

# Critical Issues in Model-Based Surrogate Functions in Estimation of Distribution Algorithms

Roberto Santana, Alexander Mendiburu, and Jose A. Lozano

Intelligent Systems Group

Department of Computer Science and Artificial Intelligence

University of the Basque Country (UPV/EHU)

Paseo Manuel de Lardizabal 1, 20080, San Sebastian, Guipuzcoa, Spain

{roberto.santana,alexander.mendiburu,ja.lozano}@ehu.es

**Abstract.** In many optimization domains the solution of the problem can be made more efficient by the construction of a surrogate fitness model. Estimation of distribution algorithms (EDAs) are a class of evolutionary algorithms particularly suitable for the conception of model-based surrogate techniques. Since EDAs generate probabilistic models, it is natural to use these models as surrogates. However, there exist many types of models and methods to learn them. The issues involved in the conception of model-based surrogates for EDAs are various and some of them have received scarce attention in the literature. In this position paper, we propose a unified view for model-based surrogates in EDAs and identify a number of critical issues that should be dealt with in order to advance the research in this area.

**Keywords:** estimation of distribution algorithms, surrogate functions, function approximation, probabilistic modeling, most probable configuration, abductive inference.

## 1 Introduction

Surrogate functions are approximations of objective or fitness functions that usually allow a more efficient search for optimal solutions in evolutionary algorithms (EAs). There is a variety of techniques used to construct surrogate functions in EAs [17,44]. They include instance-based learning methods, machine learning methods, and statistical learning methods [44]. In this position paper we focus on methods that are based on the use of probabilistic models for estimation of distribution algorithms (EDAs) [22,24]. These methods combine machine learning techniques and statistical procedures to learn and exploit the models.

The characteristic feature of EDAs with respect to other EAs is the use of probabilistic modeling to capture the most relevant features of the selected solutions. EDAs have been praised for their capacity to capture and exploit the interactions between the problem variables, limiting to a large extent the disruption of partial solutions [21,43]. In many cases, the probabilistic model learned

by EDAs is able to explicitly represent the problem structure which amounts to produce in each generation a candidate model for problem decomposition.

Since probabilistic models are a by-product of EDAs, a natural question is to what extent can these models be used as surrogates and which are the tasks that can be accomplished with this type of surrogates. These questions have been seldom addressed in the literature from a general perspective. When considering probabilistic models as surrogates, the focus has been on a particular type of EDAs, those based on Markov networks [5,6]. Furthermore, while the main application of surrogates are in the replacement of the original fitness value by its approximation, there are several scenarios in which they can be used in an indirect way (e.g., comparing and ranking solutions, partial evaluation, etc.). One of our contributions in this paper is addressing all these issues from a common perspective, pointing to the links with other relevant aspects of probabilistic modeling in EDAs. We also expand our analysis to cover uses of model-based surrogates for multi-objective problems (MOP) [11].

The paper is organized as follows. In the next section, we introduce EDAs and discuss the role of probabilistic modeling in these algorithms. Section 3 proposes a classification for model-based fitness-surrogates. Section 4 analyzes alternatives for learning model-based surrogates, and different criteria to assess their quality are reviewed in Section 5. Model-based fitness surrogates of multi-objective functions are discussed in Section 6. The conclusions of our paper and some lines for future work are presented in Section 7.

## 2 Estimation of Distribution Algorithms

In this paper, the joint generalized probability distribution of  $\mathbf{x}$  is represented as  $p(\mathbf{X} = \mathbf{x})$  or  $p(\mathbf{x})$ .  $p(\mathbf{x}_S)$  will denote the marginal generalized probability distribution for  $\mathbf{X}_S$ . We use  $p(X_i = x_i \mid X_j = x_j)$  or, in a simplified form,  $p(x_i \mid x_j)$ , to denote the conditional generalized probability distribution of  $X_i$  given  $X_j = x_j$ .

