

# Multi-objective Optimization with Joint Probabilistic Modeling of Objectives and Variables

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

Computational Intelligence Group, School of Computer Science,  
Technical University of Madrid, Campus de Montegancedo,  
28660 Boadilla del Monte, Madrid, Spain  
{hkarshenas, mcbielza, pedro.larranaga}@fi.upm.es,  
roberto.santana@upm.es

**Abstract.** The objective values information can be incorporated into the evolutionary algorithms based on probabilistic modeling in order to capture the relationships between objectives and variables. This paper investigates the effects of joining the objective and variable information on the performance of an estimation of distribution algorithm for multi-objective optimization. A joint Gaussian Bayesian network of objectives and variables is learnt and then sampled using the information about currently best obtained objective values as evidence. The experimental results obtained on a set of multi-objective functions and in comparison to two other competitive algorithms are presented and discussed.

**Keywords:** Multi-objective Optimization, Estimation of Distribution Algorithms, Joint Probabilistic Modeling.

## 1 Introduction

Real-world problems usually include several criteria that should be fulfilled at the same time when trying to solve them. In many of these problems none of the criteria can be preferred over the others by the decision maker (the person who the optimization results are meant for) in order to apply single-function optimization techniques to solve them. On the other hand the criteria may be conflicting, i.e. trying to improve one of them will result in worse values for some other. Therefore it seems more reasonable to try solving them as Multi-objective Optimization Problems (MOPs).

Multi-Objective Evolutionary Algorithms (MOEAs) [1,6,29,34] have been successfully applied to many MOPs and obtained competitive results. Estimation of Distribution Algorithms (EDAs) [16,19,23,25] are proposed as a new computation paradigm based on evolutionary algorithms that replace the traditional recombination operators by learning and sampling a probabilistic model for advancing the search in solution space. Different Multi-objective EDAs (MEDAs) [20,26,32,33] have been proposed for solving MOPs. The main idea in these algorithms is to incorporate the selection and replacement strategies of MOEAs

in the model-building framework of EDAs which will allow to utilize the power of EDAs for solving MOPs.

Although most of the study on MOPs has been focused on problems with a few number of objectives, very often practical optimization problems involve a large number of criteria. In addition, after the initial success of applying MOEAs to problems with two or a small number of objectives, efforts have been oriented towards investigating the scalability of these algorithms with respect to the number of objectives [8,13]. Therefore problems with many-objectives are receiving an increasing attention in the fields of decision making and multi-objective optimization. Another line of research related to many-objective problems is to reduce the optimization complexity by exploiting the relationships between the objectives. Objective reduction is one of the proposed methods that seeks the minimum objective subset by considering the conflicting and non-conflicting objectives [3,4,18].

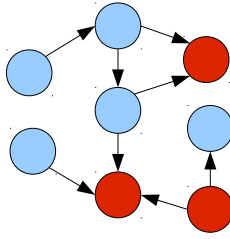
In this paper an MEDA using a generalized Gaussian Bayesian Network (GBN) is proposed for solving MOPs. This probabilistic model not only allows capturing the dependencies between problem variables, but also permits modeling the dependencies between objective functions and the dependencies between objectives and variables which can be used for factorizing MOP into simpler subproblems. Furthermore it allows the insertion of information about good objective values to the model as evidence.

A similar idea has been used in the Evolutionary Bayesian Classifier-based Optimization Algorithm (EBCOA) [21,22] for single-objective optimization where a single class variable is introduced as a node in the Bayesian classifier models. However, there the class variable is inserted into the model with a fixed relation structure and only having a predefined limited number of different values used to classify the solutions to fitter and worse groups. The algorithm presented here extends the scope to multi-objective problems using a more general Bayesian network and includes the objective variables as the nodes in the model. The dependency structure between the objectives is also explored in the course of evolution, capturing the relation between the objectives of the problem.

The rest of this paper is organized as follows: in Section 2 the proposed EDA and its probabilistic model learning process are described in detail. The conducted experiments and their results are presented in Section 3. Section 4 gives the conclusive remarks and lines of future research.

## 2 Joint Modeling of Objectives and Variables

In a typical Evolutionary Algorithm (EA), the objective function values of solutions are used for selecting parents or replacing offspring, and beyond that the new-solution-generating part only relies on the information provided by variables. However if this new-solution-generator could exploit the objectives information, then it might be able to generate better solutions. In the case of EDAs, incorporating the objectives information to the probabilistic model can help finding out how solutions are involved in building fitter solutions and also to capture the relationship between objectives and the variables in the multi-objective context.



