation and sampling being better, as they concern both objectives and variables. Moreover, although the choice of probabilistic model in an EDA is important, it should be noted that the

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



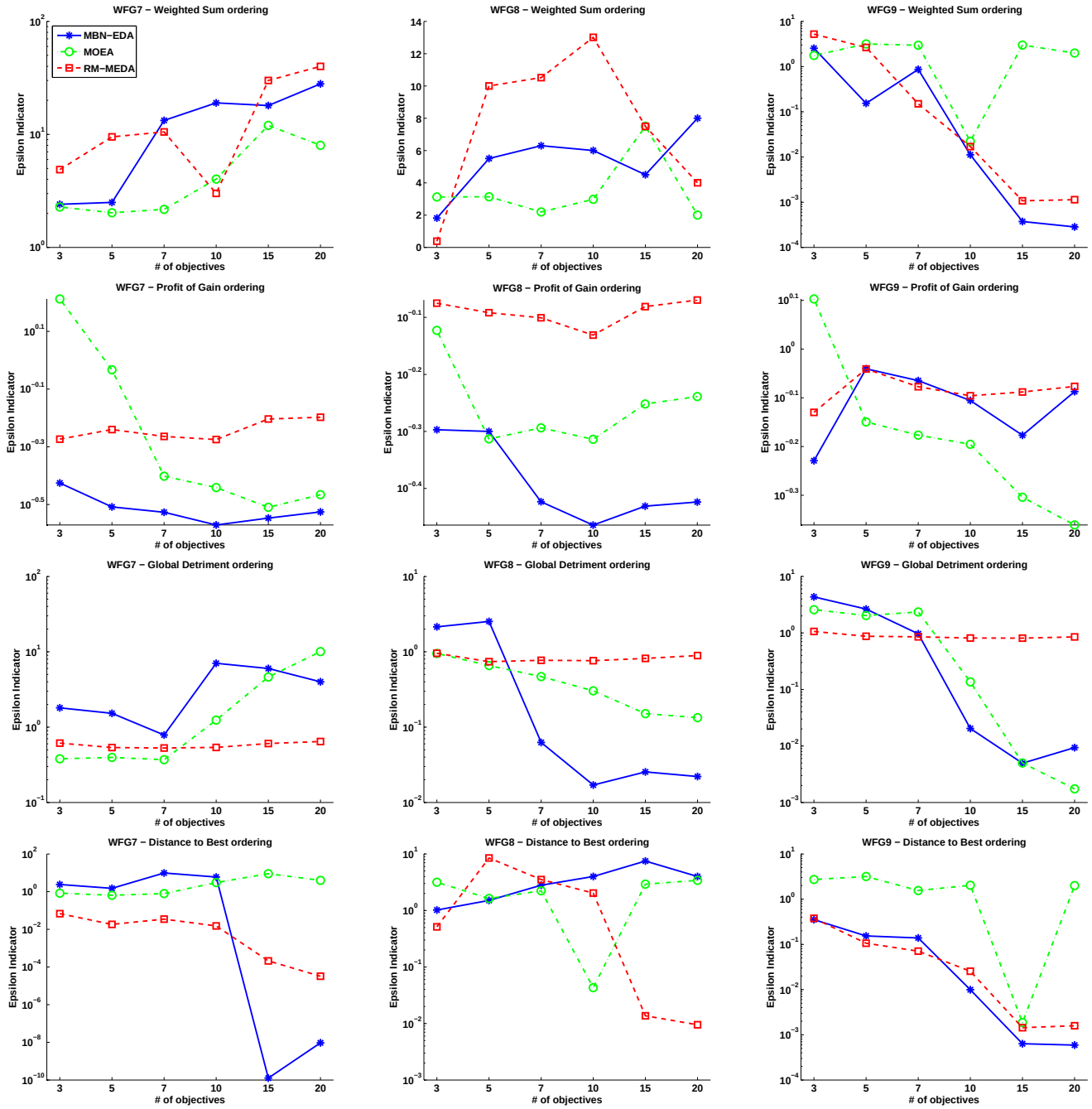



Fig. 8. The average epsilon indicator values for WFG7 (left column), WFG8 (middle column) and WFG9 (right column) problems.

difference between MBN-EDA and RM-MEDA performance is not only due to the difference in their probabilistic models. We have previously shown [24] that the incorporation of objectives into the same probabilistic model can result in significantly better performance.

In some of the problem instances (e.g., WFG6 with $G_{PG}$), with the increase in the objective space dimension, the algorithms seem to obtain better Pareto set approximations that result in lower quality indicator values (meaning better approximations). Note, however, that like the algorithms, the computation of quality indicator values is also affected by the increase in the objective space dimension. In larger objective spaces, the Pareto set approximations obtained by the algorithms will become sparser, as they are using small populations. Also since a small reference set is used to evaluate the algorithms, the differences in the performance of an algorithm in different objective space dimensions will not be clear. However, since an equal reference set is used for each specific objective space dimension, the indicator values can be used to compare the performance of different algorithms in that dimension.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



TABLE III

THE RESULTS OF STATISTICAL DIFFERENCE TESTS FOR DIFFERENT WFG PROBLEMS WITH DIFFERENT NUMBER OF OBJECTIVES AND FOUR DIFFERENT RANKING METHODS. THE BOLD ENTRIES SHOW THE ALGORITHM OBTAINING THE BEST RANKING ACCORDING TO THE STATISTICAL TEST. THE NUMBERS IN THE PARENTHESES SHOWS THE NUMBER OF ALGORITHMS THAT ARE SIGNIFICANTLY WORSE AND BETTER THAN EACH ALGORITHM, RESPECTIVELY, CONSIDERING A 0.05 SIGNIFICANCE LEVEL (REFER TO THE TEXT FOR MORE DISCUSSION OF THE STATISTICAL TEST).

| | | Weighted Sum | | | Profit Gain | | | Global Detriment | | | Distance To Best | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MBN-EDA | MOEA | RM-MEDA | MBN-EDA | MOEA | RM-MEDA | MBN-EDA | MOEA | RM-MEDA | MBN-EDA | MOEA | RM-MEDA |
| WFG1 | 3 | **(2, 0)** | (0, 2) | (1, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 2) | (1, 1) |
| | 5 | **(2, 0)** | (0, 2) | (1, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 2) | (1, 1) |
| | 7 | **(2, 0)** | (0, 2) | (1, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 2) | (1, 1) |
| | 10 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) |
| | 15 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) |
| | 20 | **(1, 0)** | (1, 0) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (0, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |
| WFG2 | 3 | **(2, 0)** | (0, 2) | (1, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 2) | (1, 1) |
| | 5 | **(2, 0)** | (0, 2) | (1, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 7 | **(2, 0)** | (0, 2) | (1, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |
| | 10 | **(2, 0)** | (0, 2) | (1, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) |
| | 15 | **(2, 0)** | (1, 1) | (1, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 20 | **(2, 0)** | (0, 2) | (1, 1) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (1, 1) |
| WFG3 | 3 | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(1, 0)** | (0, 2) | (1, 0) |
| | 5 | **(2, 0)** | (0, 1) | (1, 0) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 7 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 10 | **(2, 0)** | (0, 2) | (1, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 15 | **(2, 0)** | (1, 1) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 20 | **(2, 0)** | (1, 1) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| WFG4 | 3 | **(2, 0)** | (0, 2) | (1, 0) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(1, 0)** | (0, 2) | (1, 0) |
| | 5 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 7 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |
| | 10 | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |
| | 15 | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |
| | 20 | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(1, 0)** | (0, 2) | (1, 1) |
| WFG5 | 3 | (1, 1) | (0, 2) | **(2, 0)** | (0, 0) | (0, 0) | (0, 0) | **(1, 0)** | (1, 0) | (0, 2) | (1, 1) | (0, 2) | **(2, 0)** |
| | 5 | (0, 1) | (0, 1) | **(2, 0)** | (0, 2) | **(1, 0)** | (1, 0) | (1, 0) | **(1, 0)** | (0, 2) | (0, 1) | (0, 1) | **(2, 0)** |
| | 7 | (0, 2) | (1, 1) | **(2, 0)** | (0, 2) | **(1, 0)** | (1, 0) | (1, 0) | **(1, 0)** | (0, 2) | (0, 1) | (0, 1) | **(2, 0)** |
| | 10 | (0, 2) | (1, 1) | **(2, 0)** | (0, 2) | (1, 0) | **(1, 0)** | (1, 1) | **(2, 0)** | (0, 2) | (1, 1) | (1, 1) | **(2, 0)** |
| | 15 | (0, 2) | (1, 1) | **(2, 0)** | (0, 2) | (1, 0) | (1, 0) | (1, 1) | **(2, 0)** | (0, 2) | (0, 2) | (1, 1) | **(2, 0)** |
| | 20 | (0, 2) | (1, 1) | **(2, 0)** | (0, 2) | **(1, 0)** | (1, 0) | (1, 1) | **(2, 0)** | (0, 2) | (1, 1) | (1, 1) | **(2, 0)** |
| WFG6 | 3 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) |
| | 5 | (0, 2) | (1, 0) | **(1, 0)** | **(1, 0)** | (1, 0) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(1, 0)** | (0, 2) | (1, 1) |
| | 7 | (0, 0) | (0, 0) | (0, 0) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 10 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 15 | **(2, 0)** | (0, 1) | (0, 1) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 0) | **(1, 0)** |
| | 20 | (0, 1) | (0, 1) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(1, 0)** | (0, 1) | (0, 0) |
| WFG7 | 3 | (0, 1) | (0, 1) | (0, 1) | (0, 1) | (0, 1) | (0, 1) | **(1, 0)** | (0, 1) | (0, 2) | (0, 2) | (0, 2) | **(1, 0)** |
| | 5 | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) | **(1, 0)** | (0, 2) | (1, 0) |
| | 7 | (0, 0) | (0, 1) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) | (0, 1) | (0, 2) | **(2, 0)** |
| | 10 | (0, 1) | (0, 1) | **(2, 0)** | **(2, 0)** | (1, 1) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(1, 0)** | (0, 2) | (1, 0) |
| | 15 | (0, 0) | (0, 0) | (0, 1) | **(1, 0)** | (1, 0) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 20 | (0, 0) | **(1, 0)** | (0, 1) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 2) | (1, 1) |
| WFG8 | 3 | **(1, 0)** | (0, 2) | (0, 1) | **(1, 0)** | (0, 2) | (0, 1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) |
| | 5 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 2) | (1, 0) | **(1, 0)** | (0, 0) | (0, 1) | **(1, 0)** | (0, 2) | (1, 1) |
| | 7 | **(1, 0)** | (0, 1) | (1, 0) | **(2, 0)** | (0, 0) | (0, 0) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (0, 2) | (1, 1) |
| | 10 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |
| | 15 | (0, 0) | (0, 0) | (0, 0) | **(2, 0)** | (1, 1) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |
| | 20 | (0, 0) | (0, 0) | (0, 0) | **(2, 0)** | (1, 1) | (0, 2) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (1, 1) | (0, 2) |
| WFG9 | 3 | **(1, 0)** | (0, 0) | (0, 1) | **(1, 0)** | (0, 1) | (0, 1) | (0, 0) | (0, 0) | (0, 0) | **(1, 0)** | (0, 2) | (1, 0) |
| | 5 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(1, 0)** | (0, 0) | (0, 1) | **(1, 0)** | (0, 2) | (1, 0) |
| | 7 | **(1, 0)** | (0, 2) | (1, 0) | **(1, 0)** | (0, 0) | (0, 0) | **(1, 0)** | (1, 0) | (0, 2) | (1, 0) | (0, 2) | (1, 1) |
| | 10 | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (0, 1) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |
| | 15 | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (1, 0) | (0, 1) | **(2, 0)** | (1, 1) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |
| | 20 | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (1, 0) | (0, 1) | **(1, 0)** | (1, 0) | (0, 2) | **(2, 0)** | (0, 1) | (0, 1) |