In EDAs, the new population of solutions is sampled from a probability distribution, which is estimated from a database that contains the selected solutions from the current generation. Thus, the interactions between the different variables that represent the solutions are explicitly expressed through the joint probability distribution associated with the solutions selected at each generation. A pseudo-code of EDAs is described in Algorithm 1.

The termination criteria of an EDA can be a maximum number of generations, a homogeneous population or no improvement after a specified number of generations. The probabilistic model learned at step 5 has a significant influence on the behavior of the EDA from the point of view of complexity and performance.

**Algorithm 1.** Estimation of distribution algorithm

---

```

1  Set  $t \leftarrow 0$ . Generate  $M$  solutions randomly.
2  do {
3      Evaluate the solutions using the fitness function.
4      Select a set  $D_t^S$  of  $N \leq M$  solutions according to a selection method.
5      Calculate a probabilistic model of  $D_t^S$ .
6      Generate  $M$  new solutions sampling from the distribution represented
        in the model.
7       $t \leftarrow t + 1$ 
8  } until Termination criteria are met.

```

---

**2.1 Probabilistic Modeling in EDAs**

The main role of probabilistic modeling in EDAs is to capture an accurate representation of the regularities of the selected solutions. Frequently, these regularities correspond to the most common configurations of subsets of variables and patterns of interactions between the variables. Probabilistic models learned by EDAs can be classified according to the type of learning they use into two groups: 1) Models that apply parametrical learning; 2) Models that apply structural *and* parametric learning. In the first case, the structure of the model is known a priori and only the parameters are learned from data. In the second case, the structure and the parameters are learned from data. The original, an still primary application of a probabilistic model in EDAs is to generate new solutions. However, probabilistic models of fitness functions can be applied in different situations:

- To create surrogate functions that help to diminish the number of evaluations for costly functions [30,41,42].
- To obtain models of black box optimization problems for which an analytical expression of the fitness function is not available.
- To unveil and extract problem information that is hidden in the original formulation of the function or optimization problem [6,37].
- To design improved (local) optimization procedures based on the model structure [7,32,34,41].

**2.2 MOEDAs**

In MOPs, two or more, often conflicting, objectives have to be simultaneously optimized. Different variants of MOEDAs have been applied to these problems [21,25,48]. Particularly relevant for our analysis is the recently introduced MOEDA based on the joint probabilistic modeling of variables and objectives [20]. In this approach, a variable  $Y_i, i \in \{1 \dots k\}$  is associated to the objective  $f_i$  in such a way that  $Y_i$  takes values in the image of  $f_i$ , i.e.  $y_i = f_i$ .

A probability distribution  $\hat{p}(x_1, \dots, x_n, y_1, \dots, y_k)$  is defined as the joint probability distribution of variables and objectives. The selected solutions from which

$\hat{p}$  is learned should correspond to good candidate sets for Pareto set approximations [20]. The probabilistic model  $\hat{p}(x_1, \dots, x_n, y_1, \dots, y_k)$  will eventually capture conditional probabilistic relationships between objectives and variables. This model will capture the variable-objective mapping.

The key issue here is that the dependencies represented in the model could serve as a characterization of the relationships in the Pareto set approximations, possibly revealing characteristic patterns in this set of solutions.

### 3 Types of Model-Based Fitness Surrogates

In this section we propose a general classification of model-based fitness surrogates according to their use and the type of models they are based on. We start by identifying two situations in which surrogate functions may be needed:

1.  $\lambda$ -error surrogate: Approximating the fitness function to a desired level of accuracy  $\lambda$ .
2.  $\alpha$ -ranking surrogate: Using surrogate values to rank solutions with a level of accuracy  $\alpha$ .

Let  $\hat{f}(\mathbf{x})$  be a surrogate function of  $f(\mathbf{x})$ .  $\hat{f}(\mathbf{x})$  is a  $\lambda$ -error surrogate if  $|f(\mathbf{x}) - \hat{f}(\mathbf{x})| < \lambda$ ,  $\forall \mathbf{x}$ . The  $\lambda$ -error surrogate defines a family of parametric functions that depend on  $\lambda$ .