**Fig. 1.** An example of a joint model of 3 objectives and 5 variables

The deterministic functions given for a problem, link the objective values to variables. EDAs try to represent the problem structure by probabilistically approximating the dependencies between variables and then use them for obtaining new solutions with better objective values. Including the objectives in the model will allow the algorithm exploit the probabilistic approximation of the relationships learnt for them (e.g. based on the expected value of the objectives). This is especially useful in multi-objective problems with different, possibly conflicting, objective functions. Furthermore, the model structure makes it possible to identify redundancy, conditional independence, or other relationships between objectives. Fig. 1 shows a sample joint model that depicts different dependencies between variables and objectives.

The probabilistic model used in this paper is a Bayesian network. As it is usual for many continuous EDAs and because of its special analytical properties, it is assumed that the problem solutions follow a Gaussian distribution and therefore, the probabilistic model will be a Gaussian Bayesian Network (GBN). To learn the joint GBN, the strings of selected solutions are extended by appending the objective values of each solution to its variable values. Fig. 2 shows an overview of the algorithm. The main steps are each described next in detail.

## 2.1 Selection

The set of promising solutions are selected using a truncation scheme where given a factor  $\tau$  and a population of  $N$  individuals, the selection operator first sorts the population and then selects the best  $\tau N$  solutions. One of the frequently used sorting algorithms in the multi-objective optimization is the non-dominated sorting introduced in [7]. The algorithm starts with sorting the solutions to different non-dominated fronts according to the dominance relation.

After grouping the population into several disjoint fronts, the solutions in each of the fronts are ranked using a crowding distance measure which favors those solutions that are in scattered areas of the front. The crowding distance computes how close each solution is to its neighbors with regard to different objective functions. To apply the truncation selection using non-dominated sorting, the non-dominated fronts are added to the set of selected solutions one by one according to their ranks and when a complete front can not be added anymore a subset of its solutions are selected according to their crowding distances. A slight modification to this algorithm is to select solutions one-by-one from the

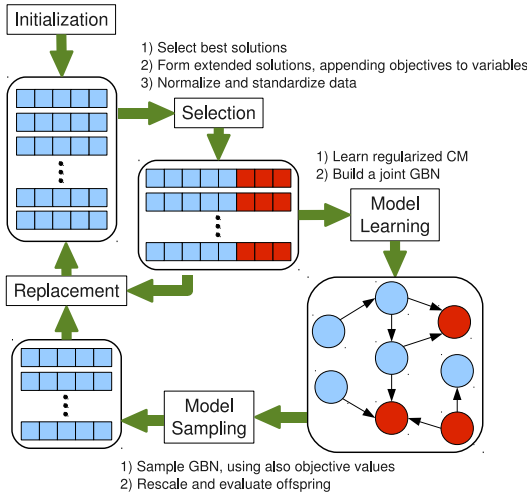


Fig. 2. An overview of the proposed algorithm

last eligible front and after removing each selected individual, recomputing the crowding distance of all individuals still in that front. Although this modification will cause an increase in the computational cost, it can help to improve the diversity of the selected solutions [33].

Another sorting algorithm applied in this paper is based on Fitness Averaging (FA). It is a very simple and fast sorting mechanism for multi-objective optimization problems. Instead of treating different objective functions using dominance relation, a single fitness value is assigned to each solution which is the average of different objective values for that solution (after normalizing the objective values). Since this kind of fitness assignment gives a complete ordering between the solutions, selection can be performed using any desired selection scheme.

### 2.2 Model Learning

As a Bayesian network, the graphical structure in GBN is represented with a directed acyclic graph (DAG) and the conditional probability density of the continuous nodes are given by the following normal distribution [14,17]

$$p(x_i|pa(X_i)) = \mathcal{N}(\mu_i + \sum_{X_j \in Pa(X_i)} w_j(x_j - \mu_j), \sigma_i^2) \tag{1}$$

where  $\mu_i$  and  $\sigma_i$  are the parameters of the joint Gaussian distribution  $p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ ,  $Pa(X_i)$  is the parent set of the  $i$ th variable according to the structure, and  $w_j$  are the weights of the conditional density function.

If the notation  $(X_1, \dots, X_p) = (V_1, \dots, V_n, O_1, \dots, O_m)$  is used to represent the joint extended vector of problem variables and objectives, then the following factorization shows the probability density encoded in the resulting GBN

$$p(v_1, \dots, v_n, o_1, \dots, o_m) = \prod_{i=1}^n p(v_i | pa(V_i)) \cdot \prod_{j=1}^m p(o_j | pa(O_j)) \tag{2}$$

where  $pa(V_i)$  and  $pa(O_j)$  are the parents value setting of variable  $V_i$  and objective  $O_j$  respectively.