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



TABLE IV
CHARACTERISTICS OF THE UNCONSTRAINED MOPS USED IN CEC09 BENCHMARK.

| Name | No. Objectives | No. Variables | $F^*$ Size |
|---|---|---|---|
| UF1 | 2 | 30 | 1000 |
| UF2 | 2 | 30 | 1000 |
| UF3 | 2 | 30 | 1000 |
| UF4 | 2 | 30 | 1000 |
| UF5 | 2 | 30 | 21 |
| UF6 | 2 | 30 | 1000 |
| UF7 | 2 | 30 | 1000 |
| UF8 | 3 | 30 | 10000 |
| UF9 | 3 | 30 | 10000 |
| UF10 | 3 | 30 | 10000 |
| DTLZ2 | 5 | 30 | 5000 |
| DTLZ3 | 5 | 30 | 5000 |
| WFG1 | 5 | 30 | 5000 |

### B. CEC09 Unconstrained Problems

In this section we compare the performance of the proposed algorithm on the 13 unconstrained problems of the CEC09 benchmark [90] with two other state-of-the-art MOEAs. The first 7 problems of this benchmark are bi-objective, the next three are three-objective and the last three contain five objectives. Two of the five-objective problems are modified versions of DTLZ2 and DTLZ3 problems [15], and the other five-objective problem is the previously studied WFG1. In CEC09 benchmark, solutions of size $n = 30$ are considered for all of these problems (TABLE IV).

*1) Implementation Details and Experimental Design:* The MOEAs used for comparison in this section are the decomposition-based MOEA (MOEA/D) [64] and a hypervolume indicator-based MOEA [59] which we call MOEA-HypE in this paper. In MOEA/D the MOP at hand is decomposed into several single-objective problems using a number of weight vectors and a decomposition method. Here, the Tchebycheff method is used for MOP decomposition in MOEA/D. The choice of weight vectors which should be given in advance to the algorithm is very important and can influence its performance. For two- and three-objective MOPs, we have used the weight vector generation method proposed in the original paper by setting the number of different weight levels to respectively $H = 99$ and $H = 19$, resulting in population sizes of $N = 100$ and $N = 210$, respectively. For five-objective MOPs, the method proposed in [91] is used to generate a well distributed set of weight vectors with a population size of $N = 300$ and a random pool of 15000 weight vectors. The rest of algorithm parameters are set to their default values according to the original paper.

The MOEA-HypE algorithm uses a sample of points in the objective space to obtain an estimation of the hypervolume in the objective space dominated by each solution. These estimated hypervolume values are then used to rank the solutions and select a subset as parents with a tournament selection strategy (with a tournament size of 5) for offspring generation. In the replacement step, population and offspring solutions are aggregated and sorted into a number of non-dominated sets, using the non-dominated sorting algorithm [57]. However, instead of computing the crowding distances of solutions within each non-dominated set, the estimated hypervolume values of the solutions are used to select the elite solutions for forming the population of the next generation. In this study we have used a sample of 10000 points for hypervolume estimation as it is suggested in the original paper. The rest of algorithm parameters are set to their default values.

Both of the MOEAs considered here use the same genetic operators as the ones used by MOEA in Section IV-A1 (simulated binary crossover and polynomial mutation) for generating new solutions. However, MOEA-HypE generates $N/2$ new offspring solutions in each generation, whereas MOEA/D generates $N$ new offspring solutions by evolving all of the solutions in the population. This means that during optimization, MOEA/D processes twice the number of solutions that is searched by MOEA-HypE. Therefore, to have a fair comparison between the algorithms we impose a similar maximum number of fitness evaluations for both of the algorithms by setting the population size of MOEA-HypE to twice the size of population in MOEA/D.

MBN-EDA is tested with three different ranking methods in these experiments. From the ranking methods introduced in Section III-C, we have selected $G_{DB}$ and $G_{GD}$. In order to extend the diversity of solutions when using $G_{DB}$, a tournament selection strategy similar to that used by MOEA-HypE is adopted for this ranking method. In addition to these two ranking methods, MBN-EDA is also tested when using HypE method for ranking and selecting the solutions to be able to compare joint modeling in MBN-EDA with genetic operators in MOEAs. Other parameters of MBN-EDA, like population size, are set equal to those of MOEA-HypE.

The quality indicator suggested in CEC09 benchmark for comparing the results is the inverted generational distance (IGD) [92]. This indicator accounts for the diversity of the approximated Pareto front as well as its convergence to the optimal Pareto front. Given a sample of points $F^*$, well-covering the optimal Pareto front of an MOP, the IGD value for an approximated Pareto front $F$ is computed as:

$$IGD_{F^*}(F) = \frac{\sum_{s \in F^*} \min\{d(s, s'), \forall s' \in F\}}{|F^*|} \tag{15}$$

where $d(\cdot, \cdot)$ gives the Euclidean distance between two points. A smaller value for this indicator means a better approximation. The size of the sample set provided in CEC09 benchmark for the optimal Pareto front of each of the MOPs is shown in TABLE IV.

In addition to this quality indicator, we have also evaluated the approximated fronts with epsilon and hypervolume indicators. The sample of the optimal Pareto front, $F^*$, is used as the reference set when computing the epsilon indicator. The hypervolume of each approximated Pareto front is computed with respect to a point in the objective space which should be worse than all of the points in the approximated fronts, usually referred to as the nadir point. This point is equally set for all of the algorithms to 10000 in all of the objective dimensions and for all of the problems. A higher value for this indicator implies a better approximation.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



*2) Results:* Fig. 9–12 show the results of these indicators for the fronts approximated by each of the algorithms on each of the problems, in 20 independent runs. The maximum number of generations is equally set for all of the algorithms to 300.

