Journal of Heuristics (2021) 27:549–573
https://doi.org/10.1007/s10732-021-09469-x

Analysis of Bayesian Network Learning Techniques for a Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm: a case study on MNK Landscape

Marcella S. R. Martins[1] · Mohamed El Yafrani[2] · Myriam Delgado[1] · Ricardo Lüders[1] · Roberto Santana[3] · Hugo V. Siqueira[1] · Huseyin G. Akcay[4] · Belaïd Ahiod[5]

Received: 5 November 2019 / Revised: 9 October 2020 / Accepted: 11 January 2021 /
Published online: 6 February 2021

Abstract

This work investigates different Bayesian network structure learning techniques by thoroughly studying several variants of Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA), applied to the MNK Landscape combinatorial problem. In the experiments, we evaluate the performance considering three different aspects: optimization abilities, robustness and learning efficiency. Results for instances of multi- and many-objective MNK-landscape show that, score-based structure learning algorithms appear to be the best choice. In particular, HMOBEDA_{k2} was capable of producing results comparable with the other variants in terms of the runtime of convergence and the coverage of the final Pareto front, with the additional advantage of providing solutions that are less sensible to noise while the variability of the corresponding Bayesian network models is reduced.

Keywords Many-objective optimization · Estimation of distribution algorithms · Structure learning techniques · Robustness

1 Introduction

According to Bennett and Parrado-Hernández (2006), the fields of machine learning and mathematical programming are increasingly intertwined. We have observed a great synergy between them in the past few years, with optimization problems taking place at the heart of most machine learning approaches and machine learning being used to improve several optimization algorithms.

✉ Marcella S. R. Martins
marcella@utfpr.edu.br

Extended author information available on the last page of the article

Estimation of distribution algorithm (EDA) (Mühlenbein and Paab 1996) is a class of evolutionary algorithm (EA) that explores the search space by building a probabilistic model from a set with the current best candidate solutions. Since new solutions are sampled from the probabilistic model, evolution is guided toward more promising areas of the search space. Playing a central role in the connection between optimization heuristics and machine learning approaches, EDAs based on Probabilistic graphical models (PGMs) (Lauritzen 1996) combine evolutionary optimization with graph and probability theories giving rise to powerful optimizers based on mathematical foundations for multivariate statistical modeling. PGMs are widely used in evolutionary optimization, especially in EDAs when interactions among variables are considered (Multivariate EDAs). EDAs based on PGM have gained attention from the evolutionary optimization community as they can provide more useful information about decision variables compared to other EAs.

EDAs have achieved a good performance for several problems including environmental monitoring network design (Kollat et al. 2008), protein side chain placement problem (Santana et al. 2008) and table ordering (Bengoetxea et al. 2011). They have also been applied to solve the multi-objective Knapsack Problem (Shah and Reed 2011), multi-objective optimization problems (MOPs) in a noisy environment (Shim et al. 2013) and combinatorial many-objective optimization problems (MaOPs) (Martins et al. 2018). Usually they integrate both the model building and sampling techniques into evolutionary optimizers using special selection schemes (Khan et al. 2002). Recently, the role of the probabilistic model has been extended to model the dependencies between variables and objectives (Karshenas et al. 2014). In addition, EDAs can be notably enhanced by adding a local optimizer that can refine the solutions found by sampling from the PGM (Marti et al. 2008; Martins et al. 2017, 2018).

In this work, we study this type of enhancement in the context of many-objective optimization (Ribeiro et al. 2020), investigating the approach called Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA) (Martins et al. 2018, 2017), using a PGM based on the joint probabilistic modeling of decision variables, objectives, and parameters of the local optimizer. Structure learning methods have been extensively studied in Cooper and Herskovits (1992), Tsamardinos et al. (2003), Tsamardinos et al. (2006), Santhanam and Wainwright (2012) resulting in several algorithms in various settings. However, most of these works are focused on learning the PGM structure while ignoring the performance of the optimization algorithm.

The main goal in this paper is to investigate PGM structure learning techniques considering the data provided by several variants of an optimization algorithm (HMOBEDA). Each variant results from a different Bayesian Network (BN) learning method. The data considered here are the best candidate solutions obtained during the evolutionary process from each variant. Exploring the structure learning algorithms can lead to more efficient methods considering the differences in the structural information they capture and their sensibility to noisy solutions. One of the main contributions of this paper is the analysis of the behavior and performance of the algorithms variants considering three different aspects: multi-criteria optimization, robustness and learning capabilities. We aim to evaluate the learning algorithms considering (i) optimization performance based on run time, convergence and coverage of the final Pareto fronts; (ii) robustness based on sensibility to noise; (iii) learning capacity based on accuracy to recover the problem structure (for instance, in terms of the Structural Hamming Distance (SHD) (Tsamardinos et al. 2006) from the target model). This analysis is particularly novel and relevant since we address many-objective optimization problems, and HMOBEDA is one of few algorithms that learns and exploits relationships between objectives and variables as well as the parameters of an embedded local search procedure.

In this work, we compare the K2 algorithm (Cooper and Herskovits 1992), the Hill-Climbing using K2-metric (HC-K2) (Moran et al. 2009), the Incremental Association Markov Blanket (IAMB) (Tsamardinos et al. 2003), the PC-algorithm (PC) (Colombo and Maathuis 2014) and the Max-Min Hill Climbing (MMHC) (Tsamardinos et al. 2006). These methods are applied to the Bayesian networks modeling phase in HMOBEDA. The idea is to contrast different score-based, constraint-based and hybrid learning techniques when applying PGM in the context of multi-objective optimization. For this, we address a combinatorial problem, namely the Multi-objective NK-landscape (MNK) model which has been recently explored in other works in the literature (Aguirre and Tanaka 2007; Santana et al. 2015). In particular, EDAs that use different types of probabilistic models, including Bayesian networks, have already been investigated for MNK problems (Martins et al. 2018, ?). The hypervolume and Inverted Generational Distance (IGD) indicators are considered for the statistical analysis of the results.

This paper is organized as follows. Section 2 provides a brief introduction to Multi-objective optimization, Bayesian Network concepts and the addressed MNK-Landscape model. Section 3 details the HMOBEDA. Results from numerical experiments are shown and discussed in Sect. 4 with conclusions and future directions presented in Sect. 5.

2 Background

This section presents some basic concepts and background information about the main topics addressed in this paper. Thus, a review on multi-objective optimization and Bayesian networks is presented, and the addressed benchmarking problem is revisited.

2.1 Multi-objective optimization

Real-world problems are generally characterized by several competing objectives. While in the case of single-objective optimization one optimal solution is usually required to solve the problem (Puchta et al. 2016, 2020; Santos et al. 2017), this is not true in multi-objective optimization (Ribeiro et al. 2020). The standard approach to solve this difficulty lies in finding all possible trade-offs among the multiple, competing objectives.

A general MOP includes decision variables, objective functions, and constraints, where objective functions and constraints are functions of the decision variables (Zitzler and Thiele 1999). Mathematically, a maximization MOP can be defined as:

$$
\begin{aligned}
&\max_{\mathbf{x}}\ \mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_R(\mathbf{x}))\\
&\text{subject to}\\
&\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_k(\mathbf{x})) \leq 0,\\
&\mathbf{x} = (x_1, x_2, \ldots, x_Q) \in X,\\
&\mathbf{z} = (z_1, z_2, \ldots, z_R) \in Z,
\end{aligned}
\tag{1}
$$

where $\mathbf{x} = (x_1, \ldots, x_Q)$ is a Q-dimensional decision variable vector defined in a universe X; $\mathbf{z}$ is the objective vector, with R objectives, where each $f_r(\mathbf{x})$ is a single-objective function, Z is the objective space and $\mathbf{h}(\mathbf{x}) \leq 0$ is the set of constraints which determines a set of feasible solutions X_f. When R is greater than three, the problem is referred to as a Many Objective Optimization Problem (MaOP). These problems are usually more challenging than the problems with $R \leq 3$ due to the higher dimensionality of the objective space and the existence of many conflicting objective functions.

The set of MOP and MaOP solutions includes decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another—these decision vectors are called the Pareto optimal set. The idea of Pareto optimally is based on the Pareto dominance. In a maximization problem, a solution $\mathbf{u}$ dominates a solution $\mathbf{v}$ if $f_r(\mathbf{u}) \geq f_r(\mathbf{v})$ for all $r \in \{1, 2, 3, \ldots, R\}$, and $f_r(\mathbf{u}) > f_r(\mathbf{v})$ for some $r \in \{1, 2, 3, \ldots, R\}$. A solution is Pareto optimal if it is not dominated by any other feasible solution.

The set of non-dominated solutions (the Pareto set) lies, in the objective space, on a surface known as Pareto optimal front. The goal of the optimization is to find a representative set of solutions with the corresponding objective vectors along the Pareto optimal front.

Generating the Pareto set can be computationally expensive and it is often infeasible due to the computational complexity of the problems. For this reason, a number of stochastic search strategies such as Evolutionary algorithms and Estimation of distribution algorithms have been developed. These approaches usually do not guarantee the identification of optimal trade-offs, instead, they try to find a good approximation. Because these algorithms are population-based, they are able to approximate the whole Pareto front of a MOP in a single run.

2.2 Bayesian networks

Claimed as a paradigm shift in the field of evolutionary computation, EDAs employ explicit probability distributions (Larrañaga and Lozano 2001). Among the most general probabilistic models for discrete variables used in EDAs are Bayesian networks. In this sub-section, we briefly describe some techniques used for learning the structure of Bayesian networks.

Bayesian networks are addressed in this paper for modeling multinomial data with discrete variables and generate new solutions using the particular conditional probability (Henrion 1988) described by Eq. 2:

$$
p(y_m^k | \mathbf{pa}_m^{j,B}) = \theta_{y_m^k | \mathbf{pa}_m^{j,B}} = \theta_{mjk} \tag{2}
$$

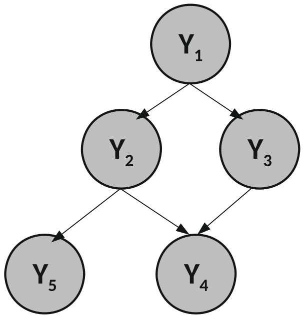


Fig. 1 Bayesian network structure example with $M = 5$ random variables

where $\mathbf{Y} = (Y_1, \ldots, Y_M)$ is a vector representation of M random variables and y_m the m-th component of it; B is the structure and Θ a set of local parameters; $\mathbf{Pa}_m^B$ represents the set of parents of the variable Y_m, where $\mathbf{pa}_m^{j,B} \in \{\mathbf{pa}_m^{1,B}, \ldots, \mathbf{pa}_m^{t_m,B}\}$ denotes a particular combination of values for $\mathbf{Pa}_m^B$, t_m is the total number of different possible instantiations of the parent variables of Y_m given by $t_m = \prod_{Y_v \in \mathbf{Pa}_m^B} s_v$, s_v is the total of possible values (states) that Y_v can assume. The parameter θ_{mjk} represents the conditional probability that variable Y_m takes its $k-$th value (y_m^k), knowing that its parent variables have taken their j-th combination of values ($\mathbf{pa}_m^{j,B}$). This way, the parameter set is given by $\Theta = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m, \ldots \boldsymbol{\theta}_M\}$, where $\boldsymbol{\theta}_m = (\theta_{m11}, \ldots, \theta_{mjk}, \ldots, \theta_{m,t_m,s_m})$ and M is the total number of nodes in the BN. Figure 1 represents a Bayesian network whose structure B is defined by a directed graph with $M = 5$ nodes representing 5 random variables. In this example the parents of all variables are given by $\mathbf{Pa}_1^B = \emptyset$, $\mathbf{Pa}_2^B = \{Y_1\}$, $\mathbf{Pa}_3^B = \{Y_1\}$, $\mathbf{Pa}_4^B = \{Y_2, Y_3\}$ and $\mathbf{Pa}_5^B = \{Y_2\}$. Assuming that each variable Y_m could assume values {0,1}, we would have $s_v = 2, \forall v$. Considering, for example, the specific variable Y_4, we would have $t_4 = 4$, then $\boldsymbol{\theta}_4 = (\theta_{4(00)0}, \theta_{4(01)0}, \theta_{4(10)0}, \theta_{4(11)0}, \theta_{4(00)1}, \theta_{4(01)1}, \theta_{4(10)1}, \theta_{4(11)1})$, where $\theta_{4(j_2 j_3)k}$ is the conditional probability that variable Y_4 takes its $k-$th value (0 or 1), knowing that its parent variables have taken ($j_2 j_3$) combination of values. Therefore, for all nodes in Fig. 1 a total of 22 conditional probabilities should be estimated: 2 for Y_1, 8 for Y_4 and 4 for each remaining variable.