To further increase the learning accuracy of the probabilistic model, a regularization method is employed in estimating the parameters. The covariance matrix (CM) shrinkage technique discussed in [30] is applied to the covariance matrix computation of the joint Gaussian distribution before using it for learning the GBN. In this method the unbiased empirical estimation of the covariance matrix is combined with a diagonal matrix using a linear shrinkage. The shrinkage intensity is analytically computed using the statistical analysis of empirical covariance and correlation matrices. The regularized estimation increases the sparsity of the inverse covariance matrix which represents the relationships between variables.

Before passing the set of selected (extended) solutions to the model learning algorithm, they are standardized to have a mean of zero and a variance of one, in order to facilitate the parameter estimation [31]. The learning algorithm is based on the search and score strategy and employs a local search method for moving in the space of all possible DAG structures. It checks all edge additions, deletions and removals applicable to the current network structure and selects the one that results in the best improvement of the scoring metric and terminates if no further improvement is possible [5]. The scoring metric used here is a penalized log-likelihood metric with a Bayesian Information Criterion (BIC) penalization. The parameter stored for the node corresponding to the  $i$ th variable is the weight vector  $\mathbf{w}$  needed for computing the conditional probability distribution of (1)

$$\mathbf{w} = \frac{1}{A} \frac{\Sigma_{(Pa(X_i), X_i)}}{A^T} \tag{3}$$

where  $\Sigma_{(K,L)}$  denotes the sub-matrix obtained from  $\Sigma$  by selecting the rows in set  $K$  and columns in set  $L$ , and the lower triangular matrix  $A$  is obtained from a *Cholesky* decomposition [27] of the parent part of the covariance matrix

$$AA^T = \Sigma_{(Pa(X_i), Pa(X_i))} \tag{4}$$

The condition for terminating the algorithm is to reach a maximum number of iterations. Therefore if the learning algorithm gets stuck in a local optimum point (or possibly the global optimum) before the number of iterations elapses, it will restart the search from another point randomly chosen in the DAGs space [31].

### 2.3 Model Sampling

The sampling procedure used in this study for the joint GBN is very similar to that of a typical Bayesian network sampling method except that special care

is needed to handle objective nodes included in the model. The Probabilistic Logic Sampling (PLS) or Forward Sampling [14] works by first computing an ancestral or topological ordering between variables and then generating values for the variables according to this ordering, using the conditional probability distributions stored in the model.

Two strategies can be applied for sampling a joint probabilistic model: a) using the current best found values for the objectives, or b) generating dummy values for objectives in the process of sampling. A combination of these two methods is also possible. In the former case, the objective values of the best solutions of the population (i.e. the parents set) is passed to the sampling method to be used whenever an objective node has appeared as the parent of a variable node and its value is required for sampling. In the latter method, the objective nodes are completely treated as variable nodes and values are generated for them using the probabilities encoded in the model. Although these values are not used after sampling, and the new generated solutions need to be evaluated using objective functions, the dummy objective values can be used for generating new values for variable nodes and can lead to a more consistent sampling with regard to the learnt model. The results presented in this paper are obtained using the second method.

Since the new solutions are generated using a bell-shaped normal distribution, it may happen that some of the generated values fall outside of the domain of the variables. Therefore usually a repairment step becomes necessary after the new solutions are generated in the continuous EDAs based on Gaussian distribution. The repairing technique used in this paper is to reset those values that are out of the bound, to a random value in the acceptable domain of the variable. To increase the possibility of appropriate value resetting, the variables are both normalized and standardized before modeling takes place and the mean of each variable in each solution is computed according to the probabilistic model. Then the generated unacceptable value is replaced with a random value between the computed mean and the domain-bound (upper or lower) that was breached.

## 2.4 Replacement

The newly generated solutions should be incorporated into the original population for further exploitation in next generations. The *Elitist* replacement used in this paper selects the best solutions, in terms of Pareto dominance, of the current original and offspring populations for inclusion in the following generation's population. The non-dominated sorting discussed earlier for selection is used here to order the combined population of original and offspring solutions and then select the best solutions. The number of solutions to be selected is determined by population size.

The following pseudo-code summarizes the described algorithm steps proposed for joint modeling of variables and objectives.

```

Joint Variable-Objective Estimation of Distribution Algorithm:
P[0] = Generate an initial population
F[0] = Compute the objective values(P[0])
t = 0
While termination criteria are not satisfied do
  {S, G} = Select a subset of solutions(P[t], F[t])
  D = Form the extended solutions(S, G)
  // D is duly normalized and standardized
  C = Compute the regularized covariance matrix(D)
  M = Learn the GBN model(C)
  Q = Sample offspring from the model(M, G)
  H = Compute the objective values(Q)
  {P[t+1], F[t+1]} = Replace offspring in population(P[t], F[t], Q, H)
  t = t + 1
End while

```

### 3 Experiments

The proposed algorithm is tested on a number of test functions and the results are compared to those of other algorithms. To test the effect of including the objectives in the model building and sampling of the algorithm two versions of the algorithm are considered here. The first algorithm learns a joint model of both the objectives and variables, which will be called Joint GBN-EDA (JGBN-EDA). The other version does not consider objective information and only uses variables for model learning, very similar to the Estimation of Gaussian Network Algorithm (EGNA) [15], and will be called (normal) GBN-EDA in this paper.