The results show that the algorithms have different behaviors on each of the tested MOPs. In general it seems that the problems in CEC09 benchmark are better optimized when using the decomposition method in MOEA/D for guiding the search. With this method, each of the subproblems will guide the search into a different subspace of the objective space, depending on the weight vector which is used for decomposition. Therefore the algorithm is able to search different regions of the objective space even until the final stages of the evolution.

However, for some of the MOPs in this benchmark, like UF1, UF3, UF6 and UF7, where the optimal Pareto set and its corresponding optimal Pareto front have almost similar geometry, the other dominance-based algorithms are able to obtain better approximations of the Pareto optimal front. Especially on problems UF1 and UF7, MOEA/D is outperformed by the HypE method for solution ranking, whether using genetic operators or joint model estimation for generating new solutions. An explanation for this behavior is that using decomposition on some of the MOPs causes the algorithm to miss certain information about the promising areas of the search space, thus reducing the effectiveness of this method.

It can be observed that the overall performance of MBN-EDA with HypE selection method is superior to the performance observed with the other two ranking methods, according to the values of quality indicators. This suggests that the hypervolume indicator estimation provides a better solution ranking than $G_{DB}$ and $G_{GD}$ methods. A closer look at the quality of the approximated fronts along different generations of the evolution has also revealed that HypE is less affected by the specific geometry of the search space. Again, there are some problem instances like UF5, UF6 and UF10 for which $G_{DB}$ provides a better solution ranking than HypE, resulting in better approximate fronts according to the indicator values.

The comparison between MOEA-HypE and MBN-EDA-HypE indicates that for some of the tested MOPs like UF1, UF8, DTLZ2 and WFG1, which cover MOPs with different number of objectives, the non-dominated fronts obtained by MBN-EDA-HypE are better according to the indicator values. Since both of these algorithms are using similar selection methods, this improvement in the results can be directly attributed to the better solution search in MBN-EDA with joint modeling. Our investigation of the populations evolved in MOEA-HypE and MBN-EDA-HypE suggests that joint modeling in MBN-EDA allows the algorithm to rapidly improve its approximated front in early generations of the search, whereas with the genetic operators in MOEA-HypE usually the improvement of the approximated front is slower. However, the diversity of the population in MBN-EDA may not be preserved very well during evolution and the algorithm can enter a stagnation state. On the other hand, the with genetic operators in MOEA-HypE more diverse populations are generated during search and except for some of the MOPs, the algorithm can constantly improve its approximated front.

A point that should be noted here is that when using HypE selection method, the algorithms are directly optimizing the hypervolume indicator. Thus, using the same indicator to evaluate their achieved results may not properly reflect their performances, especially because the hypervolume indicator may overrate certain regions of the approximated fronts [33]. For the algorithms with HypE selection method, the results provided by IGD and epsilon indicators are a better reference of their performance.

## V. MOP Structure Estimation

A major concern of this study is to analyze the MOP structures estimated by MBN-EDA in the course of evolution. These structures are important not only because they can improve optimization by providing information about different types of (in)dependencies existing in the problem (as shown in Section IV-A2), but also because they can give decision makers more control over the selection of the desired information from the Pareto set approximations [93] and better insight into the way different variables influence the objectives or how objectives interact [94]. MBN-EDA's ability to retrieve the MOP structure is tested in different case studies by examining the structures learnt for the WFG1 problem with five objectives and 16 variables, which has an already known MOP structure. To include the factor of different training data for estimating the MOP structure in the analysis, two of the previously introduced ranking methods in Section III-C, i.e., $G_{PG}$ and $G_{DB}$, are used with MBN-EDA in the study.

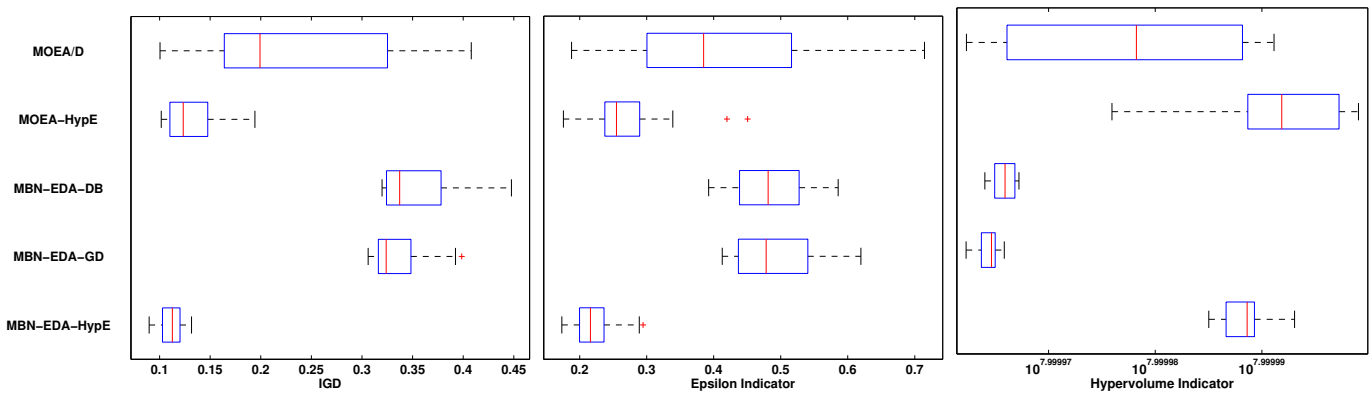
In the first case study, nine irrelevant variables are added to the problem and uniformly distributed among other variables. These variables do not affect the outcome of objective functions in the MOP. Fig. 13 shows the absolute weight of the links encoded in the MBN's bridge substructure between objectives and variables along the evolution path of MBN-EDA. The weights are averaged over 20 independent runs. We found that MBN-EDA is able to clearly distinguish between relevant and irrelevant variables in the studied MOP. The low weight of the links between objectives and irrelevant variables in the estimated MBNs is because either the objectives and these variables have been encoded as conditionally independent of the other variables and objectives or any existing link has been assigned a very small weight, allowing the algorithm to bypass the noise introduced by these variables to the problem. Although the models are learnt from different initial populations in different runs, the structural information encoded between objectives and variables is very similar. It is also shown that the populations selected according to the $G_{PG}$ ranking method help to better distinguish between relevant and irrelevant variables especially in the final generations where the algorithm focuses on specific regions of the search space.

The second case study analyzes the structures learnt for an eight-objective WFG1 problem with three pairs of similar objectives. Fig. 14 compares the absolute weight of the arcs between similar objectives and between dissimilar objective pairs, encoded in the class substructure of MBN in different generations of MBN-EDA. The results are averaged over 20

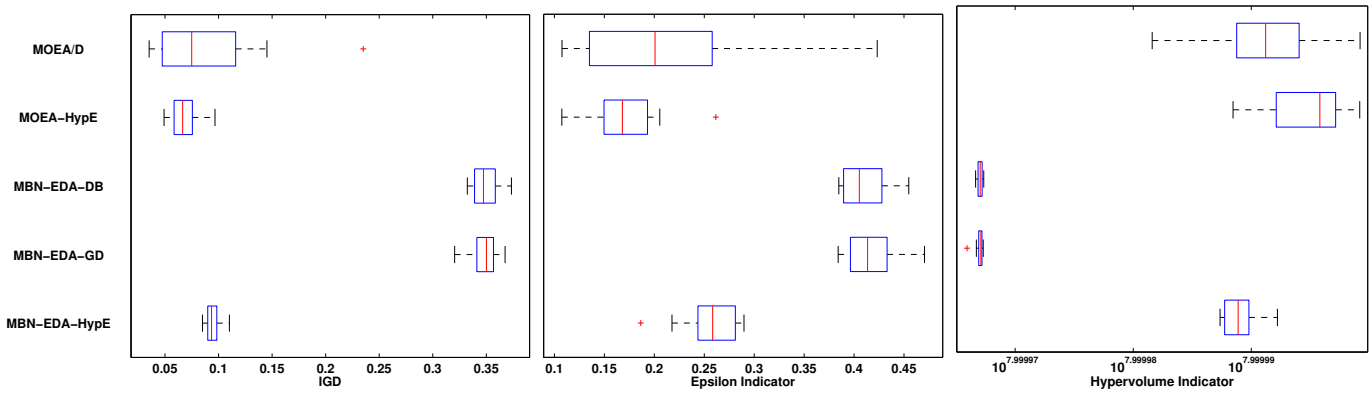
Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