$\hat{f}(\mathbf{x})$  is an  $\alpha$ -ranking surrogate if:

$$q \left( \text{sign}(f(\mathbf{x}) - f(\mathbf{y})) = \text{sign}(\hat{f}(\mathbf{x}) - \hat{f}(\mathbf{y})) \right) \geq \alpha \quad (1)$$

$$\text{sign}(\mathbf{x}) = \begin{cases} -1 & \text{for } \mathbf{x} < 0 \\ 0 & \text{for } \mathbf{x} = 0 \\ 1 & \text{for } \mathbf{x} > 0 \end{cases} \quad (2)$$

where  $q$  gives the proportion of pairs of solutions  $\mathbf{x}, \mathbf{y}$ , out of all possible pairs, that satisfy that the ordering relationship between  $\mathbf{x}$  and  $\mathbf{y}$ , defined by  $f$  is respected by the surrogate function. If  $\alpha = 1$ , then the surrogate function produces the same ranking that the original fitness function for any set of solutions.

Examples of models that can be respectively analyzed as a  $\lambda$ -error surrogate and an  $\alpha$ -ranking surrogate are the Markov fitness model (MFM) model [5] and the Boltzmann distribution used in the context of EDAs [28].

The MFM is based on a Markov network formed by a set of maximal cliques  $K = \{K_1, \dots, K_m\}$ . Then, for any solution  $\mathbf{x}$ , the model of the fitness function is given by:

$$-\ln(f(\mathbf{x})) = \sum_i \alpha_i V_{K_i}(\mathbf{x}) \quad (3)$$

where  $V_k$  are the characteristic functions of a Walsh decomposition of the fitness function, and  $\alpha_i$  are the model coefficients [5]. Given a sufficiently-sized set of

solutions and their fitness, the MFM can be found solving a system of equations in the parameters.

The Boltzmann probability distribution  $\hat{p}(\mathbf{x})$  is defined as

$$\hat{p}(\mathbf{x}) = \frac{e^{\frac{f(\mathbf{x})}{T}}}{\sum_{\mathbf{x}'} e^{\frac{f(\mathbf{x}')}{T}}}, \quad (4)$$

where  $\sum_{\mathbf{x}'} e^{\frac{f(\mathbf{x}')}{T}}$  is the so-called partition function, and  $T$  is the temperature of the system that can be used as a parameter to smooth the probabilities.

By definition, the MFM provides a  $\lambda$ -error surrogate model where  $\lambda$  can be estimated as the greatest difference, among all the solutions, between the original fitness value and its approximation. The Boltzmann distribution guarantees that  $\text{sign}(\hat{p}(\mathbf{x}^i) - \hat{p}(\mathbf{x}^j)) = \text{sign}(f(\mathbf{x}^i) - f(\mathbf{x}^j))$ . Therefore, the Boltzmann distribution is an 1-ranking surrogate model of  $f(\mathbf{x})$ .

## 4 Learning of Model-Based Surrogates of Single-Objective Problems

The question of how to learn a model-based surrogate is a fundamental one. Particularly, in optimization domains for which we intend to extract as much knowledge as possible from the fitness function with the minimal number of function evaluations. In this section we analyze probabilistic modeling in EDAs from the perspective of fitness surrogates.

When a sufficiently large set of evaluated solutions is available, and the structure of the problem is known, i.e., the cliques of the Markov network, a MFM can be learned in a straightforward way from the data. If the structure is not available, then it has to be recovered from the data and this is accomplished using statistical methods for learning probabilistic graphical models (PGMs) from data [40]. Computing the Boltzmann distribution requires the computation of the partition function which is infeasible when the dimension space of the solutions grows. Therefore, instead of computing the Boltzmann distribution, usually a PGM is learned from the data. We call this approximation PGM-based approach.