The performance results are compared against two other algorithms: The Non-dominated Sorting Genetic Algorithm (NSGA-II) [7], considered as a reference algorithm in many multi-objective optimization studies, is based on traditional GA and uses special type of crossover and mutation methods to deal with real-valued strings; The Regularity-Model based Multi-objective Estimation of Distribution Algorithm (RM-MEDA) [33] assumes a certain type of smoothness for Pareto set and uses the Local Principal Component Analysis (LPCA) algorithm to build a piece-wise continuous manifold with a dimension equal to one less than the number of objectives.

The two algorithms use the non-dominated sorting method for selecting promising solutions. While both algorithms generate an offspring population of the size of the original population, only NSGA-II employs a replacement mechanism where the best solutions in both offspring and original populations are selected for the next generation. In RM-MEDA the newly generated offspring solutions completely replace the original population. All of the algorithms are implemented in Matlab<sup>®</sup>.

The implementation of JGBN-EDA is done using the Matlab<sup>®</sup> toolbox for EDAs [28]. The algorithm has been tested using both the non-dominated sorting and the fitness averaging for selection. However, the results obtained with the

simpler fitness averaging were superior to those obtained with the other algorithm. Such a behavior is also reported by others [10]. Therefore the results presented here are those obtained with fitness averaging. The non-dominated sorting is used only in the replacement step of the algorithm as was discussed above.

To evaluate the non-dominated fronts obtained by each algorithm the Inverted Generational Distance (IGD) measure is used that accounts both for diversity and convergence to the true Pareto front at the same time. The measure is given by the following relation taken from [33]

$$IGD_{F^*}(F) = \left( \sum_{s \in F^*} \{ \min d(s, s'), \forall s' \in F \} \right) / |F^*| \tag{5}$$

where  $d()$  gives the Euclidean distance between the two points and  $F^*$  is a uniform sampling of the true Pareto front of the problem. The statistical significance in the differences between the results is checked with the Kruskal-Wallis test. Kruskal-Wallis performs a non-parametric one-way ANOVA to accept or reject the null hypothesis that independent samples of two or more groups come from distributions with equal medians, and returns the  $p$ -value for that test. The test significance level is set to 0.05.

### 3.1 Test Functions

There are many test sets proposed in the literature for evaluating multi-objective optimization algorithms [8,9,11,24]. However not all of these functions exhibit real-world problem properties that can be used to evaluate different aspects of an algorithm. Features like scalability and non-separability both in the variable and objective spaces, multi-modality, biased or disconnected Pareto fronts that can pose significant challenges to any multi-objective optimization algorithm.

In this study, the Walking Fish Group (WFG) functions proposed by Huband et al. [11,12] are used for experiments. All of the functions, which are to be minimized, can be represented with the following relation

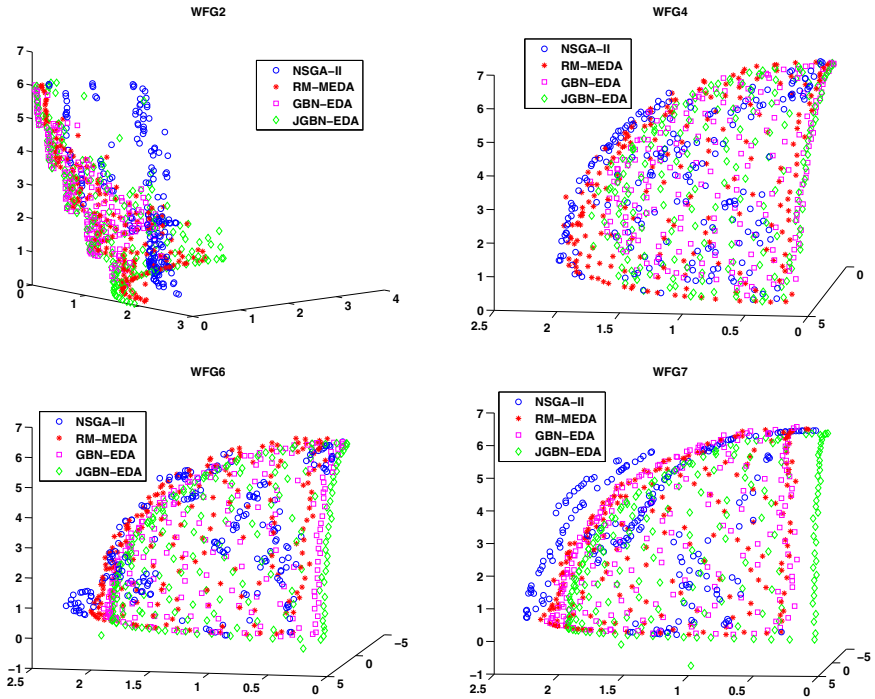
$$f_j(\mathbf{x}) = D \cdot x_m + S_j \cdot h_j(x_1, \dots, x_{m-1}) \tag{6}$$