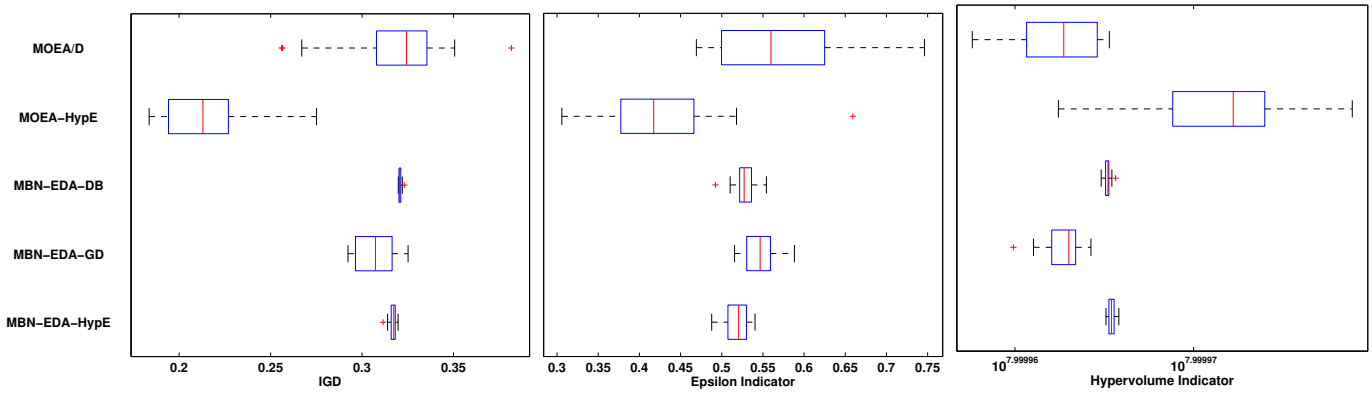




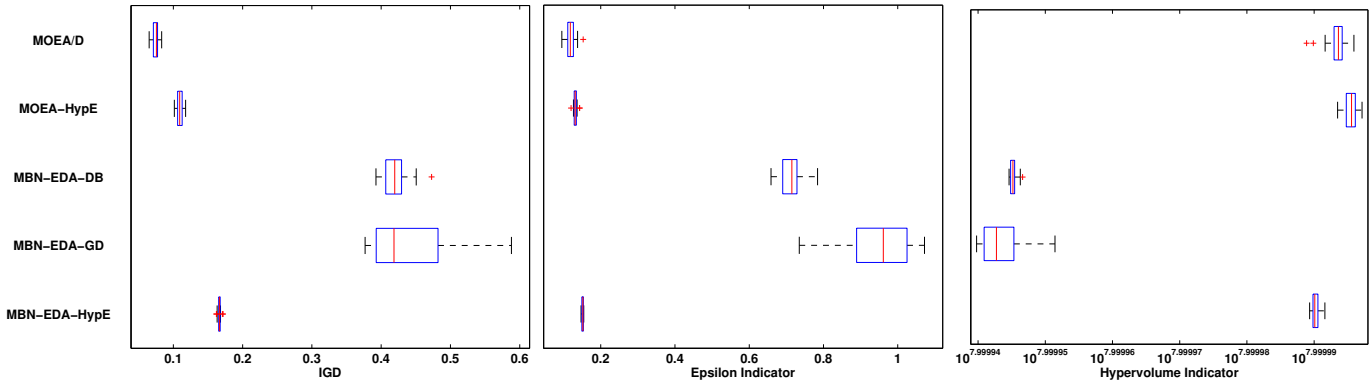

(a) UF1




(b) UF2




(c) UF3



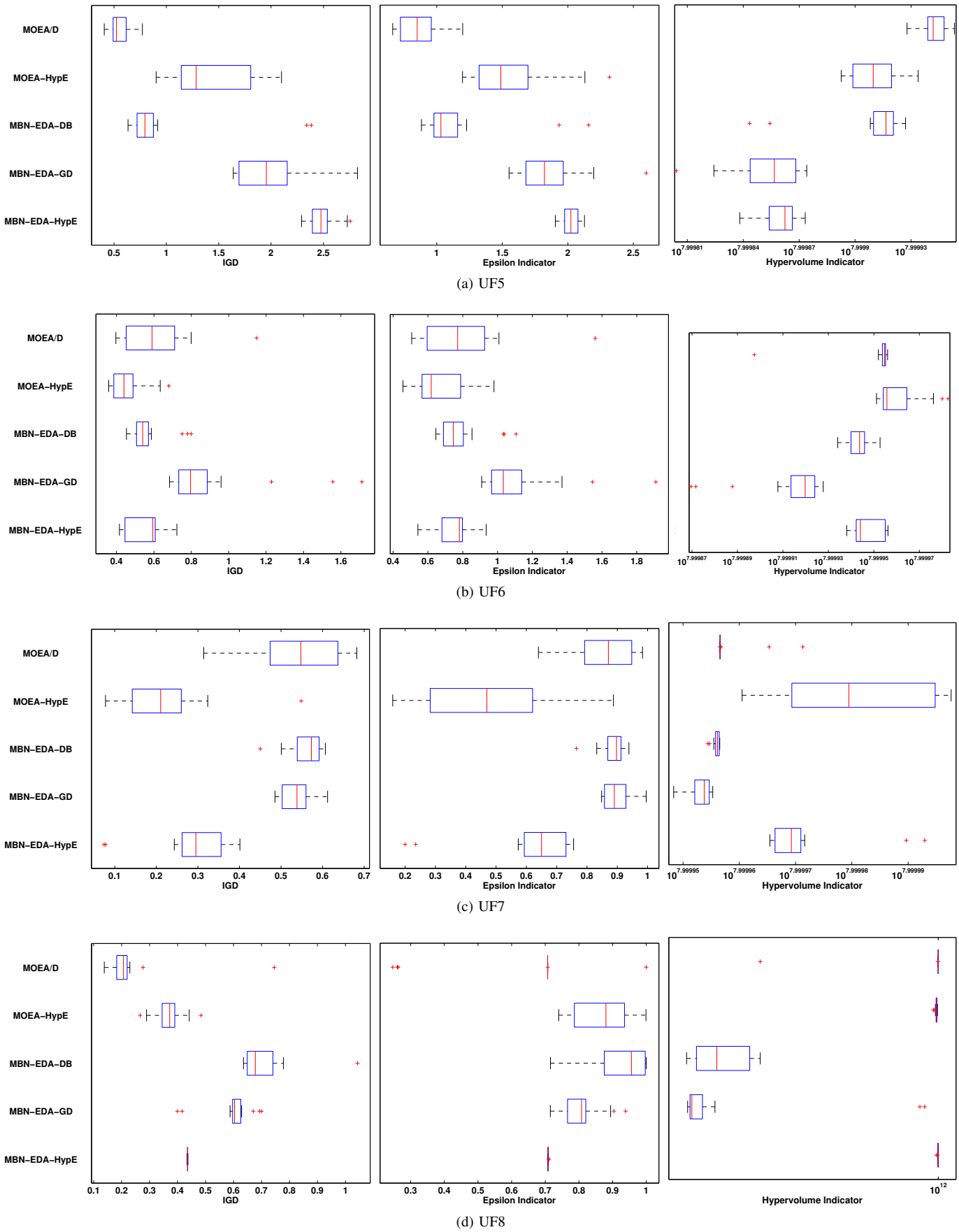

(d) UF4

Fig. 9. The IGD, epsilon and hypervolume indicators values for UF1–UF4 problems in CEC09 benchmark.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.