The parameters of Θ and B are usually unknown, and to determine them the literature presents two possibilities: Maximum Likelihood Estimate (MLE) and Bayesian Estimate. In this work, we address the last method.

In terms of BN structures learning process, some authors proposed different methods. We highlight three approaches: score-based learning, constraint-based learning, and hybrid methods (Yuan and Malone 2013).

Score-based techniques apply heuristic optimization methods to sort the structures selecting the one which maximizes the value of a scoring metric. The simple greedy search algorithm, local hill-climbing, simulated annealing, tabu search, K2 algorithm and evolutionary computation are important representatives of this class. In what follows we will discuss two scoring metrics to be used with Score-based techniques.

According to Scanagatta et al. (2019), one of the most adopted scores is the Bayesian Dirichlet equivalence (BDe) metric (Buntine 1991; Cooper and Herskovits 1992;

Heckerman et al. 1995), which measures the posterior probability of a chosen BN given the available data. It assumes the Dirichlet distribution (with parameters α_{mjk}) as prior probability and a uniform prior distribution of all possible structures (Larrañaga et al. 2012).

The BDe metric is defined by Eq. 3:

$$p(B|Pop) = \prod_{m=1}^{M} \prod_{j=1}^{t_m} \frac{\Gamma(\alpha_{mj})}{\Gamma(\alpha_{mj} + N_{mj})} \prod_{k=1}^{s_m} \frac{\Gamma(\alpha_{mjk} + N_{mjk})}{\Gamma(\alpha_{mjk})} \tag{3}$$

where N_{mjk} is the number of observations in the data set Pop for which Y_m assumes the k-th value given the j-th combination of values from its parents, with $\mathbf{N}_{mj} = \{N_{mj1}, \ldots, N_{mjs_m}\}$, $\Gamma(x) = (x-1)!$ and $\alpha_{mjk} = \sum_{k=1}^{s_m} \alpha_{mjk}$. The product over $j \in \{1, \ldots, t_m\}$ runs for all combinations of parents of Y_m and the product over $k \in \{1, \ldots, s_m\}$ runs for all possible values of Y_m. The Dirichlet parameter α_{mjk} stands for prior information about the number of instances that have Y_m set to its k-th value and the set of parents of Y_m is instantiated to its j-th combination. In the so-called K2 metric (Cooper and Herskovits 1992) for instance, parameters α_{mjk} are set to 1 as there is no prior information about the problem, and Eq. 3 reduces to Eq. 4:

$$p(B|Pop) = \prod_{m=1}^{M} \prod_{j=1}^{t_m} \frac{(s_m - 1)!}{(N_{mj} + s_m - 1)!} \prod_{k=1}^{s_m} (N_{mjk})! \tag{4}$$

Constraint-based learning methods typically use statistical tests to identify conditional independence relations from the data and build a BN structure that best fits those relations. Some examples are the Incremental Association Markov Blanket (IAMB) (Tsamardinos et al. 2003) and PC-Stable implementation algorithm (PC) (Colombo and Maathuis 2014), the most commonly used among constraint-based ones (Scutari et al. 2018).

Hybrid methods combine the two approaches: it uses conditional independence tests to reduce the search space, and at the same time, it applies network scores to find out the optimal network structure. An important algorithm of this class is the Max-Min Hill Climbing (MMHC) (Tsamardinos et al. 2006), in which constraint-based learning is used to create a skeleton graph and the score-based is addressed to find a high-scoring network structure, a subgraph of the skeleton.

2.3 MNK-landscape problem

The single *NK fitness landscapes* is a family of combinatorial problems proposed in Kauffman (1993) aiming at exploring the way in which the neighborhood structure and the strength of the interactions between neighboring variables (subfunctions) are linked to the search space ruggedness.

Let $\mathbf{X} = (X_1, \ldots, X_N)$ denote a vector of discrete variables and $\mathbf{x} = (x_1, \ldots, x_N)$ an assignment to the variables.

An NK fitness landscape is defined by the following components:

Table 1 Example of subfunction assignments for the neighborhood structure shown in Fig. 2 ($r = 1$)

$f_{1,q}(x_q, \Pi_1(x_q))$	Values of $(x_q, \Pi_1(x_q))$							
	000	001	010	011	100	101	110	111
$f_{1,1}(x_1, x_3, x_4)$	0.58	0.18	0.74	0.12	0.97	0.42	0.16	0.71
$f_{1,2}(x_2, x_5, x_6)$	0.23	0.17	0.13	0.95	0.17	0.15	0.16	0.31
$f_{1,3}(x_3, x_2, x_4)$	0.94	0.34	0.28	0.31	0.97	0.03	0.87	0.97
$f_{1,4}(x_4, x_2, x_5)$	0.01	0.51	0.06	0.67	0.31	0.47	0.57	0.41
$f_{1,5}(x_5, x_1, x_6)$	0.81	0.74	0.61	0.30	0.71	0.54	0.71	0.21
$f_{1,6}(x_6, x_1, x_3)$	0.03	0.13	0.17	0.94	0.17	0.18	0.27	0.08

- Number of variables, N.
- Number of neighbors per variable, K.
- A set of neighbors, $\Pi(X_q) \in \mathbf{X}$, for X_q, $q \in \{1, \ldots, N\}$ where $\Pi(X_q)$ contains K neighbors.
- A subfunction f_q defining a real value for each combination of values of X_q and $\Pi(X_q)$, $q \in \{1, \ldots, N\}$.

Both the subfunction f_q for each variable X_q and the neighborhood structure $\Pi(X_q)$ are randomly set.

For a set of given parameters, the problem consists in finding the global maximum of the function $z_{NK}(\mathbf{x})$.

The MNK-landscape problem is a multi-objective version of the NK fitness landscape model with R objectives (Aguirre and Tanaka 2004), $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_R(\mathbf{x})) : \mathcal{B}^N \rightarrow \mathcal{R}^R$. Each objective function is determined by a different instance of an NK-landscape, over the same binary string $\mathbf{x}$, where N is the number of variables[1], R is the number of objectives, $z_r(\mathbf{x})$ is the r-ith objective function, and $\mathcal{B} = \{0, 1\}$. $\mathbf{K} = \{K_1, \ldots, K_R\}$ is a set of integers where K_r is the neighborhood size in the r-th landscape.

The MNK-landscape problem can be formulated as follows:

$$
\begin{aligned}
&\max_{\mathbf{x}}\ \mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), \ldots, z_R(\mathbf{x}))\\
&\text{subject to } \mathbf{x} \in \{0, 1\}^N,\\
&\\
&\text{with}\\
&z_r(\mathbf{x}) = \frac{1}{N}\sum_{q=1}^{N} f_{r,q}(x_q, \Pi_r(x_q)),\\
&r \in \{1, \ldots R\},\\
&q \in \{1, \ldots N\},
\end{aligned}
\tag{5}
$$

where the fitness contribution $f_{r,q}$ of variable x_q is a real number in [0, 1] drawn from a uniform distribution.

Table 1 presents an example of sub-function values for a particular landscape for one specific objective ($r = 1$, for example), with $N = 6$ and $K_1 = 2$ with a neighborhood structure shown in Fig. 2.

[1] The number of variables N is noted Q in Sects. 3 and 4.

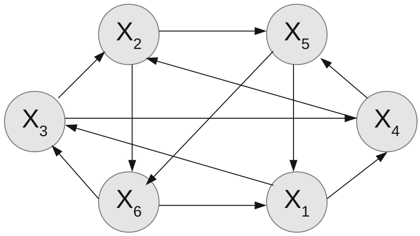


Fig. 2 Example of neighborhood structure ($r = 1$)

In this particular example the objective function for one objective among others ($r = 1$) in a multi-objective optimization can be expressed by $z_1(\mathbf{x}) = \frac{1}{N}(f_{1,1}(x_1, x_3, x_4) + f_{1,2}(x_2, x_5, x_6) + f_{1,3}(x_3, x_2, x_4) + f_{1,4}(x_4, x_2, x_5) + f_{1,5}(x_5, x_1, x_6) + f_{1,6}(x_6, x_1, x_3))$. Suppose $\mathbf{x} = (0, 1, 1, 0, 0, 1)$, the objective function for this objective is represented by $z_1(\mathbf{x}) = \frac{1}{6}(f_{1,1}(0, 1, 0) + f_{1,2}(1, 0, 1) + f_{1,3}(1, 1, 0) + f_{1,4}(0, 1, 0) + f_{1,5}(0, 0, 1) + f_{1,6}(1, 0, 1))$, and considering the subfunctions from Table 1, it results in $z_1(\mathbf{x}) = \frac{1}{6}(0.74 + 0.15 + 0.87 + 0.06 + 0.74 + 0.18) = 0.46$. More details and examples about the MNK-landscape model can be found in Santana et al. (2015).

3 Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA)

HMOBEDA is a hybrid EDA approach introduced in Martins et al. (2016) [2]. The term hybrid refers to the inclusion of local search (LS) mechanisms into its PGM-based framework to improve the performance and allow the algorithm to better refine the search process.

HMOBEDA uses a probabilistic model based on Bayesian networks for the objectives, variables and local search parameters to sample new individuals. Therefore, every solution is represented by a joint vector containing $Q + R + L$ elements denoted $\mathbf{y} = (\mathbf{x}, \mathbf{z}, \mathbf{p}) = (X_1, \ldots, X_Q, Z_1, \ldots, Z_R, P_1, \ldots, P_L)$, where $(X_1, \ldots, X_Q)$ are the decision variables, $(Z_1, \ldots, Z_R)$ are the objectives and $(P_1, \ldots, P_L)$ are the LS parameters. The general scheme of the HMOBEDA is presented in Fig. 3.

3.1 HMOBEDA main steps

In the context of the addressed MNK-landscape problem, the *Initialization* phase loads the problem instance for a given M, N and K (both the subfunctions and the neighborhood structure are obtained from a uniform distribution) and randomly generates an initial population. Each solution is a binary string of size $N = Q$ and the corresponding objectives are calculated through the MNK-landscape model.

A *Local Search* based on the Hill Climbing procedure is used to generate a neighborhood for each solution at each iteration. The best found solution is updated at each iteration once a neighboring solution with a better fitness is found (first improvement strategy).

[2] The source codes are available at https://bitbucket.org/marcella_engcomp/hmobeda.