where  $D$  and  $S_j$  are scaling factors,  $h_j$  are shape functions (e.g. concave, convex, mixed, disconnected or degenerated) defined on  $m - 1$  ( $m$  is the number of objectives) position variables, and  $x_m$  is the distance variable. Position and distance variables are obtained by applying different transformation function on the input variables. The test set includes 9 functions that all have dissimilar variable domains and different Pareto optimal trade-off magnitudes, have no extremal nor medial values for variables, and undergo complex transformations to create other properties like deceptiveness and many-to-one mappings.

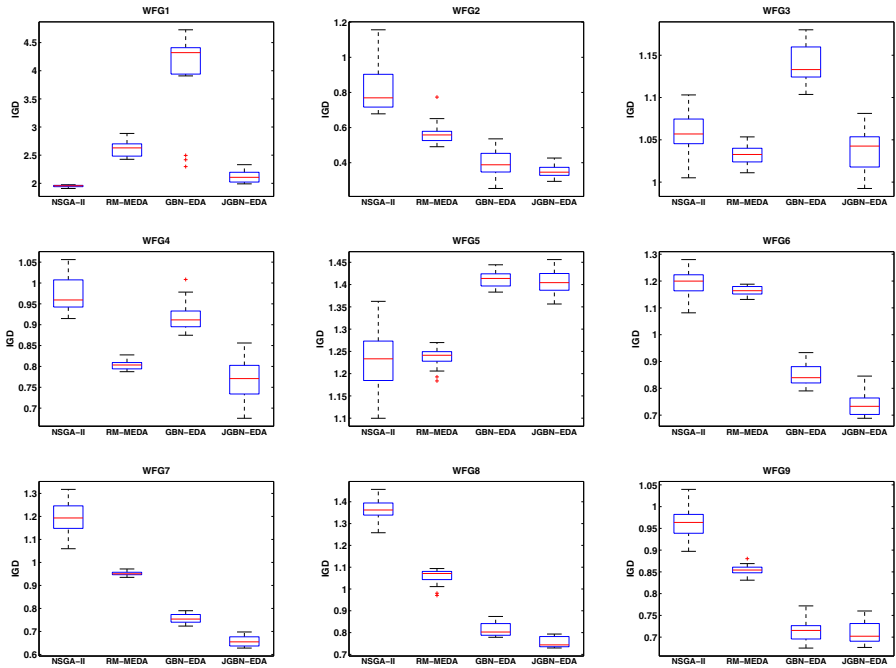


### 3.2 Experimental Results

In the following experiments the algorithms are tested on WFG functions with 5, 10 and 20 objective functions. The number of objectives are related to the number of position and distance variables with two factors that determine the number of problem variables. Therefore to keep the computational costs of the experiments in an affordable level, the number of variables is set to 28, 50 and 50 respectively. The reported results for each algorithm on each test function are computed from 25 runs in the case of 5 and 10 objectives, and 10 runs for 20 objectives. For all algorithms the maximum number of function evaluations is set to  $1.5 \cdot 10^5$ ,  $3 \cdot 10^5$  and  $7.5 \cdot 10^5$ , and the population size to 600, 1000 and 1500 respectively for different number of objectives. Figure 3 shows the typical fronts obtained by the algorithms for some of the test functions (WFG2, WFG4, WFG6 and WFG7) with 3 objectives and 20 variables to give an idea of how the algorithms are performing for smaller number of objectives. The population size is set to 200 in this case.