(a) UF5

(b) UF6

(c) UF7

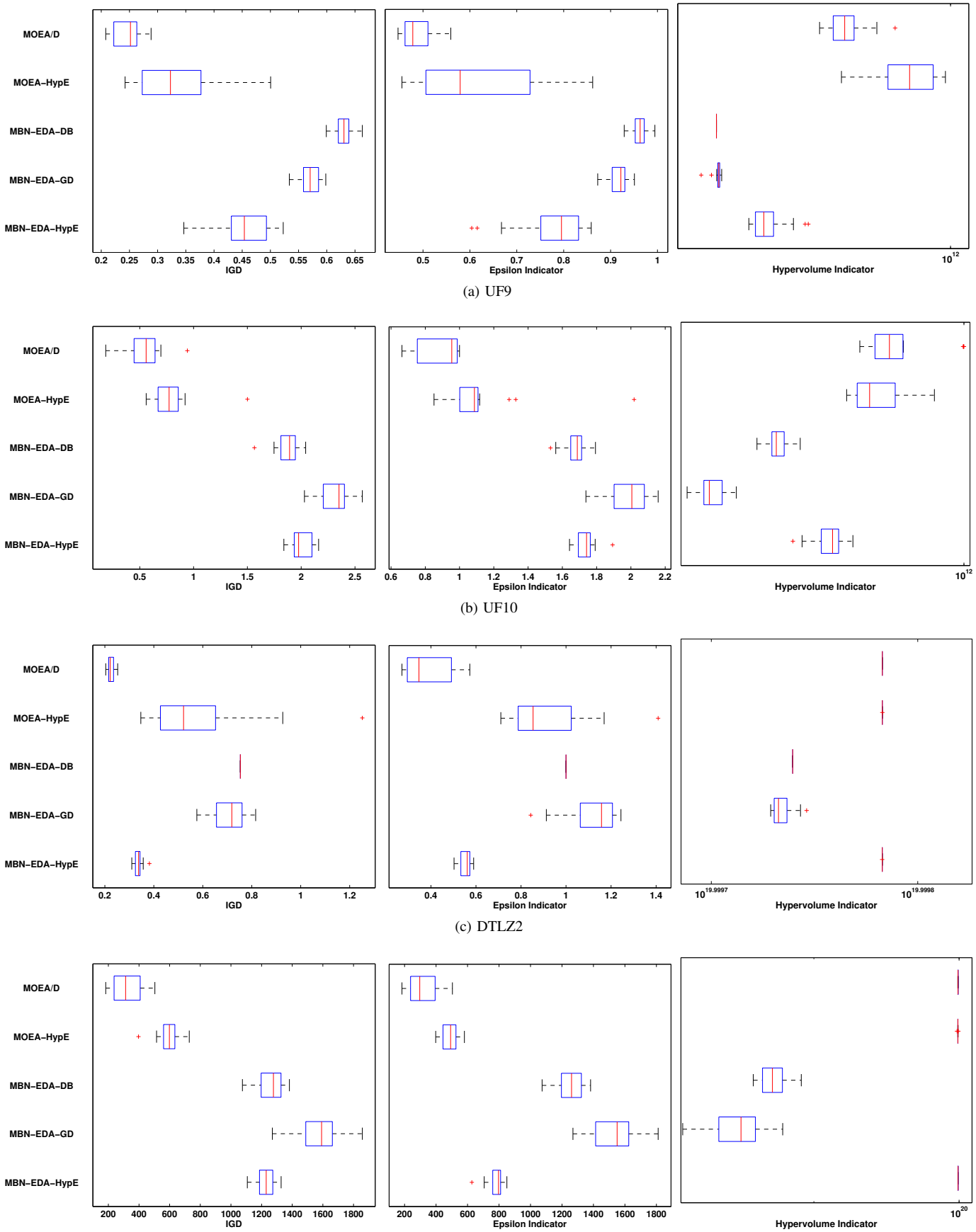
(d) UF8

Fig. 10. The IGD, epsilon and hypervolume indicators values for UF5–UF8 problems in CEC09 benchmark.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.






(a) UF9

(b) UF10

(c) DTLZ2

(d) DTLZ3

Fig. 11. The IGD, epsilon and hypervolume indicators values for UF9, UF10, DTLZ2 and DTLZ3 problems in CEC09 benchmark.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



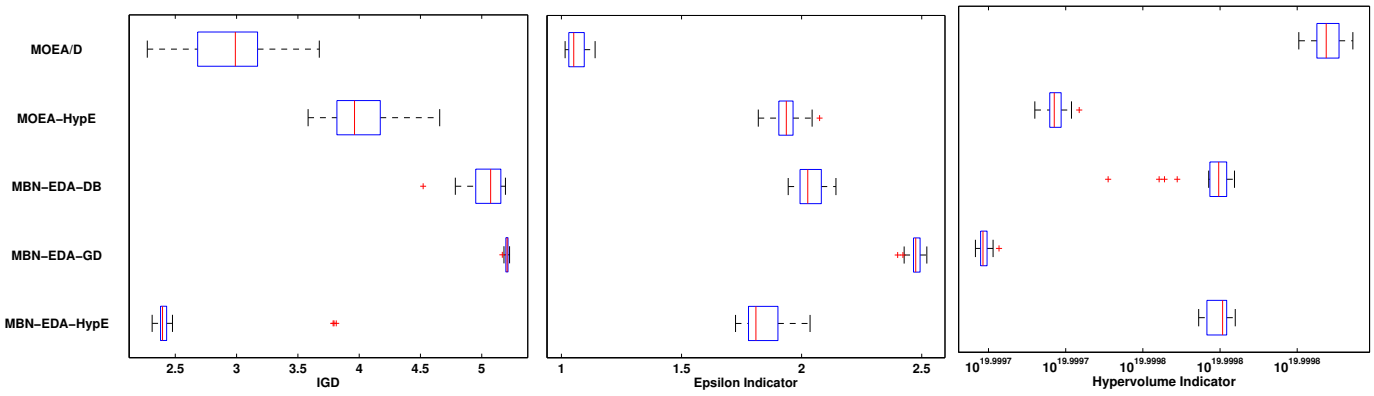



Fig. 12. The IGD, epsilon and hypervolume indicators values for WFG1 problem in CEC09 benchmark.

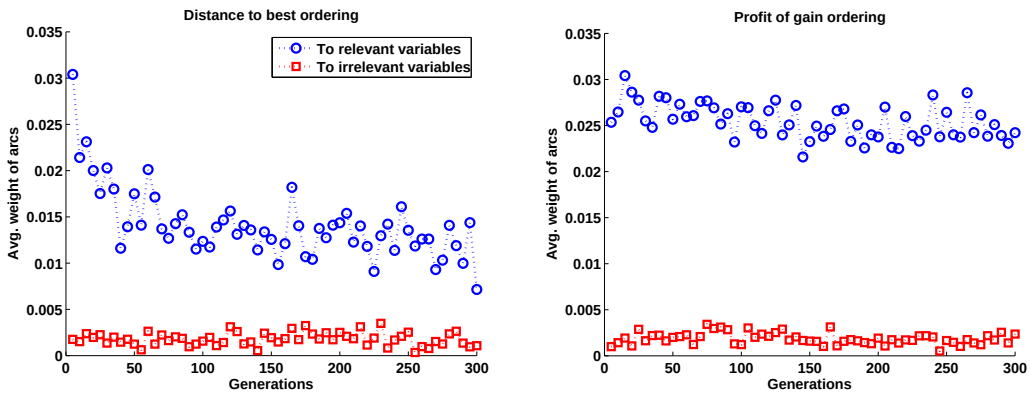



Fig. 13. The average weight of the links from objectives to variables in 5-objective WFG1 problem with irrelevant variables.

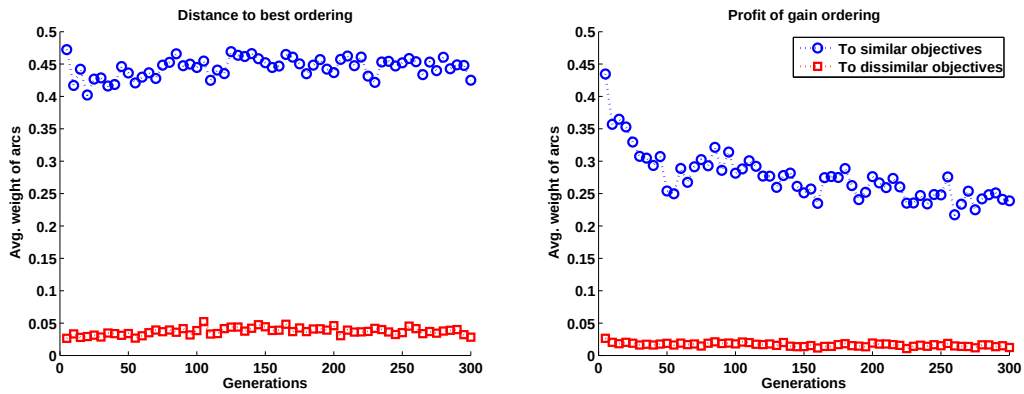



Fig. 14. The average weight of the links between objectives in 8-objective WFG1 problem with three pairs of similar objectives.

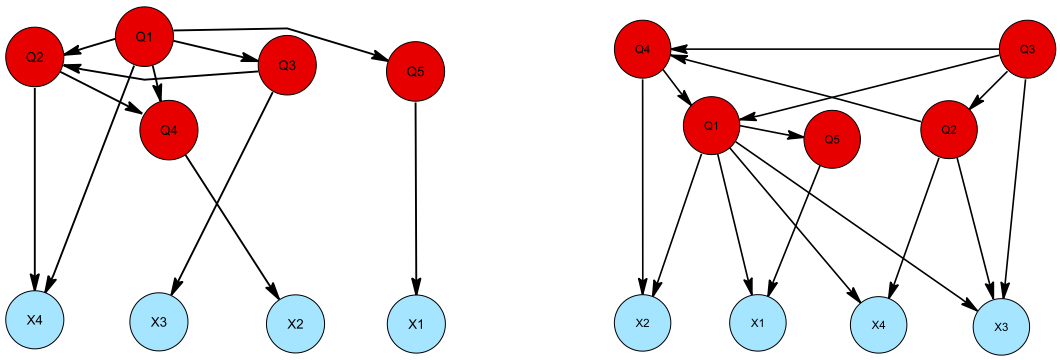
independent runs. The relatively high weights between similar objectives show that MBN-EDA is correctly encoding a strong dependency between these objectives compared with other dependencies in the class subgraph. Note that a closer inspection of the models learnt in different runs with different initial populations has revealed that such a dependency between similar objectives is encoded in the model in all runs, i.e., 100% of the time. We also find that the information about objectives similarity in the MBN class subgraph is better captured from the populations selected according to the $G_{DB}$ ranking method.