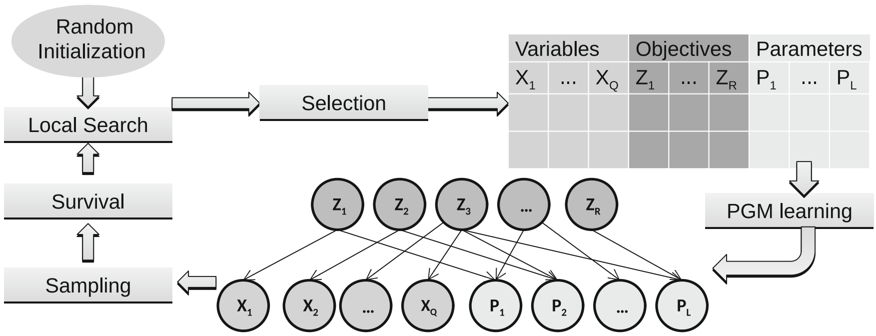


Fig. 3 The HMOBEDA framework

In order to select a total of N_{PGM} individuals from the current population, the Non-dominated Sorting (ND) (Srinivas and Deb 1994) technique is applied. After ND organizes the population based on a set of PFs (the second PF is dominated by the first, the third by the second and so on...), the *Selection* procedure randomly selects two solutions and the one positioned in the best front is chosen. If they lie on the same front, it chooses that solution with the greatest crowding distance (Deb et al. 2002).

Aiming to learn the probabilistic model, the BN structure and parameters are estimated in the *PGM Learning* block. Different algorithms can be considered: in this work we compare several structure learning algorithms running over the set of N_{PGM} best individuals. This way the BN structure encodes a factorization of the joint probability distributions or the probability mass function (pmf) given by:

$$p(\mathbf{y}) = \prod_{r=1}^{R} p(z_r|\mathbf{pa}_r^B). \prod_{q=1}^{Q} p(x_q|\mathbf{pa}_q^B). \prod_{l=1}^{L} p(p_l|\mathbf{pa}_l^B) \tag{6}$$

where $\mathbf{pa}_r^B$, $\mathbf{pa}_q^B$ and $\mathbf{pa}_l^B$ represent combinations of values for the parents of objective, decision variable and LS parameter nodes respectively, with $\mathbf{Pa}_q^B \subseteq \{Z_1, \ldots Z_R\}$, $\mathbf{Pa}_l^B \subseteq \{Z_1, \ldots Z_R\}$, $\mathbf{Pa}_r^B = \emptyset$, which means $p(z_r|\mathbf{pa}_r^B) = p(z_r)$ for $r = 1, \ldots, R$. Therefore, according to Eq. 6 and Fig. 3, the BN adopted in HMOBEDA is a naive model which does not consider arcs among variables Z, X or P. Other models could be adopted like the Hierarchical Bayesian Optimization Algorithm (Pelikan et al. 2003) which has been reported to be more suitable to deal with dependencies between variables, objectives and to automatically control the application of local search operators. However, expanding the PGM model is out of the scope of this paper.

In the *Sampling* block, the obtained PGM is used to sample the set of new individuals. As discussed in Martins et al. (2017), Martins et al. (2018), the main advantage of using the HMOBEDA framework, is that not only decision variables, but also LS parameters can be obtained through the *Sampling* block. Note that a naive Bayesian model is adopted to facilitate the sampling process: fixing objective values as target evidences enables the estimation of their associated decision variables and LS parameters. Therefore, after sampling, decision variables $(X_1, \ldots, X_Q)$ and LS parameters

$(P_1, \ldots, P_L)$ more related to the objectives fixed as evidences can be drawn for each new individual.

HMOBEDA considers evidences fixed as combinations (all of them with the same probability of occurrence) of maximum and minimum values for the objectives, i.e., the ideal point Z^*[3] plus the estimated extreme points of the current approximation of the Pareto front. These values are uniformly distributed according the number of objectives in each generation (excluding the combination with minimum values for all objectives).

Finally, objectives $(Z_1, \ldots, Z_R)$ are calculated based on the fitness function (in the case of surrogate assisted approaches, PGM can also be used to sample the objective values or the least squares method can provide objective value approximations). The union of the sampled and the current populations in the *Survival* block is used to create the new population for the next generation, and the main loop continues until the stop condition is achieved.

In this paper, we present seven variants of HMOBEDA using different score, constraint and hybrid-based PGM learning techniques: HMOBEDA$_{k2}$, HMOBEDA$_{hc-k2}$, HMOBEDA$_{iamb}$, HMOBEDA$_{pc}$, HMOBEDA$_{mmhc}$, HMOBEDA$_{sparse}$ and HMOBEDA$_{tabu}$. HMOBEDA$_{k2}$ uses the score based K2 algorithm as the structure learning technique; HMOBEDA$_{hc-k2}$ is also score-based which considers Hill-Climbing using K2-metric (HC-K2); HMOBEDA$_{iamb}$ and HMOBEDA$_{pc}$ use constraint-based methods applying, respectively, PC-Stable implementation algorithm (PC) and Incremental Association Markov Blanket (IAMB); HMOBEDA$_{mmhc}$ is a hybrid approach based on Max-Min Hill Climbing (MMHC) algorithm and in this paper we apply K2 as its score metric; HMOBEDA$_{sparse}$ is score-based approach based on sparse regularization (Aragam et al. 2019) and HMOBEDA$_{tabu}$ is also a score-based which uses Tabu as a search strategy (Russell Stuart and Norvig 2009). HMOBEDA$_{sparse}$ and HMOBEDA$_{tabu}$, both, consider BDe as their score metric (Heckerman et al. 1995).

The choice of these structure learning approaches is based on their adoption in the recent literature (Martins et al. 2017; Ding and Zhuang 2018; Scutari et al. 2018; Tsagris 2019). They also represent examples of popular and state-of-the-art score-based, constraint-based and hybrid approaches of structure learning algorithms according to Scutari (2009).

In this paper, every new BN structure is learned from scratch at each iteration. In the future, we can explore strategies using a previous BN to improve the computational time such as in Pelikan et al. (2008). Moreover, metrics and procedures can be used to evaluate the accuracy of the model with respect to a reference underlying problem structure (Pelikan and Hauschild 2012; Lima et al. 2011; Echegoyen et al. 2007; Brownlee et al. 2012). Although most authors agree that capturing important problem interactions is beneficial for the model, others acknowledge this accuracy but consider that complex models might be inefficient for the evolutionary search (Echegoyen et al. 2011). Note that our BN model includes not only the objectives and variables of the

[3] Usually high values for maximization problems: the ideal point Z^* is the maximum value of each objective achieved so far.

problem, but also the parameters of the algorithm. Therefore, the learning of problem structure can be considered in the future as well.

4 Experiments and results

In this section, we compare HMOBEDA variants obtained from BN structure learning algorithms discussed in the previous section: score-based (HMOBEDA$_{k2}$, HMOBEDA$_{hc-k2}$, HMOBEDA$_{sparse}$ and HMOBEDA$_{tabu}$), constraint-based (HMOBEDA$_{iamb}$ and HMOBEDA$_{pc}$), and a score-constraint-based hybrid method (HMOBEDA$_{mmhc}$).

The comparison is performed over MNK-landscape instances using the Hypervolume (HV$^-$) indicator and the IDG metric. The traditional Hypervolume (HV) measures the space (considering a reference point) which is dominated by at least one point in the approximated Pareto front provided by a solution set of an algorithm, while HV$^-$ considers the difference between the hypervolume from both the solution set and the reference set. The IGD metric is the average distance from every point in the Pareto front associated with the reference set to the nearest point in the approximated Pareto front. So, smaller values of HV$^-$ and IGD correspond to high quality solutions of the non-dominated sets in terms of convergence (measured by HV$^-$) and coverage (measured by IGD) when compared with the reference set. The Pareto optimal front for each instance of the addressed problem is not known. Therefore, we use a reference set which is constructed by gathering all non-dominated solutions from all the HMOBEDA variants over all executions.

MNK-landscape instances are sampled using different ruggedness factors $K \in \{2, 4, 6, 8, 10\}$, number of objectives $R \in \{2, 3, 5, 8\}$ and sizes $N = Q \in \{20, 50, 100\}$. A total of 60 instances are generated, one for each combination of K, R and N. The same strategy is adopted in Martins et al. (2018).

As shown in previous works (Martins et al. 2018, 2017, 2016, 2017), the BN parameters Θ are estimated by Bayesian Estimate using Dirichlet prior. For all HMOBEDA variants, structure learning algorithms set parent nodes as objectives in the Bayesian network. This is because, fixing the objective values as target evidences allows a straightforward estimation of their associated decision variables and LS parameters.

The parameters for all HMOBEDA variants considered in this section are: population size $Pop = 100$, number of selected individuals $N_{PGM} = Pop/2$ for building the PGM, and number of sampled individuals $N_{smp} = 10 * N$. The LS online configuration of HMOBEDA variants during the evolution has the following elements in vector **p**: number of LS iterations $N_{iter} \in \{5, 6 \ldots, 20\}$ i.e., 16 possible discrete values between 5 and 20; type of neighbor fitness calculation $T_{Fnbh} \in \{1, 2\}$ with 1 being a linear combination of objectives and 2 being alternating objectives (i.e., one by one for each LS iteration); neighborhood type $T_{nbh} \in \{1, 2\}$ with 1 corresponding to defining double bit-flip operator and 2 being the single bit-flip from 0 to 1. These parameters have been defined experimentally in previous works (Martins et al. 2017, 2016).

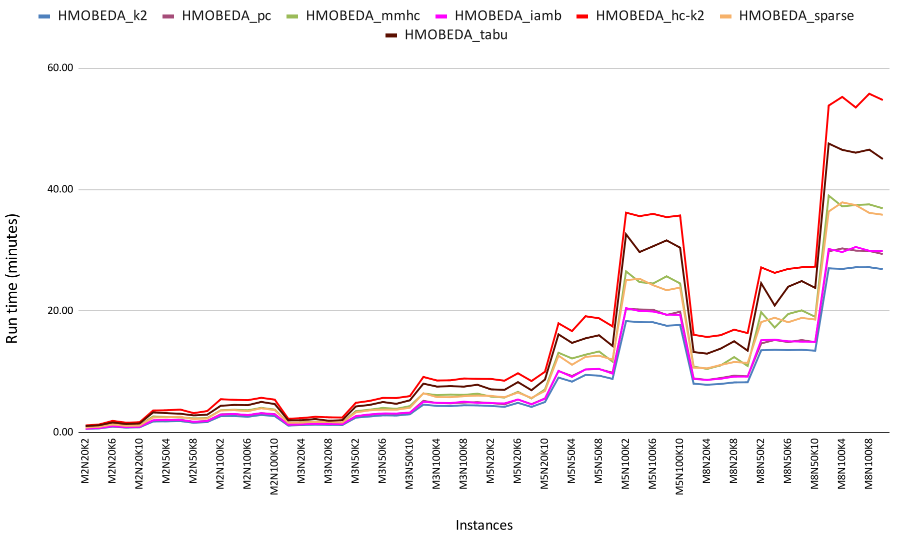


Fig. 4 Average run-times (min) for each instance and HMOBEDA variant

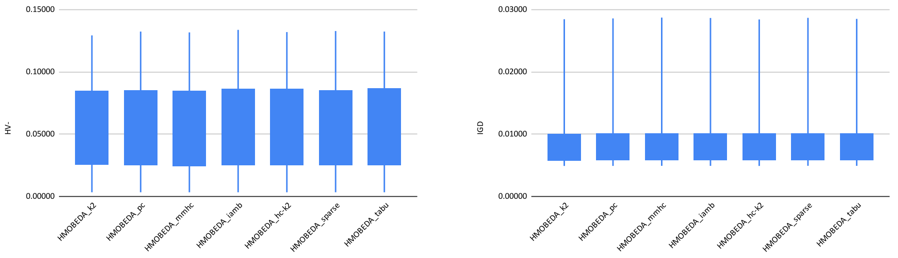
The stopping condition is the maximum number of fitness evaluations (Max_{eval}) including repair procedures and LS iterations. All algorithms stop when the total number of fitness computations achieves $200,000$ evaluations. A total of 30 independent executions of each algorithm is performed for each instance of the addressed problem to get average performance metrics.