**Fig. 3.** Typical Pareto fronts obtained for different algorithms performance on some of (WFG) functions with 3 objectives

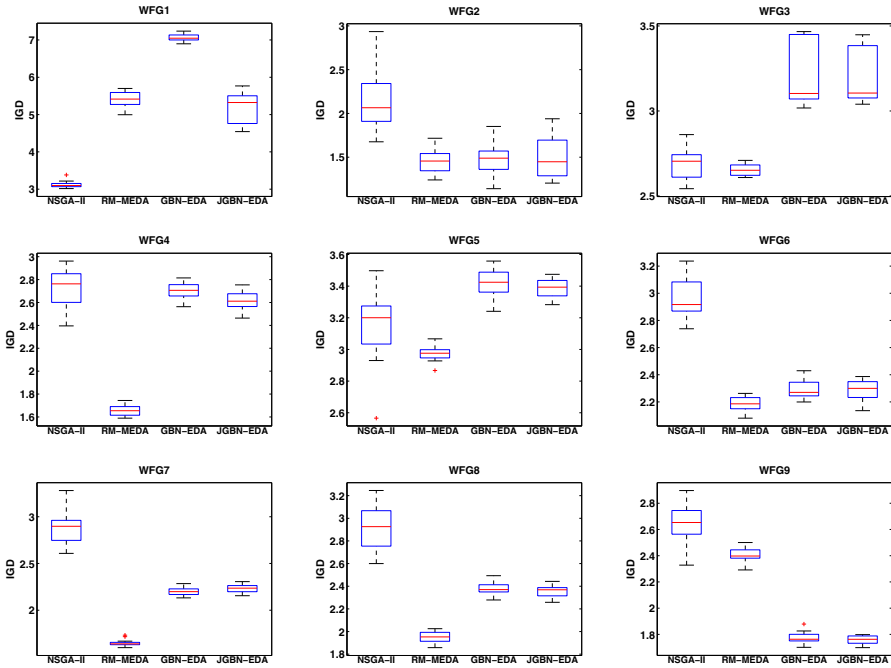


**Fig. 4.** Comparison of different algorithms performance on (WFG) functions with 5 objectives

Fig. 4 shows the results obtained for each of the algorithms for WFG functions with 5 objectives. As it can be seen in the figure, the incorporation of objectives in the modeling of JGBN-EDA enables this algorithm to obtain a (significantly) better performance on most of the functions. A direct comparison of GBN-EDA and JGBN-EDA shows the effectiveness of joint modeling where except for WFG5 and WFG9 functions, the latter algorithm can obtain significantly better results on all other functions ( $p = 0.05$ ).

WFG5 is a separable and deceptive function while WFG9 is a non-separable, biased, multi-modal and deceptive function. According to the presented results, for deceptive functions the information provided by objectives does not have a great impact on the probabilistic model learnt in JGBN-EDA for generating better solutions. Also, the algorithm is not able to properly utilize the separability of variables in WFG5 to obtain better fronts. However, the results obtained for WFG9 shows that non-separability and multi-modality of this problem is completely addressed with the modeling used in GBN-EDAs which makes them able to dominate the other two competitor algorithms on this function.

As the number of objectives grows to 10 and 20 (Figs. 5 and 6), the proposed algorithm performance deteriorates in comparison to other algorithms. The other two algorithms also show a diverse behavior on different number of

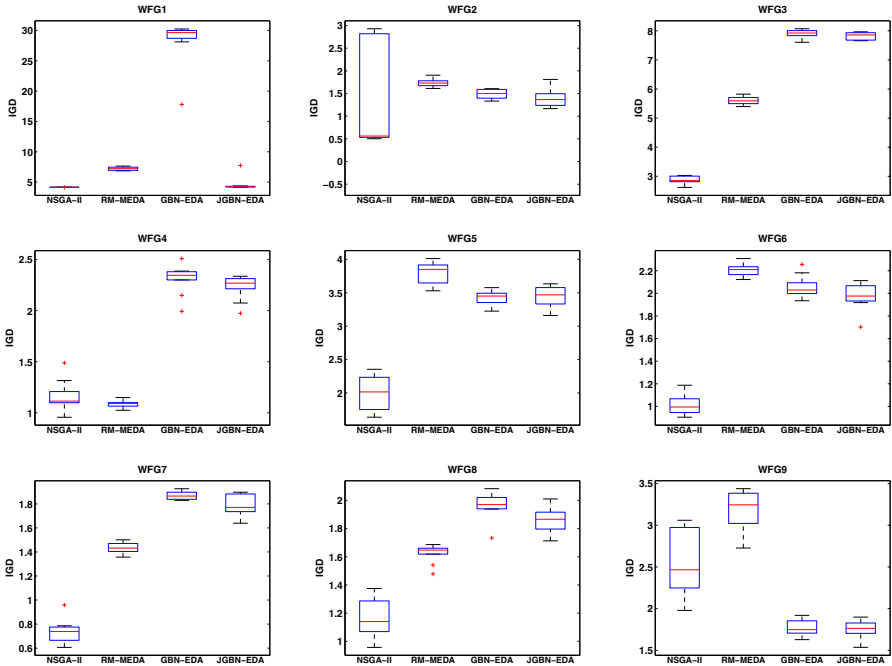


**Fig. 5.** Comparison of different algorithms performance on (*WFG*) functions with 10 objectives

objectives. While the fronts resulted by NSGA-II on 10 objectives functions are not comparative to other algorithms, it is able to find significantly better results for most of the functions with 20 objectives. RM-MEDA is showing an apposite behavior, being the better algorithm on the 10 objective case. For these high number of objectives the performance of the proposed JGBN-EDA seems to be less varying when compared on both 10 and 20 objectives WFG functions.