The MFM and PGM-based approaches coincide in that both need to learn a structural representation from data when it is not available a priori. However, they differ in the way parameters are computed, and the way they are applied in EDAs.

### 4.1 Fitness-Blind versus Fitness-Aware Model Computation

In most commonly used EDAs, the probabilistic model is learned from the set of selected solutions. Marginal probabilities are computed based on the frequencies of the solutions in the selected set. No information about the fitness of the solutions is directly encoded in the model. The rationale is that since solutions have been selected according to their fitness, and the goal of the algorithm is to

“model the best solutions”, all relevant information has been already considered at the time of selection.

However, disregarding the fitness information of the selected solutions, at the time of learning the model, or even not using the fitness information of the non selected solutions means that valuable information obtained from searching the space of solutions is wasted. This is particularly remarkable if we intend to use the PGM as fitness surrogate.

In [36], the idea of learning the marginal probabilities of the PGM from the fitness values of all the evaluated solutions was proposed and applied to different EDAs. Similar ideas have been shown to improve the efficiency of EDAs that learn multivariate models [29,46]. Other methods [27] follow an intermediate approach in which solutions are first classified according to their fitness into two or more groups, and then one or more models are learned using the class labels associated to the solutions. These methods are able to incorporate, to a greater extent, the information contained in the fitness values.

In this paper we introduce the terms of *fitness-blind* and *fitness-aware* model-based fitness surrogates to refer to methods that respectively use non or extensive information about the fitness during learning of the probabilistic model. We state that the design of fitness-aware model learning methods is a critical issue for model-based surrogates. Results confirming the benefits of fitness-aware learning for diminishing the number of function evaluations [29,36,46] are an example of the potential of these methods for learning more accurate surrogates.

## 5 Evaluation of the Models as Surrogates

Once a model has been learned, one important question is how to evaluate the quality of the model as a surrogate. In this section we review the methods used for evaluating model-based surrogates. Evaluation of the models is mostly approached comparing the original fitness values and the approximations produced by the model on a set of solutions. Two different elements of this process can be identified:

1. The method for generating the solutions that are later used to evaluate the models.
2. The measures for evaluating the models from the solutions and their associated objective evaluations.

The way the sample of solutions is selected is usually a question overlooked in the literature. However, we emphasize this is a very important question.

Let us suppose the number of solutions to be generated is  $k$ . We identify as relevant the following procedures to generate them:

- The  $k$  solutions are randomly generated [6].
- The  $k$  solutions correspond to the best known values (for a single objective) of the function or they are the selected solutions [6,15,35].

- Solutions correspond to the  $k$  most probable configurations (MPC) of the model [14,16].

The way solutions are sampled is very related to the criteria taken into consideration and the measures used to evaluate the model. Among the measures that can serve to evaluate different facets of the models are:

- Correlation between the probabilities assigned by the models to the solutions and their fitness values [6].
- Distance between the ranking of solutions induced by the fitness functions and their models.
- Expected fitness value of the model, i.e.  $\sum_x \hat{p}(x)f(x)$ .
- Entropy of the model.
- Sum of the probabilities assigned to the solutions [35].

Some methods used for generating solutions better fit with some particular approaches to measure the quality of the model. For random solutions, we can compare different models in terms of the correlations or the expected fitness values. In [6], the analysis of the correlation has been successfully employed to analyze the fitness modeling capabilities of EDAs based on Markov networks. The entropy of the model can also be used. A model that assigns the same or very similar probability to all the solutions is of scarce interest.

If the  $k$  best known solutions are used to evaluate the models, all the previous criteria can be used but, in addition, we can evaluate the capacity of the model to generate solutions different to those that have been previously evaluated (exploration, or generalization capability of the model). The exploration capability can be estimated by computing first the total probability assigned to the best solutions [35]. If the sum of the probabilities given by the model to the  $k$  best solutions is very high (e.g.  $\sum_{\mathbf{x}} p(\mathbf{x}) = 0.9$ ) then we can assume its capacity of exploration is very limited. The  $k$  most probable configurations can be computed using algorithms that employ abductive inference and dynamic programming as those used in [26] to generate better solutions at an earlier step of the evolution.