Shapiro-Wilk normality test is used to verify whether the performance metric results are normally distributed and the analysis of variance test (ANOVA) is used in this case. In the case of non-normal distribution, Kruskal-Wallis (de Mattos Neto et al. 2020) and Dunn-Sidak's post-hoc tests are considered when performing multiple comparisons. Mann-Whitney-Wilcoxon (Santana et al. 2019) test is used when only two approaches are compared.

4.1 Optimization performance: multicriteria analysis

In this section, we aim to compare the learning structure algorithm considering three different performance criteria: computational cost, PF convergence and PF coverage, measured by run-times, HV^{-} and IGD, respectively.

The average run-times (min) for each instance and variant are presented in Fig. 4. They are averaged over 30 executions of the variant in each instance. We notice that score-based algorithms present the lowest (for HMOBEDA_{k2}) and the highest (for HMOBEDA_{hc-k2}) computational run-times. The Kruskal-Wallis and Dunn-Sidak's post-hoc tests are applied with a significance level $\alpha = 5\%$, indicating that, for almost all instances, the differences are statistically significant. Note that Fig. 4 shows run-times increase according to the instance complexity as expected for all algorithms, but HMOBEDA_{k2} requires less computational effort than other algorithms.



(a) **(b)**

Fig. 5 Boxplot of **a** HV$^-$ and **b** IGD values averaged over 60 instances for each HMOBEDA variant

Aiming to evaluate if the lowest computational cost required by HMOBEDA$_{k2}$ yields to poor performance in terms of convergence and coverage, we conducted experiments comparing the hypervolume and distances of the resulting PFs with the reference PF. Figure 5a, b show the boxplots of hypervolume difference HV$^-$ and IGD metric values averaged over 60 instances for each HMOBEBA variant. These plots show that HMOBEDA$_{k2}$ has a similar performance when compared to the others with regards to HV$^-$ and IGD.

In addition, by considering each individual instance with 30 independent executions of each algorithm, the Kruskal-Wallis and Dunn-Sidak's post-hoc tests with a significance level $\alpha = 5\%$ show that there is no statistically significant differences from HMOBEDA$_{k2}$ to the others for most instances regarding both HV$^-$ indicator and IGD metric.

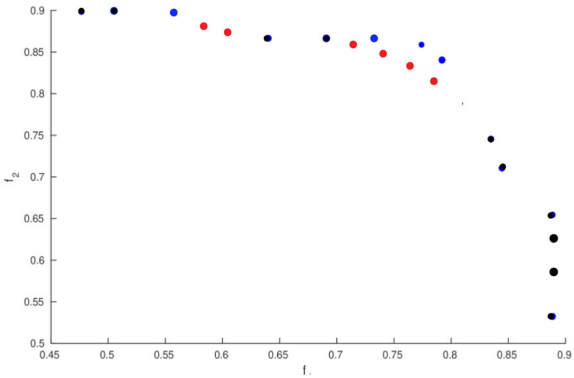
Based on the previous results, under multicriteria analysis, we conclude that HMOBEDA$_{k2}$ is the best option because it dominates the others in one criterion (run-times) without degenerating the other two (convergence and coverage).

4.2 Robustness performance: sensibility analysis

In this section, we are interested in verifying the capability of the BN models to provide solutions less sensitive to noise. In this work, a solution is robust if small perturbations around it do not affect significantly the corresponding objective values (Meneghini et al. 2016). In this paper, this is accomplished by a sensibility analysis performed by (i) sampling a set of solutions coded into the final BN model for each execution and (ii) adding noise to them to evaluate the new Pareto front. Robust solutions are then identified if minor changes are observed in the resulting noisy Pareto front. The sensibility analysis encompasses the following steps.

1. Capture the final BN model and sample a set of $N_{smp}^{end} = 1000 * Pop$ solutions called Pop_s where Pop is the population size;
2. Apply a perturbation/noise on each solution vector $\mathbf{x}$ of Pop_s, and compute a noisy solution set Pop_{sn}. Because this paper addresses a combinatorial problem, the noise is represented by a double bit-flip from 0 to 1 on the decision variable vectors of Pop_s. The probability for a solution to be mutated using a double bit-flip is defined as $p_{sn} = 0.15$;

Fig. 6 Example (one execution) of approximated Pareto fronts PF_s in blue and PF_{sn} in red for 2 objectives, $N = 20$ and $K = 8$. Overlapping points are shown in black (Color figure online)



3. Non-dominated solutions are then obtained from sets Pop_s and Pop_{sn}.
4. Compute the pmf function given by Eq. 6 for each non-dominated solution. This solution is represented by a circle proportional to its marginal probability; $P(Z_1 = z_1, \ldots, Z_M = z_M)$ (Martins et al. 2017);
5. For robustness evaluation purposes, select solutions with high probability (e.g. $p_{limit} > 0.5$).
   The selected solutions are used to build two approximated Pareto fronts PF_{sn} and PF_s with and without noise, respectively. An example of such Pareto fronts is shown in Fig. 6.

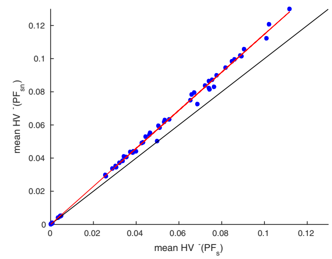
The Pareto fronts provided by the two populations Pop_s and Pop_{sn} are compared each other by their respective HV$^-$ indicators and IGD metrics. It aims at investigating how sensitive are the most probable solutions provided by each PGM learning technique. The results are shown in Fig. 7 and 8 as linear regressions in red with the corresponding β coefficient, and the quadrant bisector in black. Note that HMOBEDA$_{k2}$ and HMOBEDA$_{hc-k2}$ have coefficients (in bold) close to the bisector.

The data used in the scatter plots are average values over 30 executions of each algorithm in each instance given by Tables 2 and 3 of the Appendix. The Mann-Whitney-Wilcoxon test with $\alpha = 5\%$ is applied for the statistical analysis of the results. There are no statistically significant differences between PF_s and PF_{sn} for HMOBEDA$_{k2}$ and HMOBEDA$_{hc-k2}$ in almost all instances.

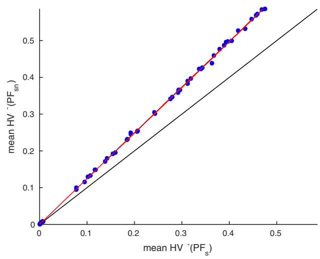
Based on the results of Figs. 7 and 8 supported by the statistical analysis of Tables 2 and 3, HMOBEDA$_{k2}$ and HMOBEDA$_{hc-k2}$ are less sensitive to noise. Therefore, the structure learning algorithm implemented by K2 and HC-K2 can provide a robust PGM model for HMOBEDA. However, HMOBEDA$_{hc-k2}$ is time consuming as shown in Fig. 4.

4.3 Learning performance: PGM variability analysis

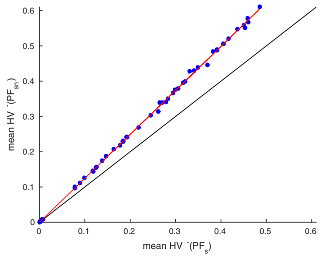
In this section, the learning capability of each algorithm is evaluated based on the distance between the PGM and a target model using the Structural Hamming Distance (SHD) (Tsamardinos et al. 2006).



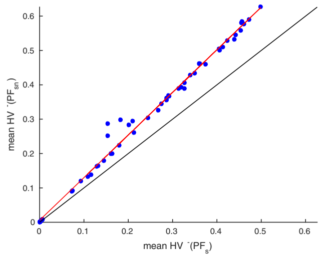
(a) HMOBEDA$_{k2}$



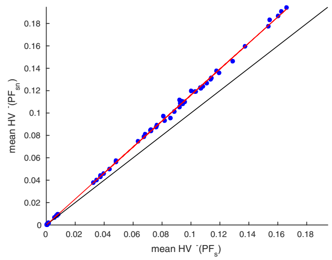
(b) HMOBEDA$_{pc}$



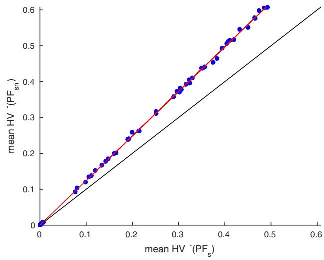
(c) HMOBEDA$_{mmhc}$



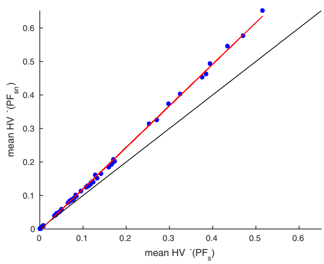
(d) HMOBEDA$_{iamb}$



(e) HMOBEDA$_{hc-k2}$

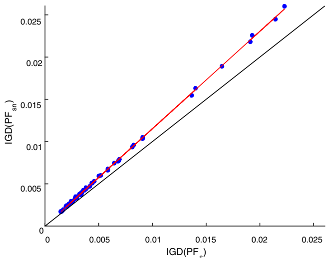


(f) HMOBEDA$_{sparse}$

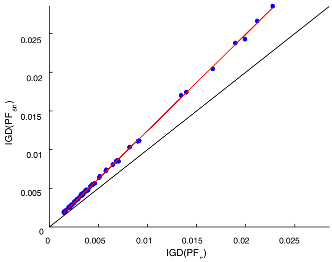


(g) HMOBEDA$_{tabu}$

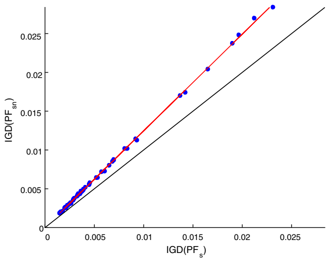
Fig. 7 Linear regression of average HV^{-} indicators (PF$_s$ versus PF$_{sn}$) in red for each HMOBEDAvariant: **a** HMOBEDA$_{k2}$ with **fi = 1.1478**, **b** HMOBEDA$_{pc}$ with $\beta = 1.2426$, **c** HMOBEDA$_{mmhc}$ with $\beta = 1.2459$, **d** HMOBEDA$_{iamb}$ with $\beta = 1.2404$, **e** HMOBEDA$_{hc-k2}$ with **fi = 1.1613**, **f** HMOBEDA$_{sparse}$ with $\beta = 1.2420$, **g** HMOBEDA$_{tabu}$ with $\beta = 1.2410$ (Color figure online)