The mixed concave-convex front of WFG1 with some flat regions in the Pareto set is best addressed by the recombination method employed in NSGA-II. The inclusion of objective values in the modeling of JGBN-EDA has a major influence in improving the performance on this function when compared to GBN-EDA. On the other hand, the disconnected front of WFG2 causes a diverse performance of NSGA-II, while the EDAs are almost performing similar on 10 and 20 objectives functions. The significantly better performance of JGBN-EDA on WFG9 is repeated for 10 and 20 objectives.

It seems that when the number of objectives increase to a high number (e.g. 20 objectives), the modeling present in EDAs gets saturated for small populations and can not help the algorithm to make progress. In fact for such a large number of objectives small population sizes may not suffice to represent the Pareto front.



**Fig. 6.** Comparison of different algorithms performance on (*WFG*) functions with 20 objectives

This issue also poses a challenge for representing the true Pareto front to be used for the computation of performance measure. Although in this study front approximations with several hundred thousands of points have been used for this purpose, some areas of the front may not yet be covered properly.

### 4 Conclusion and Further Research

A joint modeling of objectives and variables was proposed to be used for multi-objective optimization in this paper. The proposed EDA, learns a Gaussian Bayesian network to encode the relationships between the variables and objectives, from a set of extended solutions formed by appending the objectives to the variables of each solution. The learnt probabilistic model will have both variable and objective nodes. The sampling procedure generates new values for problem variables using the objective values when such dependencies are encountered in the model.

The performance of the algorithm was tested on a number of multi-objective test functions with different real-world properties. The results show that the incorporation of objectives in the modeling can help the algorithm to obtain

better fronts on some of the functions. The results were also compared against two other algorithms and indicated that the idea of including objective values in the modeling step of an EDA is promising and can help to obtain better fronts in multi-objective optimization.

The algorithm had some problems in detecting the correct search bias for some deceptive functions using the proposed joint modeling. Also for some of the functions, the algorithm was not able to obtain competitive fronts when applied to high number of objectives. The effect of proper population sizing on the performance of the algorithm specially for many objectives problems should be studied in more detail. Other ways of utilizing the objective values in the sampling process can be used to reduce this shortcoming.

Nevertheless the proposed algorithm can be seen as an alternative for using modeling in the multi-objective optimization. The information provided by dependencies between the objectives can be further investigated for obtaining the relationships in problems with many objectives. The factorization obtained by the explicit inclusion of objectives in the model is also another possibility to simplify the problem. A future line of research associated with this is to force some special kind of relations in the joint probabilistic model, like those discussed for class-bridge decomposable multi-dimensional Bayesian network classifiers [2], that will allow to decompose the problem into smaller subproblems, thus easing the learning and sampling tasks.

**Acknowledgments.** This work has been partially supported by TIN2010-20900-C04-04, Consolider Ingenio 2010-CSD2007-00018 and Cajal Blue Brain projects (Spanish Ministry of Science and Innovation).

## References

1. Abraham, A., Jain, L., Goldberg, R. (eds.): *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. Advanced Information and Knowledge Processing. Springer, Berlin (2005)
2. Bielza, C., Li, G., Larrañaga, P.: *Multi-dimensional classification with Bayesian networks*. Technical Report UPM-FI/DIA/2010-1, Artificial Intelligence Department, Technical University of Madrid, Madrid, Spain (2010)
3. Brockhoff, D., Zitzler, E.: *Dimensionality reduction in multiobjective optimization: The minimum objective subset problem*. In: Waldmann, K.-H., Stocker, U.M. (eds.) *Operations Research Proceedings 2006*, pp. 423–429. Springer, Berlin (2007)
4. Brockhoff, D., Zitzler, E.: *Objective reduction in evolutionary multiobjective optimization: Theory and applications*. *Evolutionary Computation* 17(2), 135–166 (2009)
5. Buntine, W.: *Theory refinement on Bayesian networks*. In: *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, vol. 91, pp. 52–60. Morgan Kaufmann, San Francisco (1991)
6. Coello, C.: *An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends*. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1999)*, vol. 1, pp. 3–13 (1999)