Few available implementations of EDAs incorporate methods for constructing and using model-based fitness surrogates and evaluate their quality. Some of the methods described above for generating the solutions and some measures to evaluate the quality of the models have been implemented as part of the MATEDA-2.0 software [38]. A critical issue for extending the application of model-based fitness surrogates is their implementation as part of available software.

The main aspect to be emphasized from the analysis presented in this section is that while models can serve as surrogate to approximate the fitness of solutions, carefully selected solutions can be also very important to obtain an accurate evaluation of the models as fitness surrogates.

## 6 Model-Based Fitness Surrogates of Multi-objective Problems

Extending model-based surrogates to MOPs is a critical issue for the applicability of these algorithm. The problem of fitness surrogates for MOPs is more complex than in the single-objective case because different classes of surrogates can be designed for different types of possible approximations. Among the approximation tasks that can be approached using model-based surrogates are:

1. Learning a single model of each objective function.
2. Learning a joint model of all the objectives.
3. Learning a model of the Pareto set.

The first task can be approached with the same methods analyzed for the single-objective case. However, notice that probabilistic models of a single objective could incorporate, as input variables, the information about the other objectives. This type of information should be particularly useful when there is some degree of redundancy or statistical dependency between the objectives.

Several objectives can be simultaneously estimated by using multi-dimensional classifiers [1,12]. Nevertheless, learning this type of classifiers is not, in general, an easy task.

Models of the Pareto set can output the probability that a given solution, its objective vector, or a combination of both, belongs to the Pareto set. This type of model could not be used as a surrogate of the individual objective functions but can serve to estimate the “global” fitness of a solution, understanding this fitness value as the capacity of the solution to belong to the Pareto set.

Understanding the differences between these three approximation tasks is an important question for the conception of model-based surrogates for MOPs. We briefly review some of the statistical methods used in multi-objective optimization to represent the relationships between variables and objectives. These methods can be instrumental in the conception of strategies for model-based fitness surrogates in MOEDAs.

1. Analysis of the objectives correlation matrix [13,23]: In [13], PCA is used for detecting conflicting objectives. A method based on a different, unsupervised, feature selection is proposed in [23].
2. Explicit modeling of variable dependencies [3,25,31,47]: This is the main goal of several variants of multi-objective EDAs.
3. Joint clustering of variables and objectives [45]. Similar solutions are grouped by finding so-called modules of the decision space [45]. Biclustering and dendrogram visualization are used with this purpose.
4. Objective-based mixture modeling of variables’ dependencies [2]: Solutions are first clustered based on their objective similarity, and then a model of the variables relationships is learned in each cluster.
5. Gaussian joint modeling of objectives and variables [19,20]: In [19], a Gaussian Bayesian network is used to represent dependencies between objectives, between variables, and between objectives and variables. In [18,20],



multi-dimensional Bayesian networks, used for multi-label classification tasks [1,12], serve to model the dependencies between variables and objectives.

6. Gaussian process (GP) models to predict objective values [8,49]: The idea is to predict the Pareto-optimal set by evaluating as few designs as possible and using Gaussian processes.

Some of the methods included above are natural candidates to be applied as fitness surrogates in MOPs. They have been mainly used for generation of solutions during the sampling step and for understanding the types of relationships between the variables and objectives of the problem. Statistical methods from other areas such as quantitative genetics have also application to design model-based fitness surrogates in MOEDAs [39].

## 7 Conclusions

In this paper we have analyzed a number of critical issues related with the design and application of fitness surrogates constructed from probabilistic graphical models. A classification of model-based fitness surrogates according to the type of approximation problem they can be used for has been proposed. We have also emphasized the importance that fitness-aware learning of the probabilistic models has for taking advantage of the information available about the fitness function. We have analyzed the commonly applied methods for evaluating the quality of the model-based surrogates and discussed the usually overlooked problem of how to select a sample of solutions for evaluating the models. Finally, we have analyzed the different approaches to model-based surrogates for MOPs and identified the available methods for variable-objective mapping.