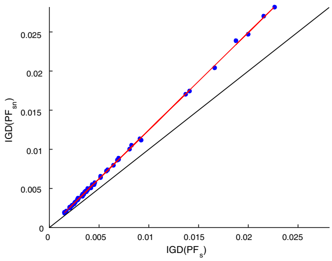
(a) HMOBEDA$_{k2}$



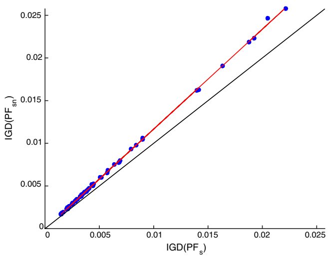
(b) HMOBEDA$_{pc}$



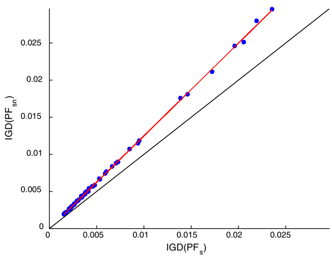
(c) HMOBEDA$_{mmhc}$



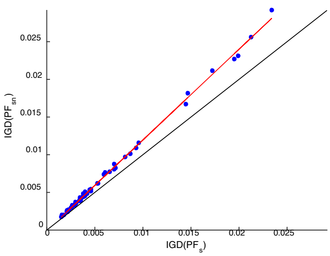
(d) HMOBEDA$_{iamb}$



(e) HMOBEDA$_{hc-k2}$



(f) HMOBEDA$_{sparse}$



(g) HMOBEDA$_{tabu}$

Fig. 8 Linear regression of average IGD metrics (PF$_s$ versus PF$_{sn}$) in red for each HMOBEDAvariant: **a** HMOBEDA$_{k2}$ with **fi = 1.1523**, **b** HMOBEDA$_{pc}$ with $\beta = 1.2452$, **c** HMOBEDA$_{mmhc}$ with $\beta = 1.2462$, **d** HMOBEDA$_{iamb}$ with $\beta = 1.2457$, **e** HMOBEDA$_{hc-k2}$ with **fi = 1.1706**, **f** HMOBEDA$_{sparse}$ with $\beta = 1.2492$, **g** HMOBEDA$_{tabu}$ with $\beta = 1.2043$ (Color figure online)

The SHD metric has been chosen due to its popularity to evaluate the learning of Bayesian network structure (Viinikka et al. 2018). This metric requires a reference structure model called target model which is different of the typical reference model used to analyze the accuracy of EDAs (usually the structure of the function or the structure of the problem when it is available). In our case, the challenge is the absence of a previously known target model. Similarly to (Viinikka et al. 2018), we thus used a target model represented by the Bayesian network learned for the best Pareto front (with the best hypervolume HV^{-}) found among all algorithms and executions. Using this target model we expect to compare how far the BN models computed by each algorithm are from the one capable of achieving the best results. For each instance, the target model is chosen among the BN models of the best Pareto fronts found by all algorithms and executions, i.e., the target model is defined as the 'best one' among 210 BNs (30 executions of each seven algorithms) with the best HV^{-}. This means that the BN corresponding to the best Pareto front (regarding hypervolume) among these 210 ones is selected as the target model for a given instance.

Therefore, we perform in this section a variability analysis among PGM structures. Furthermore, we evaluate each PGM learning technique based on its ranking, i.e., the number of times it achieves the minimum distance from the target model then the second minimum distance, and so on. In other words, we compare individual BN models directly from each execution and compute the quality of each learning technique based on the distance of its resulting BN from a target model.

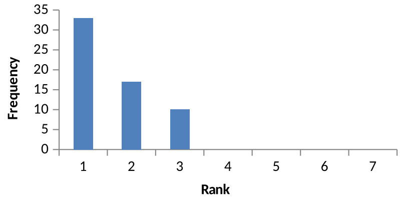
The SHD metric compares the structure of learned and target networks (Tsamardinos et al. 2006). We define the SHD as the number of operations required to match the learned and target networks using add, remove and reverse (edges) operations. A ranking of HMOBEDA variants is then computed for each problem instance based on the number of times an algorithm presents the best SHD values over 30 executions. Figure 9 shows the histogram of the rank distributions for each HMOBEDA variant considering all problem instances.

According to Fig. 9a, b, we notice that $HMOBEDA_{k2}$ and $HMOBEDA_{hc-k2}$ are ranked in the first four positions with better results for $HMOBEDA_{k2}$ (greater number of times being in the first rank). Similarly to the previous sections, $HMOBEDA_{k2}$ appears as the best option as it provides BN structures close to the PGM target model more often. By approaching the target model more often, $HMOBEDA_{k2}$ is expected to better represent the underlying relationship among objectives, variables, and parameters that provides better optimization results with even less computational effort.

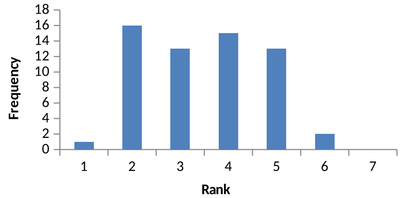
Based on the experiments, we conclude that $HMOBEDA_{k2}$ outperformed the other variants because, in spite of using less computational time (see Fig. 4), it provides comparable optimization results, less sensitive solutions to noise, and BN structures close to the target model more often.

5 Conclusion

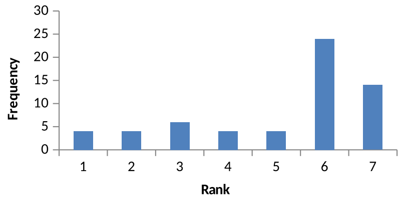
In this paper we have explored different BN learning techniques for a hybrid EDA which is based on a joint Bayesian network model of variables, local search parameters and objectives applied to multi- and many-objective combinatorial optimization. We



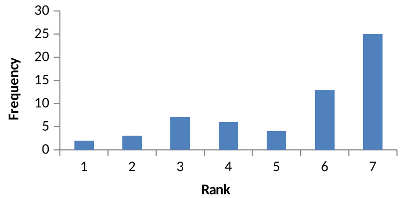
(a) HMOBEDA$_{k2}$



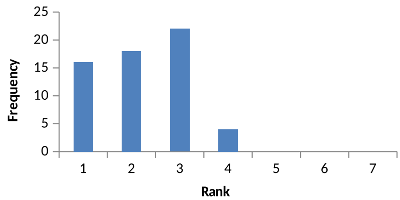
(b) HMOBEDA$_{pc}$



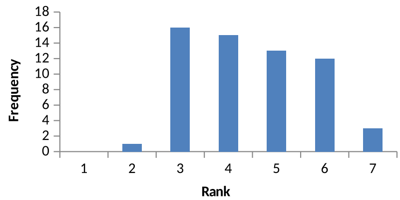
(c) HMOBEDA$_{mmhc}$



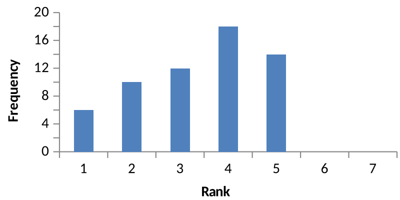
(d) HMOBEDA$_{iamb}$



(e) HMOBEDA$_{hc-k2}$



(f) HMOBEDA$_{sparse}$



(g) HMOBEDA$_{tabu}$

Fig. 9 Histogram for the HMOBEDA variants: number of times the corresponding algorithm presents the ranks 1, 2, 3, 4, 5, 6 and 7 along the 30 executions considering all instances: **a** HMOBEDA$_{k2}$, **b** HMOBEDA$_{pc}$, **c** HMOBEDA$_{mmhc}$, **d** HMOBEDA$_{iamb}$, **e** HMOBEDA$_{hc-k2}$, **f** HMOBEDA$_{sparse}$, and **g** HMOBEDA$_{tabu}$

have modified the original HMOBEDA to build other versions with different score-, constraint- and hybrid-based learning techniques: HMOBEDA$_{k2}$, HMOBEDA$_{hc-k2}$, HMOBEDA$_{iamb}$, HMOBEDA$_{pc}$ and HMOBEDA$_{mmhc}$.

In the experiments, we have analyzed the performance of each of those HMOBEDA variants on instances of the MNK-landscape problem with 2, 3, 5 and 8 objectives considering the hypervolume (HV$^-$) indicator and the Inverted Generational Distance (IGD) metric.

For the instances considered in this work, we observed that, under a multicriteria optimization analysis, HMOBEDA$_{k2}$ has outperformed the others in one performance criterion (run times) without decreasing the other two (convergence and coverage of its resulting Pareto fronts). Besides, a direct comparison based on the Structural Hamming Distance (SHD), between BN models generated at the end of the evolutionary process and a target BN, revealed better models (close to the target one) for HMOBEDA$_{k2}$. We believe that, this result emphasizes the better learning capability of HMOBEDA$_{k2}$ when compared with the other variants.

In addition, our analysis of robust solutions coded into the BN models has concluded that HMOBEDA$_{k2}$ and HMOBEDA$_{hc-k2}$ produce solutions that are less sensitive to noise compared to the other HMOBEDA variants. However, as HMOBEDA$_{k2}$ requires less computing time it can be considered the best option to provide BN models to HMOBEDA.

An interesting future research direction is using other types of PGM as hBOA. Moreover, in addition to the operators addressed in this work, other EDA parameters could be considered, as well as the effectiveness of learning the problem structure for an efficient search. Another direction to be considered in the future is the expansion of the analysis for different MNK parameters and configurations. Fitness landscape analysis could also be useful to acquire knowledge about the behavior and performance of structure learning algorithms for different types of instances. Finally, the work can be extended to other problems such as the multi-objective knapsack and multi-objective clustering problems.

Acknowledgements M. Delgado acknowledges CNPq, grants 309935/2017-2 e 439226/2018-0. R. Santana acknowledges support by the TIN2016-78365-R (Spanish Ministry of Economy, Industry and Competitiveness), PID2019-104966GB-I00 (Spanish Ministry of Science and Innovation), the IT-1244-19 (Basque Government) program and project 3KIA (KK-2020/00049) funded by the SPRI-Basque Government through the ELKARTEK program.

Appendix

Tables 2 and 3 present the respective HV$^-$ indicator and IGD metric to the approximated Pareto fronts provided by the two populations PF_s and PF_{sn}. The values are averaged over the results of 30 executions of each algorithm. The Mann-Whitney-Wilcoxon test with $\alpha = 5\%$ is applied for the statistical analysis of the results. Values of PF_s and PF_{sn} for each algorithm and instance with background in light blue have no statistically significant differences. The values in bold correspond to the best values for the paiwise comparison between PF_s and PF_{sn} for each HMOBEDA variant.