7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
8. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, vol. 1, pp. 825–830 (2002)
9. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) *EMO 2005*. LNCS, vol. 3410, pp. 105–145. Springer, Heidelberg (2005)
10. Garza-Fabre, M., Pulido, G., Coello, C.: Ranking methods for many-objective optimization. In: Aguirre, A.H., Borja, R.M., Garcíá, C.A.R. (eds.) *MICAI 2009*. LNCS, vol. 5845, pp. 633–645. Springer, Heidelberg (2009)
11. Huband, S., Barone, L., While, L., Hingston, P.: A scalable multi-objective test problem toolkit. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) *EMO 2005*. LNCS, vol. 3410, pp. 280–295. Springer, Heidelberg (2005)
12. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10(5), 477–506 (2006)
13. Ishibuchi, H., Tsukamoto, N., Hitotsuyanagi, Y., Nojima, Y.: Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO 2008)*, pp. 649–656. ACM, New York (2008)
14. Koller, D., Friedman, N.: *Probabilistic Graphical Models: Principles and Techniques*. In: *Adaptive Computation and Machine Learning*. The MIT Press, Cambridge (2009)
15. Larrañaga, P., Etxeberria, R., Lozano, J., Peña, J.: Optimization in continuous domains by learning and simulation of Gaussian networks. In: Wu, A. (ed.) *Proceedings of the 2000 Genetic and Evolutionary Computation Conference (GECCO 2000) Workshop Program*, pp. 201–204. Morgan Kaufmann, San Francisco (2000)
16. Larrañaga, P., Lozano, J. (eds.): *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Dordrecht (2001)
17. Lauritzen, S.L.: Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association* 87(420), 1098–1108 (1992)
18. López Jaimes, A., Coello Coello, C.A., Chakraborty, D.: Objective reduction using a feature selection technique. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO 2008)*, pp. 673–680. ACM, New York (2008)
19. Lozano, J., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.): *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. *Studies in Fuzziness and Soft Computing*, vol. 192. Springer, Heidelberg (2006)
20. Martí, L., Garcia, J., Antonio, B., Coello, C.A., Molina, J.: On current model-building methods for multi-objective estimation of distribution algorithms: Shortcomings and directions for improvement. Technical Report GIAA2010E001, Department of Informatics, Universidad Carlos III de Madrid, Madrid, Spain (2010)
21. Miquélez, T., Bengoetxea, E., Larrañaga, P.: Evolutionary computation based on Bayesian classifiers. *International Journal of Applied Mathematics and Computer Science* 14(3), 335–350 (2004)

22. Miquélez, T., Bengoetxea, E., Larrañaga, P.: Evolutionary Bayesian Classifier-Based Optimization in Continuous Domains. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 529–536. Springer, Heidelberg (2006)
23. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions i. binary parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
24. Okabe, T., Jin, Y., Olhofer, M., Sendhoff, B.: On test functions for evolutionary multi-objective optimization. In: Yao, X., Burke, E., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J., Tino, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 792–802. Springer, Heidelberg (2004)
25. Pelikan, M., Sastry, K., Cantú-Paz, E. (eds.): Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications. SCI. Springer, Heidelberg (2006)
26. Pelikan, M., Sastry, K., Goldberg, D.: Multiobjective estimation of distribution algorithms. In: Pelikan, et al. (eds.) [25], pp. 223–248
27. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. In: Adaptive Computation and Machine Learning. The MIT Press, Cambridge (2005)
28. Santana, R., Bielza, C., Larrañaga, P., Lozano, J.A., Echegoyen, C., Mendiburu, A., Armañanzas, R., Shakya, S.: Mateda-2.0: Estimation of distribution algorithms in MATLAB. *Journal of Statistical Software* 35(7), 1–30 (2010)
29. Sbalzarini, I., Mueller, S., Koumoutsakos, P.: Multiobjective optimization using evolutionary algorithms. In: Proceedings of the 2000 Summer Program of Studying Turbulence Using Numerical Simulation Databases–VIII, vol. 1, pp. 63–74. Center for Turbulence Research (November 2000)
30. Schäfer, J., Strimmer, K.: A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology* 4(1) (2005)
31. Schmidt, M., Niculescu-Mizil, A., Murphy, K.: Learning graphical model structure using L1-regularization paths. In: Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2007), vol. 2, pp. 1278–1283. AAAI Press, Menlo Park (2007)
32. Thierens, D., Bosman, P.: Multi-objective mixture-based iterated density estimation evolutionary algorithms. In: Spector, L., Goodman, E.D., Wu, A., Langdon, W.B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H., Burke, E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), pp. 663–670. Morgan Kaufmann, San Francisco (2001)
33. Zhang, Q., Zhou, A., Jin, Y.: RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation* 12(1), 41–63 (2008)
34. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation* 8(2), 173–195 (2000)