There are several areas where further work is required:

- We have focused our analysis on the use of model-based surrogates in the context of EDA applications. However, surrogate models constructed from probabilistic models can be used within other EAs like genetic algorithms [4]. The conception of methods that extend the use of probabilistic models to other EAs is a topic worth to research.
- The application of probabilistic modeling has been mainly constrained to problems with binary and continuous representation. Recent results in other domains like permutations [9,10] show that PGMs can be also applied in these domains with important gains in efficiency. Some authors have proposed the use of this type of models also as fitness surrogates [33] and more work is needed in this direction.
- Theoretical work is needed to obtain effective learning methods for model-based fitness surrogates for MOPs and for providing a taxonomy of the different approximation problems that arise in this domain.

**Acknowledgements.** This work has been partially supported by Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), TIN2010-14931, and COMBIOMED network in computational biomedicine (Carlos III Health Institute).

## References

1. Bielza, C., Li, G., Larranaga, P.: Multi-dimensional classification with Bayesian networks. *International Journal of Approximate Reasoning* 52(6), 705–727 (2011)
2. Bosman, P.A.: Design and Application of Iterated Density-Estimation Evolutionary Algorithms. PhD thesis, Universiteit Utrecht, Utrecht, The Netherlands (2003)
3. Bosman, P.A., Thierens, D.: Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms. *International Journal of Approximate Reasoning* 31(3), 259–289 (2002)
4. Brownlee, A.E.I., Regnier-Coudert, O., McCall, J.A., Massie, S., Stulajter, S.: An application of a GA with Markov network surrogate to feature selection. *International Journal of Systems Science* 44(11), 2039–2056 (2013)
5. Brownlee, A.E.I., McCall, J., Shakya, S.K.: The Markov Network Fitness Model. In: Shakya, S., Santana, R. (eds.) *Markov Networks in Evolutionary Computation*, vol. 14, pp. 125–140. Springer (2012)
6. Brownlee, A.E.I., McCall, J., Zhang, Q., Brown, D.: Approaches to selection and their effect on fitness modelling in an estimation of distribution algorithm. In: *Proceedings of the 2008 Congress on Evolutionary Computation, CEC 2008*, Hong Kong, pp. 2621–2628. IEEE Press (2008)
7. Butz, M.V., Pelikan, M., Llorá, X., Goldberg, D.E.: Automated global structure extraction for effective local building block processing in XCS. *Evolutionary Computation* 14(3), 345–380 (2006)
8. Campigotto, P., Passerini, A., Battiti, R.: Active learning of Pareto fronts. Technical Report DISI-13-001, University of Trento (2013)
9. Ceberio, J., Irurozki, E., Mendiburu, A., Lozano, J.A.: A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems. *Progress in Artificial Intelligence* 1(1), 103–117 (2012)
10. Ceberio, J., Mendiburu, A., Lozano, J.A.: The Plackett-Luce ranking model on permutation-based optimization problems. In: *2013 IEEE Congress on Evolutionary Computation (CEC)*, pp. 494–501. IEEE (2013)
11. Coello, C., Lamont, G., Van Veldhuizen, D.: *Evolutionary Algorithms for Solving Multi-objective Problems*. Springer-Verlag New York Inc. (2007)
12. de Waal, P.R., van der Gaag, L.C.: Inference and learning in multi-dimensional Bayesian network classifiers. In: Mellouli, K. (ed.) *ECSQARU 2007. LNCS (LNAI)*, vol. 4724, pp. 501–511. Springer, Heidelberg (2007)
13. Deb, K., Saxena, D.K.: On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. KanGAL Report 2005011, Kanpur Genetic Algorithms Laboratory (KanGAL). Indian Institute of Technology Kanpur (2005)
14. Echegoyen, C., Mendiburu, A., Santana, R., Lozano, J.A.: Analyzing the k most probable solutions in EDAs based on Bayesian networks. In: *Exploitation of Linkage Learning in Evolutionary Algorithms*, Evolutionary, pp. 163–189. Springer (2010)
15. Echegoyen, C., Mendiburu, A., Santana, R., Lozano, J.A.: Toward understanding EDAs based on bayesian networks through a quantitative analysis. *IEEE Transactions on Evolutionary Computation* 16(2), 173–189 (2012)
16. Höns, R., Santana, R., Larrañaga, P., Lozano, J.A.: Optimization by max-propagation using Kikuchi approximations. Technical Report EHU-KZAA-IK-2/07, Department of Computer Science and Artificial Intelligence, University of the Basque Country (November 2007)

17. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1(2), 61–70 (2011)
18. Karshenas, H.: Regularized model learning in EDAs for continuous and multiobjective optimization. PhD thesis, Technical University of Madrid, Spain (2013)
19. Karshenas, H., Santana, R., Bielza, C., Larrañaga, P.: Continuous estimation of distribution algorithms based on factorized Gaussian Markov networks. In: Shakya, S., Santana, R. (eds.) *Markov Networks in Evolutionary Computation*, pp. 157–173. Springer (2012)
20. Karshenas, H., Santana, R., Bielza, C., Larrañaga, P.: Multi-objective optimization based on joint probabilistic modeling of objectives and variables. *IEEE Transactions on Evolutionary Computation* (accepted for publication, 2013)
21. Larrañaga, P., Karshenas, H., Bielza, C., Santana, R.: A review on probabilistic graphical models in evolutionary computation. *Journal of Heuristics* 18(5), 795–819 (2012)
22. Larrañaga, P., Lozano, J.A. (eds.): *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston (2002)
23. López, A., Coello, C.A., Chakraborty, D.: Objective reduction using a feature selection technique. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008*, pp. 673–680. ACM, New York (2008)
24. Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.): *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Springer (2006)
25. Martí, L., García, J., Berlanga, A., Coello, C.A., Molina, J.M.: On current model-building methods for multi-objective estimation of distribution algorithms: Shortcomings and directions for improvement. Technical Report GIAA2010E001, Department of Informatics of the Universidad Carlos III de Madrid, Madrid, Spain (2010)
26. Mendiburu, A., Santana, R., Lozano, J.A.: Introducing belief propagation in estimation of distribution algorithms: A parallel framework. Technical Report EHU-KAT-IK-11/07, Department of Computer Science and Artificial Intelligence, University of the Basque Country (October 2007)
27. Miquélez, T., Bengoetxea, E., Larrañaga, P.: Evolutionary computation based on Bayesian classifiers. *International Journal of Applied Mathematics and Computer Science* 14(3), 101–115 (2004)
28. Mühlenbein, H., Mahnig, T.: Evolutionary algorithms: from recombination to search distributions. In: *Theoretical Aspects of Evolutionary Computing*, pp. 137–176. Springer, Berlin (2000)
29. Munetomo, M., Murao, N., Akama, K.: Introducing assignment functions to Bayesian optimization algorithms. *Information Sciences* 178(1), 152–163 (2008)
30. Oriols-Puis, Bernardó-Manilla, E., Pastry, K., Goldberg, D.E.: *Substructures surrogates for learning decomposable classification problems: Implementation and first results*, London, UK, pp. 2875–2882. ACE Press (2007)
31. Pelikan, M., Sastry, K., Goldberg, D.E.: Multiobjective estimation of distribution algorithms. In: Pelikan, M., Sastry, K., Cantú-Paz, E. (eds.) *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*. SCI, pp. 223–248. Springer, Heidelberg (2006)