Table 2 Average HV^{-} over 30 independent executions for the approximated PF_s and PF_{sn} for each HMOBEDA variant on each problem instance

	HMOBEDA$_{k2}$		HMOBEDA$_{pc}$		HMOBEDA$_{mmhc}$		HMOBEDA$_{iamb}$		HMOBEDA$_{hc-k2}$		HMOBEDA$_{sparse}$		HMOBEDA$_{tabu}$	
Instance	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}
M2N20K2	0.10211	0.12067	0.43393	0.53156	0.45325	0.55021	0.46167	0.57690	0.10337	0.11935	0.44993	0.55181	0.10695	0.12406
M2N20K4	0.08878	0.10204	0.46924	0.58356	0.48508	0.60992	0.49875	0.62620	0.08591	0.09562	0.48588	0.60513	0.51557	0.65109
M2N20K6	0.07395	0.08650	0.47509	0.58529	0.45958	0.56793	0.45652	0.58527	0.07224	0.08491	0.49292	0.60804	0.07491	0.08773
M2N20K8	0.07545	0.08712	0.45908	0.57206	0.45951	0.57773	0.45464	0.58101	0.07613	0.08766	0.47484	0.59801	0.07882	0.09055
M2N20K10	0.07222	0.08387	0.44732	0.55816	0.43614	0.54834	0.45382	0.55760	0.07174	0.08451	0.46409	0.57758	0.47081	0.57753
M2N50K2	0.05334	0.06304	0.39306	0.49572	0.40609	0.50649	0.42288	0.52844	0.08173	0.09321	0.40636	0.51213	0.08455	0.09670
M2N50K4	0.05544	0.06327	0.45702	0.56873	0.45109	0.55749	0.47275	0.59027	0.09263	0.10882	0.46626	0.57611	0.09590	0.11281
M2N50K6	0.03873	0.04393	0.41832	0.52648	0.41734	0.51988	0.44192	0.54622	0.09453	0.10828	0.43241	0.54614	0.43357	0.54598
M2N50K8	0.05101	0.05853	0.39714	0.49684	0.38274	0.48368	0.41356	0.51061	0.10660	0.12285	0.41135	0.51513	0.11061	0.12757
M2N50K10	0.03982	0.04429	0.29341	0.36589	0.30526	0.37947	0.32613	0.38993	0.04381	0.05018	0.31658	0.39300	0.04540	0.05195
M2N100K2	0.04623	0.05404	0.37929	0.47679	0.39133	0.48729	0.40484	0.50383	0.09208	0.11157	0.39406	0.49346	0.39390	0.49366
M2N100K4	0.03532	0.04067	0.40592	0.49967	0.40485	0.50665	0.43907	0.53272	0.11787	0.13780	0.41996	0.51661	0.12180	0.14234
M2N100K6	0.06709	0.07949	0.38949	0.48641	0.39138	0.48983	0.40573	0.50052	0.13725	0.15964	0.40370	0.50569	0.14240	0.16522
M2N100K8	0.04438	0.05314	0.27536	0.34097	0.27901	0.34083	0.28625	0.36108	0.04825	0.05620	0.29700	0.37310	0.29759	0.37333
M2N100K10	0.03833	0.04352	0.24209	0.30485	0.24562	0.30368	0.26732	0.32696	0.03473	0.03985	0.25116	0.31596	0.03608	0.04129
M3N20K2	0.11177	0.12980	0.34041	0.42262	0.34013	0.43009	0.36040	0.46112	0.11358	0.12998	0.35226	0.43804	0.11751	0.13424
M3N20K4	0.08607	0.09958	0.36321	0.43885	0.34981	0.43864	0.36206	0.46107	0.09199	0.10527	0.37525	0.45424	0.37601	0.45345
M3N20K6	0.09043	0.10558	0.36732	0.45996	0.37077	0.44699	0.37416	0.46046	0.08852	0.10130	0.38314	0.46450	0.38447	0.46330
M3N20K8	0.07747	0.08975	0.34332	0.42641	0.33037	0.42809	0.35026	0.43361	0.08074	0.09698	0.35701	0.44128	0.08340	0.10056
M3N20K10	0.06525	0.07495	0.31839	0.39675	0.32089	0.39954	0.32681	0.40670	0.06816	0.08127	0.32990	0.41032	0.07045	0.08419
M3N50K2	0.06586	0.07819	0.31206	0.38971	0.29386	0.36658	0.31354	0.38927	0.10015	0.11949	0.32288	0.40446	0.32452	0.40238
M3N50K4	0.06859	0.07279	0.33667	0.42355	0.31682	0.39503	0.34049	0.42825	0.11416	0.13159	0.34890	0.43767	0.11797	0.13646
M3N50K6	0.05305	0.06179	0.31262	0.38360	0.29905	0.37492	0.31953	0.39283	0.11106	0.12653	0.32464	0.39693	0.11478	0.13090
M3N50K8	0.08185	0.09449	0.29371	0.36602	0.28316	0.34980	0.29245	0.36723	0.11949	0.13570	0.30325	0.38137	0.12349	0.14029
M3N50K10	0.03438	0.04111	0.20610	0.25350	0.19376	0.24144	0.21279	0.26122	0.03762	0.04440	0.21340	0.26190	0.03898	0.04591
M3N100K2	0.04263	0.04896	0.27962	0.34604	0.26203	0.31485	0.27445	0.34493	0.09372	0.11029	0.28973	0.35809	0.27085	0.32606
M3N100K4	0.03734	0.04386	0.29565	0.36510	0.27107	0.33916	0.29099	0.36891	0.11427	0.13146	0.30558	0.37885	0.11804	0.13637
M3N100K6	0.04319	0.04935	0.29234	0.35787	0.26453	0.33897	0.28592	0.35541	0.12866	0.14636	0.30206	0.37111	0.13307	0.15188
M3N100K8	0.08922	0.10160	0.24292	0.30189	0.21898	0.26965	0.24403	0.30303	0.10290	0.11909	0.25149	0.31207	0.25308	0.31355
M3N100K10	0.03041	0.03535	0.15952	0.19461	0.13874	0.17424	0.16135	0.19907	0.03241	0.03769	0.16507	0.20198	0.03369	0.03917
M5N20K2	0.10083	0.11220	0.19244	0.24631	0.18419	0.23057	0.18206	0.29877	0.16251	0.19064	0.19065	0.23837	0.16832	0.19772
M5N20K4	0.08477	0.09835	0.20709	0.25400	0.19149	0.24176	0.20926	0.29517	0.16622	0.19453	0.21535	0.26296	0.17260	0.20166
M5N20K6	0.07654	0.08311	0.19233	0.24924	0.17779	0.21767	0.17910	0.22333	0.16013	0.18649	0.19973	0.25803	0.16550	0.19281
M5N20K8	0.07414	0.08237	0.18591	0.23242	0.18345	0.22865	0.16360	0.20013	0.15434	0.18339	0.19246	0.24042	0.16994	0.20692
M5N20K10	0.07438	0.08165	0.18343	0.23070	0.18539	0.22831	0.20070	0.28329	0.15330	0.17779	0.19004	0.23989	0.15926	0.18450
M5N50K2	0.04973	0.05030	0.14220	0.17922	0.16341	0.20650	0.14496	0.17849	0.10676	0.12186	0.14712	0.18529	0.11064	0.12669
M5N50K4	0.04650	0.05522	0.15441	0.19133	0.14703	0.18722	0.15378	0.25151	0.10841	0.12396	0.16037	0.19853	0.11211	0.12842
M5N50K6	0.05049	0.05956	0.13814	0.17098	0.12394	0.15508	0.15400	0.28740	0.09568	0.11016	0.14279	0.17688	0.12873	0.16054
M5N50K8	0.03369	0.03840	0.10765	0.13257	0.12565	0.15635	0.11624	0.13921	0.06747	0.07872	0.11148	0.13762	0.07011	0.08159
M5N50K10	0.03223	0.03724	0.10243	0.12860	0.11813	0.14429	0.10837	0.13311	0.06364	0.07458	0.10602	0.13361	0.06589	0.07733
M5N100K2	0.03354	0.03845	0.11600	0.14780	0.11738	0.14482	0.13175	0.16354	0.07628	0.08928	0.12047	0.15276	0.07892	0.09255
M5N100K4	0.03054	0.03447	0.11786	0.14880	0.09966	0.12494	0.12947	0.16213	0.07260	0.08375	0.13432	0.16753	0.07515	0.08676
M5N100K6	0.02888	0.03371	0.09522	0.11499	0.08898	0.11199	0.09325	0.11915	0.04838	0.05759	0.09881	0.11916	0.05000	0.05964
M5N100K8	0.02547	0.02997	0.07785	0.09947	0.07847	0.09601	0.07447	0.09218	0.03989	0.04617	0.08054	0.10336	0.04124	0.04792
M5N100K10	0.02575	0.02909	0.07787	0.09474	0.07854	0.10107	0.07308	0.08985	0.03757	0.04320	0.07578	0.09343	0.03881	0.04471
M8N20K2	0.00335	0.00393	0.00516	0.00643	0.00517	0.00633	0.00575	0.00730	0.00564	0.00657	0.00534	0.00666	0.00584	0.00681
M8N20K4	0.00420	0.00472	0.00667	0.00830	0.00694	0.00861	0.00561	0.00697	0.00726	0.00861	0.00690	0.00861	0.00754	0.00889
M8N20K6	0.00464	0.00515	0.00687	0.00849	0.00707	0.00867	0.00676	0.00845	0.00785	0.00925	0.00711	0.00877	0.00813	0.00957
M8N20K8	0.00489	0.00505	0.00675	0.00857	0.00600	0.00749	0.00535	0.00681	0.00826	0.00965	0.00623	0.00776	0.00857	0.01000
M8N20K10	0.00426	0.00484	0.00577	0.00713	0.00522	0.00643	0.00629	0.00778	0.00715	0.00811	0.00597	0.00741	0.00741	0.00839
M8N50K2	0.00062	0.00064	0.00140	0.00175	0.00128	0.00159	0.00160	0.00197	0.00133	0.00155	0.00145	0.00181	0.00137	0.00161
M8N50K4	0.00071	0.00081	0.00173	0.00220	0.00168	0.00207	0.00180	0.00220	0.00153	0.00176	0.00179	0.00227	0.00186	0.00228
M8N50K6	0.00075	0.00085	0.00173	0.00210	0.00167	0.00212	0.00165	0.00207	0.00148	0.00164	0.00179	0.00218	0.00153	0.00170
M8N50K8	0.00068	0.00081	0.00167	0.00204	0.00167	0.00210	0.00118	0.00149	0.00124	0.00144	0.00173	0.00211	0.00129	0.00149
M8N50K10	0.00049	0.00057	0.00101	0.00125	0.00069	0.00086	0.00106	0.00134	0.00087	0.00101	0.00110	0.00138	0.00110	0.00139
M8N100K2	0.00016	0.00018	0.00050	0.00063	0.00050	0.00061	0.00039	0.00050	0.00045	0.00051	0.00051	0.00064	0.00047	0.00053
M8N100K4	0.00026	0.00031	0.00081	0.00103	0.00090	0.00110	0.00058	0.00073	0.00063	0.00073	0.00084	0.00107	0.00066	0.00075
M8N100K6	0.00022	0.00025	0.00067	0.00083	0.00059	0.00074	0.00069	0.00085	0.00047	0.00053	0.00070	0.00086	0.00049	0.00055
M8N100K8	0.00023	0.00026	0.00064	0.00081	0.00069	0.00086	0.00059	0.00073	0.00045	0.00051	0.00067	0.00085	0.00047	0.00053
M8N100K10	0.00021	0.00024	0.00054	0.00067	0.00059	0.00074	0.00060	0.00073	0.00040	0.00045	0.00061	0.00077	0.00061	0.00077

Table 3 Average IGD over 30 independent executions for the approximated PF_s and PF_{sn} for each HMOBEDA variant on each problem instance.