Based on the observations from the above two case studies, the third case study directly inspects the structures learnt by MBN-EDA for the WFG1 problem. In the five-objective WFG1 problem considered in the case studies reported in this section, the first $k = 4$ variables determine the position of a given solution in the objective space using different shape functions $h_j$. This is then linearly combined with a distance parameter, obtained from the last $l = 12$ variables. A simplified definition of the five objective functions in this

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.






(a) Distance to best ordering

(b) Profit of gain ordering

Fig. 15. Part of the structure learnt for the 5-objective WFG1 problem showing the most significant arcs and their corresponding nodes.

problem can be given as follows [81]:

$$\begin{aligned} f_1(\boldsymbol{x}) &= a + 2 \cdot h_1\big(g_2(x_1), g_2(x_2), g_2(x_3), g_2(x_4)\big) \\ f_2(\boldsymbol{x}) &= a + 4 \cdot h_2\big(g_2(x_1), g_2(x_2), g_2(x_3), g_2(x_4)\big) \\ f_3(\boldsymbol{x}) &= a + 6 \cdot h_3\big(g_2(x_1), g_2(x_2), g_2(x_3)\big) \\ f_4(\boldsymbol{x}) &= a + 8 \cdot h_4\big(g_2(x_1), g_2(x_2)\big) \\ f_5(\boldsymbol{x}) &= a + 10 \cdot h_5\big(g_2(x_1)\big), \end{aligned} \tag{16}$$

where $a = g_1(x_5, \ldots, x_{16})$, and functions $g_1(\cdot)$ and $g_2(\cdot)$ represent a composition of transformations on the input variables.

Fig. 15 shows part of the structure learnt for this problem, consisting of significant arcs and their corresponding nodes that have an average absolute weight value greater than a threshold set to $w \geq 0.1$ (constituting about $7\%$ of the most significant arcs). While there are many links capturing the obscure (in)dependencies between variables (not depicted here), it is evident that MBN-EDA places more importance on the links between objectives in the class subgraph, and between the objectives and the first four variables in the bridge subgraph. Moreover, these dependencies conform to the function definitions given in (16). For example, the link between objective nodes $Q_2$ and $Q_4$, which is captured using both of the tested ranking methods, is supported by the fact that $h_2$ is a multiplication of $h_4$ and two other factors obtained from variable nodes $X_3$ and $X_4$. Another example is the relationship between objective node $Q_1$ and the four variable nodes, either directly or through the relationship with other objectives, since all four variables influence the value of the first objective.

An important point to note here is the significance of the information provided by the dependencies between objectives and between objectives and variables in multi-objective optimization from MBN-EDA's point of view. There are some studies in the literature that analyze how the dependencies between variables are represented in probabilistic models [95]. But, to the best of our knowledge, the importance of the dependencies involving objectives have not been considered so far in other EDAs used for multi-objective optimization. Such dependencies allow the proposed MBN-EDA to approximate how the variables can affect objective values, which is used to generate new solutions with better objective values.

## VI. Conclusions

The similarity between multi-dimensional classification and multi-objective optimization motivates the use of MBNs in the context of EDAs to solve MOPs. This paper proposes a new modeling approach in multi-objective EDAs that uses MBN estimation to learn a joint model of objectives and variables while at the same time differentiating their role in the network. This model can capture not only the relationships between variables like other EDAs, but also the relationships between variables and objectives, and between objectives. The proposed MBN-EDA is able to deal with many-objective problems by exploiting these new types of relationships encoded in the MBN and implicitly obtaining a decomposition of the MOP, which is used to generate new solutions.

MBN estimation is incorporated into continuous EDAs using Gaussian Bayesian networks where each network node encodes a conditional Gaussian distribution. To obtain a more robust estimation of the model parameters, MBN-EDA employs regularization techniques, previously applied only to single-objective EDAs. This helps the algorithm to obtain a sparser structure, avoiding the effect of possible noise in the data and simplifying many-objective optimization.

The algorithm is tested on two sets of benchmark problems and its results are compared with several state-of-the-art algorithms. The exhaustive experiments of applying MBN-EDA with different ranking methods to the WFG problems with a different number of objectives, show that, according to the epsilon quality indicator and compared with two other algorithms, a standard MOEA and a competitive EDA, this algorithm is able to obtain significantly better approximations of the Pareto set for many of these MOPs, with a significance level of $\alpha = 0.05$. We found that the choice of ranking methods has a major influence on the performance of the algorithms for some of the problems, as they determine the population used for model estimation and offspring reproduction. The results also show that the proposed MBN-EDA was unable to satisfactorily deal with some MOP properties, like deception in the values of the variables. The results of the second set of experiments on the CEC09 unconstrained problems show that on some of the MOPs in this benchmark, the joint modeling in MBN-EDA allows to find considerably better fronts according

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



to three different quality indicators.

The proposed joint model learning approach suggests a way of obtaining the MOP structure that can be used for decision making. An analysis of the structures learnt by MBN-EDA along the evolution path show that the proposed algorithm is able to distinguish between relevant and irrelevant variables, performing a type of variable selection for the objectives encoded in the model. It can also capture stronger dependencies between similar objectives. The analysis of the specific structures learnt for the five-objective WFG1 problem shows that MBN-EDA is able to obtain a very good approximation of this MOP structure and that the information provided by the dependencies between variables and objectives and between objectives, which other EDAs completely overlook, can be very important for multi-objective optimization.

In summary, the key difference between the algorithm proposed in this paper and other MOEAs is in its ability to incorporate objectives information for generating new candidate solutions. By learning a probabilistic model that consists of both variables and objectives the algorithm will not only gain the advantages of multi-objective EDAs over traditional MOEAs, but it is also able to obtain an estimation of the MOP structure discovering the relevant/irrelevant variables for each objective, and the relationships between objectives. This type of new information used in the algorithm is proven to be useful in the optimization of some of the MOPs.

Another point of algorithm strength is its use of a multivariate probabilistic model, namely a type of Bayesian network, which gives it an advantage over other multi-objective EDAs that use simpler type of models. The algorithm also uses a regularization method to improve estimation of probabilistic model parameters. Similar to other Bayesian network-based multi-objective EDAs, the use of such probabilistic model comes at a cost. Estimating a Bayesian network from a dataset is usually time-consuming and thus the running time of the algorithm also increases greatly in comparison to other MOEAs. Because of this complexity, the algorithm can be considered for application on MOPs as a higher level algorithm, when simpler algorithms fail to obtain good solutions, if more insight to the MOP structure is required or if the decision maker is not only interested in a set of solutions for the MOP.

There are many ways to extend this work. This new modeling method provides a promising platform for the experts or decision makers to incorporate preference information [96], [97] into the model as conditional (in)dependency relations between objectives and variables, as well as preferable values for some objectives. The dependencies learnt between objectives in the MOP structure can be used to analyze relationships like conflict or redundancy between sets of objectives. Another interesting study is to integrate problem decomposition methods and joint modeling in MBN-EDA and compare its performance with the current decomposition-based MOEAs. The application of MBN-EDA to real-world problems with unknown structures and to check how the captured relationships meet decision-maker expectations are also potential future areas of research.

## References


[1] I. Sbalzarini, S. Mueller, and P. Koumoutsakos, "Multiobjective optimization using evolutionary algorithms," in *Summer Program of Studying Turbulence Using Numerical Simulation Databases–VIII*, vol. 1. Center for Turbulence Research, 2000, pp. 63–74.

[2] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[3] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[4] A. Abraham, L. Jain, and R. Goldberg, Eds., *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, ser. Advanced Information and Knowledge Processing. Berlin: Springer, 2005.

[5] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed., ser. Genetic and Evolutionary Computation Series. Springer, 2007.

[6] D. E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[7] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[8] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds., *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, ser. Studies in Fuzziness and Soft Computing. Secaucus, NJ, USA: Springer, 2006.

[9] M. Pelikan, K. Sastry, and E. Cantú-Paz, Eds., *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, ser. Studies in Computational Intelligence. Secaucus, NJ, USA: Springer, 2006.

[10] P. Larrañaga, H. Karshenas, C. Bielza, and R. Santana, "A review on probabilistic graphical models in evolutionary computation," *Journal of Heuristics*, vol. 18, no. 5, pp. 795–819, 2012.

[11] D. Thierens and P. A. N. Bosman, "Multi-objective mixture-based iterated density estimation evolutionary algorithms," in *Conference on Genetic and Evolutionary Computation (GECCO '01)*, L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, Eds. San Francisco, California, USA: Morgan Kaufmann, 2001, pp. 663–670.

[12] M. Pelikan, K. Sastry, and D. Goldberg, "Multiobjective estimation of distribution algorithms," in *Scalable Optimization via Probabilistic Modeling*, ser. Studies in Computational Intelligence, M. Pelikan, K. Sastry, and E. Cantú-Paz, Eds. Berlin: Springer, 2006, vol. 33, pp. 223–248.