32. Pereira, F.B., Machado, P., Costa, E., Cardoso, A., Ochoa, A., Santana, R., Soto, M.R.: Too busy to learn. In: Proceedings of the 2000 Congress on Evolutionary Computation, CEC 2000, La Jolla Marriott Hotel La Jolla, California, USA, pp. 720–727. IEEE Press (July 2000)
33. Regnier-Coudert, O.: Bayesian network structure learning using characteristic properties of permutation representations with applications to prostate cancer treatment. PhD thesis, Robert Gordon University (2013)
34. Rivera, J.P., Santana, R.: Design of an algorithm based on the estimation of distributions to generate new rules in the XCS classifier system. Technical Report ICIMAF 2000-100, CEMAFIT 2000-78, Institute of Cybernetics, Mathematics and Physics, Havana, Cuba (June 2000)
35. Santana, R.: An analysis of the performance of the mixture of trees factorized distribution algorithm when priors and adaptive learning are used. Technical Report ICIMAF 2002-180, Institute of Cybernetics, Mathematics and Physics, Havana, Cuba (March 2002)
36. Santana, R.: Factorized distribution algorithms: Selection without selected population. In: Proceedings of the 17th European Simulation Multiconference, ESM-2003, Nottingham, England, pp. 91–97 (2003)
37. Santana, R., Armañanzas, R., Bielza, C., Larrañaga, P.: Network measures for information extraction in evolutionary algorithms. *International Journal of Computational Intelligence Systems* 6(6), 1163–1188 (2013)
38. Santana, R., Bielza, C., Larrañaga, P., Lozano, J.A., Echegoyen, C., Mendiburu, A., Armañanzas, R., Shakya, S.: Mateda-2.0: A MATLAB package for the implementation and analysis of estimation of distribution algorithms. *Journal of Statistical Software* 35(7), 1–30 (2010)
39. Santana, R., Karshenas, H., Bielza, C., Larrañaga, P.: Quantitative genetics in multi-objective optimization algorithms: From useful insights to effective methods. In: Proceedings of the 2011 Genetic and Evolutionary Computation Conference, GECCO 2011, Dublin, Ireland, pp. 91–92 (2011)
40. Santana, R., Shakya, S.: Probabilistic graphical models and Markov networks. In: Shakya, S., Santana, R. (eds.) *Markov Networks in Evolutionary Computation*, pp. 3–19. Springer (2012)
41. Sastry, K., Pelikan, M., Goldberg, D.: Efficiency enhancement of genetic algorithms via building-block-wise fitness estimation. In: Proceedings of the 2004 Congress on Evolutionary Computation, CEC 2004, Portland, Oregon, pp. 720–727. IEEE Press (2004)
42. Shakya, S., McCall, J., Brown, D.: Using a Markov network model in a univariate EDA: An empirical cost-benefit analysis. In: Beyer, H.-G., O'Reilly, U.-M. (eds.) *Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2005*, Washington, D.C, USA, pp. 727–734. ACM Press (2005)
43. Shakya, S., Santana, R. (eds.): *Markov Networks in Evolutionary Computation*. Springer (2012)
44. Shi, L., Rasheed, K.: A survey of fitness approximation methods applied in evolutionary algorithm. In: Tenne, Y., Goh, C.-K. (eds.) *Computational Intelligence in Expensive Optimization Problems*, pp. 3–28. Springer (2010)
45. Ulrich, T., Brockhoff, D., Zitzler, E.: Pattern identification in Pareto-set approximations. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008, pp. 737–745. ACM, New York (2008)

46. Valdez-Peña, I.S., Hernández-Aguirre, A., Botello-Rionda, S.: Approximating the search distribution to the selection distribution in EDAs. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2009, pp. 461–468. ACM, New York (2009)
47. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11(6), 712–731 (2007)
48. Zhou, A., Zhang, Q., Jin, Y.: Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation* 13(5), 1167–1189 (2009)
49. Zuluaga, M., Sergeant, G., Krause, A., Püschel, M.: Active learning for multi-objective optimization. In: Proceedings of the 30th International Conference on Machine Learning (ICML 2013), pp. 462–470 (2013)