	HMOBEDA$_{k2}$		HMOBEDA$_{pc}$		HMOBEDA$_{mmhc}$		HMOBEDA$_{iamb}$		HMOBEDA$_{hc-k2}$		HMOBEDA$_{sparse}$		HMOBEDA$_{tabu}$	
Instance	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}	PF_s	PF_{sn}
M2N20K2	0.01932	0.02256	0.01990	0.02425	0.01964	0.02483	0.02002	0.02473	0.01928	0.02235	0.02058	0.02510	0.01994	0.02310
M2N20K4	0.02229	0.02605	0.02273	0.02850	0.02311	0.02837	0.02264	0.02813	0.02219	0.02575	0.02357	0.02959	0.02341	0.02915
M2N20K6	0.01912	0.02184	0.01894	0.02375	0.01902	0.02377	0.01878	0.02390	0.01880	0.02186	0.01962	0.02459	0.01950	0.02266
M2N20K8	0.02145	0.02449	0.02115	0.02663	0.02119	0.02696	0.02153	0.02702	0.02047	0.02467	0.02191	0.02795	0.02126	0.02559
M2N20K10	0.01645	0.01889	0.01662	0.02040	0.01652	0.02040	0.01662	0.02038	0.01636	0.01903	0.01724	0.02113	0.01725	0.02112
M2N50K2	0.00437	0.00507	0.00440	0.00550	0.00448	0.00558	0.00449	0.00546	0.00433	0.00515	0.00457	0.00570	0.00449	0.00536
M2N50K4	0.00690	0.00779	0.00707	0.00845	0.00696	0.00871	0.00694	0.00872	0.00681	0.00776	0.00723	0.00901	0.00704	0.00805
M2N50K6	0.00585	0.00674	0.00584	0.00736	0.00602	0.00727	0.00587	0.00741	0.00580	0.00684	0.00606	0.00764	0.00605	0.00762
M2N50K8	0.01368	0.01549	0.01341	0.01700	0.01369	0.01702	0.01370	0.01703	0.01395	0.01617	0.01390	0.01760	0.01445	0.01672
M2N50K10	0.00909	0.01035	0.00919	0.01117	0.00918	0.01141	0.00910	0.01138	0.00901	0.01046	0.00953	0.01183	0.00932	0.01086
M2N100K2	0.00336	0.00385	0.00335	0.00416	0.00341	0.00429	0.00340	0.00427	0.00338	0.00389	0.00346	0.00431	0.00347	0.00431
M2N100K4	0.00460	0.00532	0.00463	0.00567	0.00452	0.00577	0.00458	0.00573	0.00448	0.00527	0.00478	0.00588	0.00463	0.00546
M2N100K6	0.00811	0.00935	0.00815	0.01033	0.00807	0.01021	0.00807	0.01000	0.00791	0.00936	0.00846	0.01071	0.00819	0.00967
M2N100K8	0.01400	0.01628	0.01396	0.01744	0.01419	0.01747	0.01410	0.01747	0.01415	0.01631	0.01460	0.01809	0.01465	0.01814
M2N100K10	0.00826	0.00961	0.00819	0.01029	0.00834	0.01022	0.00829	0.01048	0.00840	0.00981	0.00846	0.01068	0.00872	0.01017
M3N20K2	0.00695	0.00788	0.00701	0.00867	0.00698	0.00876	0.00694	0.00885	0.00689	0.00796	0.00727	0.00899	0.00715	0.00823
M3N20K4	0.00679	0.00769	0.00679	0.00849	0.00681	0.00855	0.00685	0.00866	0.00685	0.00780	0.00703	0.00879	0.00704	0.00879
M3N20K6	0.00586	0.00657	0.00576	0.00720	0.00571	0.00717	0.00571	0.00718	0.00572	0.00653	0.00592	0.00743	0.00592	0.00743
M3N20K8	0.00644	0.00748	0.00645	0.00810	0.00651	0.00805	0.00646	0.00792	0.00635	0.00748	0.00666	0.00839	0.00657	0.00776
M3N20K10	0.00518	0.00603	0.00513	0.00653	0.00530	0.00646	0.00516	0.00656	0.00517	0.00600	0.00530	0.00676	0.00534	0.00620
M3N50K2	0.00286	0.00346	0.00290	0.00366	0.00294	0.00369	0.00288	0.00360	0.00288	0.00335	0.00300	0.00380	0.00299	0.00372
M3N50K4	0.00277	0.00323	0.00278	0.00353	0.00284	0.00351	0.00279	0.00351	0.00278	0.00324	0.00287	0.00365	0.00289	0.00335
M3N50K6	0.00366	0.00421	0.00367	0.00457	0.00376	0.00467	0.00370	0.00458	0.00364	0.00425	0.00379	0.00473	0.00377	0.00441
M3N50K8	0.00504	0.00590	0.00507	0.00637	0.00521	0.00646	0.00516	0.00641	0.00509	0.00600	0.00534	0.00664	0.00527	0.00621
M3N50K10	0.00295	0.00332	0.00293	0.00362	0.00290	0.00371	0.00287	0.00375	0.00292	0.00341	0.00303	0.00376	0.00302	0.00353
M3N100K2	0.00252	0.00289	0.00251	0.00320	0.00255	0.00316	0.00258	0.00315	0.00253	0.00297	0.00260	0.00332	0.00264	0.00326
M3N100K4	0.00204	0.00239	0.00207	0.00257	0.00207	0.00261	0.00211	0.00257	0.00206	0.00239	0.00215	0.00265	0.00213	0.00247
M3N100K6	0.00224	0.00262	0.00229	0.00285	0.00226	0.00287	0.00230	0.00288	0.00224	0.00263	0.00236	0.00296	0.00232	0.00272
M3N100K8	0.00909	0.01051	0.00903	0.01110	0.00930	0.01130	0.00926	0.01120	0.00896	0.01063	0.00936	0.01149	0.00958	0.01160
M3N100K10	0.00260	0.00293	0.00259	0.00326	0.00258	0.00320	0.00262	0.00327	0.00259	0.00300	0.00268	0.00338	0.00268	0.00310
M5N20K2	0.00420	0.00470	0.00417	0.00520	0.00406	0.00522	0.00420	0.00510	0.00404	0.00469	0.00422	0.00542	0.00417	0.00486
M5N20K4	0.00384	0.00437	0.00391	0.00476	0.00380	0.00477	0.00381	0.00471	0.00383	0.00431	0.00405	0.00495	0.00396	0.00446
M5N20K6	0.00387	0.00445	0.00377	0.00480	0.00386	0.00484	0.00387	0.00479	0.00385	0.00442	0.00392	0.00497	0.00398	0.00457
M5N20K8	0.00380	0.00454	0.00398	0.00484	0.00390	0.00493	0.00388	0.00495	0.00395	0.00455	0.00413	0.00501	0.00402	0.00512
M5N20K10	0.00440	0.00506	0.00439	0.00548	0.00447	0.00553	0.00433	0.00549	0.00443	0.00499	0.00453	0.00566	0.00459	0.00516
M5N50K2	0.00347	0.00398	0.00346	0.00435	0.00353	0.00438	0.00353	0.00446	0.00349	0.00407	0.00358	0.00451	0.00362	0.00422
M5N50K4	0.00299	0.00349	0.00300	0.00375	0.00298	0.00377	0.00298	0.00372	0.00297	0.00343	0.00310	0.00388	0.00306	0.00355
M5N50K6	0.00357	0.00422	0.00361	0.00450	0.00365	0.00467	0.00360	0.00458	0.00364	0.00427	0.00375	0.00465	0.00378	0.00484
M5N50K8	0.00241	0.00280	0.00246	0.00308	0.00246	0.00304	0.00248	0.00304	0.00244	0.00286	0.00255	0.00319	0.00253	0.00296
M5N50K10	0.00246	0.00291	0.00255	0.00314	0.00262	0.00313	0.00255	0.00320	0.00252	0.00292	0.00264	0.00325	0.00261	0.00304
M5N100K2	0.00227	0.00263	0.00226	0.00280	0.00232	0.00289	0.00227	0.00284	0.00225	0.00262	0.00235	0.00289	0.00233	0.00272
M5N100K4	0.00277	0.00322	0.00277	0.00344	0.00282	0.00356	0.00280	0.00358	0.00279	0.00319	0.00290	0.00371	0.00289	0.00331
M5N100K6	0.00337	0.00395	0.00342	0.00421	0.00336	0.00434	0.00349	0.00428	0.00338	0.00393	0.00354	0.00437	0.00349	0.00407
M5N100K8	0.00220	0.00258	0.00226	0.00279	0.00224	0.00278	0.00223	0.00276	0.00219	0.00256	0.00234	0.00289	0.00228	0.00266
M5N100K10	0.00198	0.00231	0.00201	0.00253	0.00201	0.00254	0.00203	0.00253	0.00202	0.00233	0.00210	0.00262	0.00209	0.00241
M8N20K2	0.00369	0.00427	0.00363	0.00468	0.00368	0.00458	0.00369	0.00463	0.00358	0.00421	0.00375	0.00485	0.00372	0.00438
M8N20K4	0.00337	0.00380	0.00332	0.00408	0.00333	0.00420	0.00331	0.00411	0.00334	0.00381	0.00345	0.00423	0.00346	0.00395
M8N20K6	0.00333	0.00373	0.00332	0.00410	0.00331	0.00408	0.00325	0.00410	0.00326	0.00375	0.00344	0.00425	0.00338	0.00389
M8N20K8	0.00319	0.00375	0.00319	0.00403	0.00326	0.00406	0.00332	0.00404	0.00326	0.00376	0.00338	0.00421	0.00337	0.00390
M8N20K10	0.00334	0.00385	0.00327	0.00419	0.00332	0.00419	0.00333	0.00421	0.00338	0.00380	0.00338	0.00435	0.00350	0.00393
M8N50K2	0.00208	0.00242	0.00214	0.00258	0.00208	0.00264	0.00209	0.00260	0.00204	0.00242	0.00221	0.00267	0.00212	0.00252
M8N50K4	0.00203	0.00241	0.00208	0.00255	0.00206	0.00258	0.00202	0.00255	0.00202	0.00237	0.00215	0.00265	0.00209	0.00264
M8N50K6	0.00206	0.00237	0.00206	0.00262	0.00206	0.00260	0.00209	0.00254	0.00210	0.00233	0.00213	0.00272	0.00218	0.00242
M8N50K8	0.00205	0.00235	0.00207	0.00256	0.00210	0.00255	0.00204	0.00260	0.00206	0.00239	0.00214	0.00266	0.00214	0.00247
M8N50K10	0.00198	0.00233	0.00197	0.00243	0.00200	0.00252	0.00203	0.00248	0.00197	0.00228	0.00210	0.00256	0.00211	0.00256
M8N100K2	0.00170	0.00193	0.00170	0.00212	0.00177	0.00213	0.00170	0.00214	0.00170	0.00194	0.00183	0.00220	0.00176	0.00201
M8N100K4	0.00159	0.00183	0.00158	0.00201	0.00161	0.00203	0.00164	0.00198	0.00158	0.00184	0.00164	0.00209	0.00164	0.00190
M8N100K6	0.00149	0.00169	0.00149	0.00185	0.00150	0.00185	0.00151	0.00188	0.00146	0.00171	0.00154	0.00192	0.00151	0.00177
M8N100K8	0.00151	0.00175	0.00154	0.00190	0.00151	0.00192	0.00152	0.00191	0.00153	0.00174	0.00160	0.00197	0.00158	0.00181
M8N100K10	0.00148	0.00174	0.00151	0.00193	0.00154	0.00191	0.00152	0.00192	0.00152	0.00173	0.00159	0.00198	0.00160	0.00198

References

Aguirre, H.E., Tanaka, K.: Insights on properties of multiobjective MNK-landscapes. In: Proceedings of the 2004 Congress on Evolutionary Computation, vol. 1, pp. 196–203. IEEE, Portland (2004)

Aguirre, H.E., Tanaka, K.: Working principles, behavior, and performance of MOEAs on MNK-landscapes. Eur. J. Oper. Res. **181**(3), 1670–1690 (2007)

Aragam, B., Gu, J., Zhou, Q.: Learning large-scale Bayesian networks with the sparsebn package. J. Stat. Softw. **91**(11), 1–38 (2019)

Bengoetxea, E., Larrañaga, P., Bielza, C., Del Pozo, J.F.: Optimal row and column ordering to improve table interpretation using estimation of distribution algorithms. J. Heurist. **17**(5), 567–588 (2011)

Bennett, K.P., Parrado-Hernández, E.: The interplay of optimization and machine learning research. J. Mach. Learn. Res. **7**, 1265–1281 (2006)