[13] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[14] L. Martí, J. Garcia, A. Berlanga, C. A. Coello Coello, and J. M. Molina, "On current model-building methods for multi-objective estimation of distribution algorithms: Shortcommings and directions for improvement," Department of Informatics, Universidad Carlos III de Madrid, Madrid, Spain, Tech. Rep. GIAA2010E001, 2010.

[15] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *IEEE Congress on Evolutionary Computation (CEC '02)*, vol. 1, 2002, pp. 825–830.

[16] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima, "Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems," in *10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08)*. New York, NY, USA: ACM, 2008, pp. 649–656.

[17] T. Goel, R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker, "Response surface approximation of Pareto optimal front in multi-objective optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 4–6, pp. 879–893, 2007.

[18] A. López Jaimes, C. A. Coello Coello, and D. Chakraborty, "Objective reduction using a feature selection technique," in *10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08)*. New York, NY, USA: ACM, 2008, pp. 673–680.

[19] D. Craft and T. Bortfeld, "How many plans are needed in an IMRT multi-objective plan database?" *Physics in Medicine and Biology*, vol. 53, no. 11, pp. 2785–2796, 2008.

[20] S. Verel, A. Liefooghe, L. Jourdan, and C. Dhaenens, "Analyzing the effect of objective correlation on the efficient set of MNK-landscapes," in *5th International Conference on Learning and Intelligent Optimization (LION'5)*, ser. Lecture Notes in Computer Science, C. A. Coello Coello, Ed., vol. 6683. Berlin, Heidelberg: Springer, 2011, pp. 116–130.


Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.




[21] D. Saxena, J. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2013.

[22] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: Theory and applications," *Evolutionary Computation*, vol. 17, no. 2, pp. 135–166, 2009.

[23] G. Guillén-Gosálbez, "A novel MILP-based objective reduction method for multi-objective optimization: Application to environmental problems," *Computers & Chemical Engineering*, vol. 35, no. 8, pp. 1469–1477, 2011.

[24] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Multi-objective optimization with joint probabilistic modeling of objectives and variables," in *Evolutionary Multi-Criterion Optimization (EMO'11)*, ser. Lecture Notes in Computer Science, R. Takahashi, K. Deb, E. Wanner, and S. Greco, Eds., vol. 6576. Berlin, Heidelberg: Springer, 2011, pp. 298–312.

[25] P. de Waal and L. van der Gaag, "Inference and learning in multi-dimensional Bayesian network classifiers," *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pp. 501–511, 2007.

[26] C. Bielza, G. Li, and P. Larrañaga, "Multi-dimensional classification with Bayesian networks," *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 705–727, 2011.

[27] T. Miquélez, E. Bengoetxea, and P. Larrañaga, "Evolutionary computation based on Bayesian classifiers," *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 3, pp. 335–350, 2004.

[28] ——, "Evolutionary Bayesian classifier-based optimization in continuous domains," in *6th International Conference on Simulated Evolution and Learning (SEAL '06)*, ser. Lecture Notes in Computer Science, T.-D. Wang, X. Li, S.-H. Chen, X. Wang, H. A. Abbass, H. Iba, G. Chen, and X. Yao, Eds., vol. 4247. Springer, 2006, pp. 529–536.

[29] J. Pearl, "Bayesian networks: A model of self-activated memory for evidential reasoning," in *7th Conference of the Cognitive Science Society*, 1985, pp. 329–334.

[30] S. L. Lauritzen, "Propagation of probabilities, means, and variances in mixed graphical association models," *Journal of the American Statistical Association*, vol. 87, no. 420, pp. 1098–1108, 1992.

[31] D. Geiger and D. Heckerman, "Learning Gaussian networks," in *10th Conference on Uncertainty in Artificial Intelligence (UAI'94)*, R. L. de Mantaras and D. Poole, Eds. San Francisco, CA: Morgan Kaufmann, 1994, pp. 235–243.

[32] D. Goldberg, "The design of innovation: Lessons from genetic algorithms, lessons for the real world," *Technological Forecasting and Social Change*, vol. 64, no. 1, pp. 7–12, 2000.

[33] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[34] J. Martín, C. Bielza, and D. R. Insua, "Approximating nondominated sets in continuous multiobjective optimization problems," *Naval Research Logistics*, vol. 52, no. 5, pp. 469–480, 2005.

[35] J. Schwarz and J. Očenášek, "Multiobjective Bayesian optimization algorithm for combinatorial problems: Theory and practice," *Neural Network World*, vol. 11, no. 5, pp. 423–442, 2001.

[36] N. Khan, D. E. Goldberg, and M. Pelikan, "Multiple-objective Bayesian optimization algorithm," in *Conference on Genetic and Evolutionary Computation (GECCO '02)*, W. B. Langdon, E. Cantú-Paz, K. E. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. K. Burke, and N. Jonoska, Eds. Morgan Kaufmann, 2002, p. 684.

[37] P. A. Bosman and D. Thierens, "Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms," *International Journal of Approximate Reasoning*, vol. 31, no. 3, pp. 259–289, 2002.

[38] P. A. N. Bosman and D. Thierens, "Adaptive variance scaling in continuous multi-objective estimation of distribution algorithms," in *9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*, H. Lipson, Ed. New York, NY, USA: ACM, 2007, pp. 500–507.

[39] M. Costa and E. Minisci, "MOPED: A multi-objective Parzen-based estimation of distribution algorithm for continuous problems," in *Second International Conference on Evolutionary Multi-Criterion Optimization (EMO'03)*, ser. Lecture Notes in Computer Science, C. Fonseca, P. Fleming, E. Zitzler, L. Thiele, and K. Deb, Eds., vol. 2632. Berlin: Springer, 2003, pp. 282–294.

[40] T. Okabe, Y. Jin, B. Sendoff, and M. Olhofer, "Voronoi-based estimation of distribution algorithm for multi-objective optimization," in *IEEE Congress on Evolutionary Computation (CEC'04)*, vol. 2, 2004, pp. 1594–1601.

[41] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multiobjective hBOA, clustering, and scalability," in *Conference on Genetic and Evolutionary Computation (GECCO '05)*. New York, NY, USA: ACM, 2005, pp. 663–670.

[42] K. Sastry, D. Goldberg, and M. Pelikan, "Limits of scalability of multiobjective estimation of distribution algorithms," in *IEEE Congress on Evolutionary Computation (CEC'05)*, vol. 3, 2005, pp. 2217–2224.

[43] H. Soh and M. Kirley, "moPGA: Towards a new generation of multi-objective genetic algorithms," in *IEEE Congress on Evolutionary Computation (CEC'06)*, 2006, pp. 1702–1709.

[44] C. Ahn and R. Ramakrishna, "A diversity preserving selection in multiobjective evolutionary algorithms," *Applied Intelligence*, vol. 32, no. 3, pp. 231–248, 2010.

[45] H. Tang, V. A. Shim, K. C. Tan, and J. Y. Chia, "Restricted Boltzmann machine based algorithm for multi-objective optimization," in *IEEE Congress on Evolutionary Computation (CEC'10)*, 2010, pp. 1–8.

[46] L. Martí, J. García, A. Berlanga, C. A. Coello Coello, and J. M. Molina, "MB-GNG: Addressing drawbacks in multi-objective optimization estimation of distribution algorithms," *Operations Research Letters*, vol. 39, no. 2, pp. 150–154, 2011.

[47] L. Martí, J. García, A. Berlanga, and J. Molina, "Multi-objective optimization with an adaptive resonance theory-based estimation of distribution algorithm," *Annals of Mathematics and Artificial Intelligence*, 2012.

[48] M. Laumanns and J. Očenášek, "Bayesian optimization algorithms for multi-objective optimization," in *7th International Conference on Parallel Problem Solving from Nature (PPSN VII)*, ser. Lecture Notes in Computer Science, J. J. M. Guervós, P. Adamidis, H.-G. Beyer, J. L. F.-V. Martín, and H.-P. Schwefel, Eds., vol. 2439. Springer, 2002, pp. 298–307.

[49] X. Zhong and W. Li, "A decision-tree-based multi-objective estimation of distribution algorithm," in *International Conference on Computational Intelligence and Security (CIS'07)*. IEEE Computer Society, 2007, pp. 114–118.

[50] Y. Katsumata and T. Terano, "Bayesian optimization algorithm for multi-objective solutions: Application to electric equipment configuration problems in a power plant," in *IEEE Congress on Evolutionary Computation (CEC '03)*, vol. 2, 2003, pp. 1101–1107.

[51] H. Li, Q. Zhang, E. Tsang, and J. A. Ford, "Hybrid estimation of distribution algorithm for multiobjective knapsack problem," in *Fourth European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP'04)*, ser. Lecture Notes in Computer Science, J. Gottlieb and G. R. Raidl, Eds., vol. 3004. Berlin, Heidelberg: Springer, 2004, pp. 145–154.

[52] Y. Gao, X. Hu, H. Liu, and Y. Feng, "Multiobjective estimation of distribution algorithm combined with PSO for RFID network optimization," in *International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, vol. 2. IEEE Computer Society, 2010, pp. 736–739.

[53] V. A. Shim, K. C. Tan, and J. Y. Chia, "Multi-objective optimization with estimation of distribution algorithm in a noisy environment," *Evolutionary Computation*, vol. 21, no. 1, pp. 149–177, 2013.

[54] V. A. Shim, K. C. Tan, and K. K. Tan, "A hybrid adaptive evolutionary algorithm in the domination-based and decomposition-based frameworks of multi-objective optimization," in *IEEE Congress on Evolutionary Computation (CEC'12)*, 2012, pp. 1–8.

[55] V. A. Shim, K. C. Tan, and C. Y. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 5, pp. 682–691, 2012.

[56] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimization algorithm," in *Conference on Genetic and Evolutionary Computation (GECCO '99)*, W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., vol. 1. Orlando, Florida, USA: Morgan Kaufmann Publishers, 1999, pp. 525–532.

[57] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.


Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.




[58] C. W. Ahn and R. S. Ramakrishna, "Multiobjective real-coded Bayesian optimization algorithm revisited: Diversity preservation," in *9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*. New York, NY, USA: ACM, 2007, pp. 593–600.

[59] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[60] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2010.

[61] E. Hughes, "Evolutionary many-objective optimisation: Many once or one many?" in *IEEE Congress on Evolutionary Computation (CEC '05)*, vol. 1, 2005, pp. 222–227.

[62] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *IEEE Congress on Evolutionary Computation (CEC '08) – IEEE World Congress on Computational Intelligence (WCCI '08)*, 2008, pp. 2419–2426.

[63] H. Aguirre and K. Tanaka, "A study on the effects of rankings sensitive to density on many-objective MNK landscapes," in *IEEE Congress on Evolutionary Computation (CEC'10), IEEE World Congress on Computational Intelligence (WCCI 2010)*, 2010, pp. 1089–1096.

[64] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[65] M. Garza-Fabre, G. Toscano Pulido, and C. Coello Coello, "Ranking methods for many-objective optimization," in *MICAI 2009: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, A. Aguirre, R. Borja, and C. García, Eds. Berlin: Springer, 2009, vol. 5845, pp. 633–645.

[66] ——, "Alternative fitness assignment methods for many-objective optimization problems," in *Artifical Evolution*, ser. Lecture Notes in Computer Science, P. Collet, N. Monmarché, P. Legrand, M. Schoenauer, and E. Lutton, Eds. Berlin: Springer, 2010, vol. 5975, pp. 146–157.

[67] M. Garza-Fabre, G. Toscano Pulido, and C. A. Coello Coello, "Two novel approaches for many-objective optimization," in *IEEE Congress on Evolutionary Computation (CEC'10) – IEEE World Congress on Computational Intelligence (WCCI 2010)*, 2010, pp. 4480–4487.

[68] G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.

[69] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, 1995.

[70] W. Buntine, "A guide to the literature on learning probabilistic networks from data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 2, pp. 195–210, 1996.

[71] M. Pelikan, *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms*, 1st ed., ser. Studies in Fuzziness and Soft Computing. Springer, 2005, vol. 170.

[72] W. Buntine, "Theory refinement on Bayesian networks," in *7th Annual Conference on Uncertainty in Artificial Intelligence (UAI '91)*, B. D'Ambrosio and P. Smets, Eds. San Francisco, CA, USA: Morgan Kaufmann, 1991, pp. 52–60.

[73] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[74] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[75] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., ser. Springer Series in Statistics. New York: Springer, 2009.

[76] J. Schäfer and K. Strimmer, "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics," *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, online, DOI: 10.2202/1544–6115.1175, 2005.

[77] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Regularized continuous estimation of distribution algorithms," *Applied Soft Computing*, vol. 13, no. 5, pp. 2412–2432, 2013.

[78] M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling," in *Second Annual Conference on Uncertainty in Artificial Intelligence (UAI '86)*, J. F. Lemmer and L. N. Kanal, Eds., vol. 2. Elsevier, 1986, pp. 149–163.

[79] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, ser. Adaptive Computation and Machine Learning. The MIT Press, 2009.

[80] S. Shakya and J. McCall, "Optimization by estimation of distribution with DEUM framework based on Markov random fields," *International Journal of Automation and Computing*, vol. 4, pp. 262–272, 2007.

[81] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[82] R. Santana, C. Bielza, P. Larrañaga, J. A. Lozano, C. Echegoyen, A. Mendiburu, R. Armañanzas, and S. Shakya, "Mateda-2.0: Estimation of distribution algorithms in MATLAB," *Journal of Statistical Software*, vol. 35, no. 7, pp. 1–30, 2010.

[83] M. Schmidt, A. Niculescu-Mizil, and K. Murphy, "Learning graphical model structure using L1-regularization paths," in *22nd National Conference on Artificial Intelligence (AAAI '07)*, vol. 2. AAAI Press, 2007, pp. 1278–1283.

[84] K. Deb and B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[85] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[86] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[87] E. Zitzler, J. Knowles, and L. Thiele, "Quality assessment of Pareto set approximations," in *Multiobjective Optimization*, ser. Lecture Notes in Computer Science, J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds. Berlin, Heidelberg: Springer, 2008, vol. 5252, pp. 373–404.

[88] S. Helbig and D. Pateva, "On several concepts for $\epsilon$-efficiency," *OR Spectrum*, vol. 16, pp. 179–186, 1994.

[89] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[90] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," The Shool of Computer Science and Electronic Engineering, University of Essex, UK, Tech. Rep. CES-487, 2009.

[91] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *IEEE Congress on Evolutionary Computation (CEC'09)*, 2009, pp. 203–208.

[92] C. A. Coello Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, pp. 163–190, 2005.

[93] T. Ulrich, D. Brockhoff, and E. Zitzler, "Pattern identification in Pareto-set approximations," in *10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08)*. New York, NY, USA: ACM, 2008, pp. 737–744.

[94] F. Chan and S. Chung, "Multi-criteria genetic optimization for distribution network problems," *The International Journal of Advanced Manufacturing Technology*, vol. 24, pp. 517–532, 2004.

[95] R. Santana, C. Bielza, J. A. Lozano, and P. Larrañaga, "Mining probabilistic models learned by EDAs in the optimization of multi-objective problems," in *11th Annual Conference on Genetic and Evolutionary Computation (GECCO '09)*. ACM, 2009, pp. 445–452.

[96] F. di Pierro, S.-T. Khu, and D. Savic, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 17–45, 2007.

[97] C. Hirsch, P. Shukla, and H. Schmeck, "Variable preference modeling using multi-objective evolutionary algorithms," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, R. Takahashi, K. Deb, E. Wanner, and S. Greco, Eds., vol. 6576. Berlin, Heidelberg: Springer, 2011, pp. 91–105.


Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

**Hossein Karshenas** received his B.E. degree in computer engineering from Shahid Beheshti University, Tehran, Iran in 2006, and the M.E. degree in artificial intelligence and robotics from Iran University of Science and Technology (IUST), Tehran, Iran in 2009.

He is currently a Ph.D. Student at the School of Computer Science, Technical University of Madrid, where he works on evolutionary algorithms based on probabilistic modeling for optimization. His research interests include estimation of distribution algorithms, probabilistic graphical models, evolutionary algorithms, continuous and multi-objective optimization and machine learning.

**Roberto Santana** received his BS degree in Computer Science and the PhD degree in Mathematics from the University of Havana, Cuba in 1996 and 2005, respectively; and the PhD degree in Computer Science from the University of the Basque Country in 2006.

He is a researcher at the University of the Basque Country (UPV/EHU). His main research interests are evolutionary computation, probabilistic graphical models, neuroscience and bioinformatics.

**Concha Bielza** received the M.S. degree in Mathematics from Complutense University of Madrid, Madrid, Spain, in 1989 and the Ph.D. degree in Computer Science from the Universidad Politcnica de Madrid, Madrid, in 1996 (extraordinary doctorate award).

She is currently a Full Professor of Statistics and Operations Research with the Departamento de Inteligencia Artificial, Universidad Politcnica de Madrid. Her research interests are primarily in the areas of probabilistic graphical models, decision analysis, metaheuristics for optimization, data mining, classification models, and real applications, like biomedicine, bioinformatics and neuroscience.

**Pedro Larrañaga** received the MSc degree in mathematics (statistics) from the University of Valladolid and the PhD degree in computer science from the University of the Basque Country (excellence award).

He is a Full Professor in Computer Science and Artificial Intelligence at the Technical University of Madrid (UPM) since 2007. Before moving to UPM, his academic career has been developed at the University of the Basque Country (UPV-EHU) at several faculty ranks: Assistant Professor (1985-1998), Associate Professor (1998-2004) and Full Professor (2004-2007). He earned the habilitation qualification for Full Professor in 2003.

Copyright (c) 2013 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.