Brownlee, A.E., McCall, J.A., Pelikan, M.: Influence of selection on structure learning in Markov network EDAs: an empirical study. In: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, pp. 249–256 (2012)

Buntine, W.: Theory refinement on Bayesian networks. In: Uncertainty Proceedings, pp. 52–60. Elsevier (1991)

Colombo, D., Maathuis, M.H.: Order-independent constraint-based causal structure learning. J. Mach. Learn. Res. **15**(1), 3741–3782 (2014)

Cooper, G., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Mach. Learn. **9**(4), 309–347 (1992)

Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evolut. Comput. **6**, 182–197 (2002)

Ding, F., Zhuang, Y.: Distributed Bayesian network learning algorithm using storm topology. Int. J. Grid Distrib. Comput. **11**(4), 113–126 (2018)

de Mattos Neto, P.S., Marinho, M.H., Siqueira, H., de Souza Tadano, Y., Machado, V., AntoniniAlves, T., de Oliveira, J.F.L., Madeiro, F.: A methodology to increase the accuracy of particulate matter predictors based on time decomposition. Sustainability **12**(18), 7310 (2020)

Echegoyen, C., Lozano, J.A., Santana, R., Larrañaga, P.: Exact Bayesian network learning in estimation of distribution algorithms. In: 2007 IEEE Congress on Evolutionary Computation, pp. 1051–1058. IEEE (2007)

Echegoyen, C., Zhang, Q., Mendiburu, A., Santana, R., Lozano, J.A.: On the limits of effectiveness in estimation of distribution algorithms. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 1573–1580. IEEE (2011)

Heckerman, D., Geiger, D., Chickering, D.: Learning Bayesian networks: the combination of knowledge and statistical data. Mach. Learn. **20**(3), 197–243 (1995)

Henrion, M.: Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Machine Intelligence and Pattern Recognition, vol. 5, pp. 149–163. Elsevier (1988)

Karshenas, H., Santana, R., Bielza, C., Larrañaga, P.: Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables. IEEE Trans. Evol. Comput. **18**, 519–542 (2014)

Kauffman, S.A.: The Origins of Order: Self-organization and Selection in Evolution. Oxford University Press, New York (1993)

Khan, N., Goldberg, D.E., Pelikan, M.: Multi-objective Bayesian optimization algorithm. Technical report, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory-Tech Report no.2002009, Urbana, IL (2002)

Kollat, J.B., Reed, P., Kasprzyk, J.: A new epsilon-dominance hierarchical Bayesian optimization algorithm for large multiobjective monitoring network design problems. Adv. Water Resour. **31**(5), 828–845 (2008)

Larrañaga, P., Karshenas, H., Bielza, C., Santana, R.: A review on probabilistic graphical models in evolutionary computation. J. Heuristics **18**(5), 795–819 (2012)

Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, New York (2001)

Lauritzen, S.L.: Graphical Models. Oxford Clarendon Press, Oxford (1996)

Lima, C.F., Lobo, F.G., Pelikan, M., Goldberg, D.E.: Model accuracy in the Bayesian optimization algorithm. Soft. Comput. **15**(7), 1351–1371 (2011)

Marti, L., Garcia, J., Berlanga, A., Molina, J.M.: Model-building algorithms for multiobjective EDAs: directions for improvement. In: IEEE Conference on Evolutionary Computation. CEC'2008, pp. 2843–2850. IEEE, Piscataway, NJ (2008)

Martins, M.S., Delgado, M., Lüders, R., Santana, R., Gonçalves, R.A., de Almeida, C.P.: Exploring the probabilistic graphic model of a hybrid multi-objective Bayesian estimation of distribution algorithm. Appl. Soft Comput. (2018). https://doi.org/10.1016/j.asoc.2018.08.039

Martins, M.S., Delgado, M.R., Lüders, R., Santana, R., Ricardo, Gonçalves, Almeida, C.P.d.: Probabilistic analysis of Pareto Front approximation for a hybrid multi-objective bayesian estimation of distribution algorithm. In: Proceedings of the 2017 Brazilian Conference on Intelligent Systems, BRACIS'17, pp. 384–389 (2017)

Martins, M.S., Delgado, M.R., Santana, R., Lüders, R., Gonçalves, R.A., Almeida, C.P.D.: HMOBEDA: hybrid multi-objective bayesian estimation of distribution algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'16, pp. 357–364. ACM, New York, NY (2016)

Martins, M.S., El Yafrani, M., Santana, R., Delgado, M.R., Lüders, R., Ahiod, B.: On the performance of multi-objective estimation of distribution algorithms for combinatorial problems. In: IEEE Conference on Evolutionary Computation, CEC'18, pp. 1–8. arXiv:1806.09935 (2018)

Martins, M.S.R., Delgado, M.R.B.S., Lüders, R., Santana, R., Gonçalves, R.A., de Almeida, C.P.: Hybrid multi-objective Bayesian estimation of distribution algorithm: a comparative analysis for the multi-objective knapsack problem. J. Heuristics **8**, 1–23 (2017)

Meneghini, I.R., Guimaraes, F.G., Gaspar-Cunha, A.: Competitive coevolutionary algorithm for robust multi-objective optimization: the worst case minimization. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 586–593. IEEE (2016)

Moran, S., He, Y., Liu, K.: Choosing the best Bayesian classifier: an empirical study. IAENG Int. J. Comput. Sci. **36**(4), 322–331 (2009)

Mühlenbein, H., Paab, G.: From Recombination of Genes to the Estimation of Distributions I. Binary Parameters. Parallel Problem Solving from Nature. PPSN IV—Lecture Notes in Computer Science, vol. 1411, pp. 178–187. Springer, London (1996)

Pelikan, M., Goldberg, D.E., Tsutsui, S.: Hierarchical bayesian optimization algorithm: toward a new generation of evolutionary algorithms. In: SICE 2003 Annual Conference (IEEE Cat. No. 03TH8734), vol. 3, pp. 2738–2743. IEEE (2003)

Pelikan, M., Hauschild, M.W.: Learn from the past: improving model-directed optimization by transfer learning based on distance-based bias. Missouri Estimation of Distribution Algorithms Laboratory, University of Missouri in St. Louis, MO, United States. Technical Report 2012007 (2012)

Pelikan, M., Sastry, K., Goldberg, D.E.: iBOA: The incremental Bayesian optimization algorithm. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, pp. 455–462 (2008)

Puchta, E.D., Lucas, R., Ferreira, F.R., Siqueira, H.V., Kaster, M.S.: Gaussian adaptive PID control optimized via genetic algorithm applied to a step-down dc-dc converter. In: 2016 12th IEEE International Conference on Industry Applications (INDUSCON), pp. 1–6. IEEE (2016)

Puchta, E.D., Siqueira, H.V., Kaster, M.D.S.: Optimization tools based on metaheuristics for performance enhancement in a Gaussian adaptive PID controller. IEEE Trans. Cybern. **50**(3), 1185–1194 (2020)

Ribeiro, V.H.A., Reynoso-Meza, G., Siqueira, H.V.: Multi-objective ensembles of echo state networks and extreme learning machines for streamflow series forecasting. Eng. Appl. Artif. Intell. **95**, 103910 (2020)

Russell Stuart, J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River (2009)

Santana, R., Larrañaga, P., Lozano, J.A.: Combining variable neighborhood search and estimation of distribution algorithms in the protein side chain placement problem. J. Heuristics **14**, 519–547 (2008)

Santana, R., Mendiburu, A., Lozano, J.A.: Evolving MNK-landscapes with structural constraints. In: IEEE Congress on Evolutionary Computation. CEC'15, pp. 1364–1371. IEEE, Sendai (2015)

Santana, R., Mendiburu, A., Lozano, J.A.: Multi-objective NM-landscapes. In: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO'15, pp. 1477–1478. ACM, Orlando, FL (2015)

Santana Jr., C.J., Macedo, M., Siqueira, H., Gokhale, A., Bastos-Filho, C.J.: A novel binary artificial bee colony algorithm. Future Gen. Comput. Syst. **98**, 180–196 (2019)

Santhanam, N.P., Wainwright, M.J.: Information-theoretic limits of selecting binary graphical models in high dimensions. IEEE Trans. Inf. Theory **58**(7), 4117–4134 (2012)

Santos, P., Macedo, M., Figueiredo, E., Santana, C.J., Soares, F., Siqueira, H., Maciel, A., Gokhale, A., Bastos-Filho, C.J.: Application of pso-based clustering algorithms on educational databases. In: 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI), pp. 1–6. IEEE (2017)
Scanagatta, M., Salmerón, A., Stella, F.: A survey on Bayesian network structure learning from data. In: Progress in Artificial Intelligence, pp. 1–15 (2019)
Scutari, M.: Learning Bayesian networks with the bnlearn r package. arXiv:0908.3817 (2009)
Scutari, M., Graafland, C.E., Gutiérrez, J.M.: Who learns better Bayesian network structures: constraint-based, score-based or hybrid algorithms? In: International Conference on Probabilistic Graphical Models, pp. 416–427 (2018)
Shah, R., Reed, P.: Comparative analysis of multiobjective evolutionary algorithms for random and correlated instances of multiobjective d-dimensional knapsack problems. Eur. J. Oper. Res. **211**(3), 466–479 (2011)
Shim, V.A., Tan, K.C., Chia, J.Y., Al Mamun, A.: Multi-objective optimization with estimation of distribution algorithm in a noisy environment. Evol. Comput. **21**(1), 149–177 (2013)
Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evol. Comput. **2**, 221–248 (1994)
Tsagris, M.: Bayesian network learning with the pc algorithm: an improved and correct variation. Appl. Artif. Intell. **33**(2), 101–123 (2019)
Tsamardinos, I., Aliferis, C.F., Statnikov, A.R., Statnikov, E.: Algorithms for large scale Markov blanket discovery. In: FLAIRS Conference, vol. 2, pp. 376–380. AAAI Press, St. Augustine, FL (2003)
Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max–min hill-climbing Bayesian network structure learning algorithm. Mach. Learn. **65**(1), 31–78 (2006)
Viinikka, J., Eggeling, R., Koivisto, M., et al.: Intersection-validation: a method for evaluating structure learning without ground truth. Proc. Mach. Learn. Res. **84**, 1570–1578 (2018)
Yuan, C., Malone, B.: Learning optimal Bayesian networks: a shortest path perspective. J. Artif. Intell. Res. **48**(1), 23–65 (2013)
Zitzler, E., Thiele, L.: Multiple objective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. **3**, 257–271 (1999)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Marcella S. R. Martins[1] · Mohamed El Yafrani[2] · Myriam Delgado[1] · Ricardo Lüders[1] · Roberto Santana[3] · Hugo V. Siqueira[1] · Huseyin G. Akcay[4] · Belaïd Ahiod[5]

Mohamed El Yafrani
mey@mp.aau.dk

Myriam Delgado
myriamdelg@utfpr.edu.br

Ricardo Lüders
luders@utfpr.edu.br

Roberto Santana
roberto.santana@ehu.es

Hugo V. Siqueira
hugosiqueira@utfpr.edu.br

Huseyin G. Akcay
gokhanakcay@gmail.com

Belaïd Ahiod
ahiod@fsr.ac.ma

[1] Federal University of Technology - Paraná (UTFPR), Curitiba, Brazil

[2] Operations Research Group, Aalborg University (AAU), Aalborg, Denmark

[3] University of the Basque Country (UPV/EHU), San Sebastián, Spain

[4] Akdeniz University (AKU), Antalya, Turkey

[5] Mohammed V University in Rabat, Rabat, Morocco