



## Conference

## Exploring the probabilistic graphic model of a hybrid multi-objective Bayesian estimation of distribution algorithm

Marcella S.R. Martins<sup>a,\*</sup>, Myriam Delgado<sup>a</sup>, Ricardo Lüders<sup>a</sup>, Roberto Santana<sup>b</sup>, Richard A. Gonçalves<sup>c</sup>, Carolina P. de Almeida<sup>c</sup>

<sup>a</sup> Federal University of Technology - Paraná, Curitiba PR, Brazil

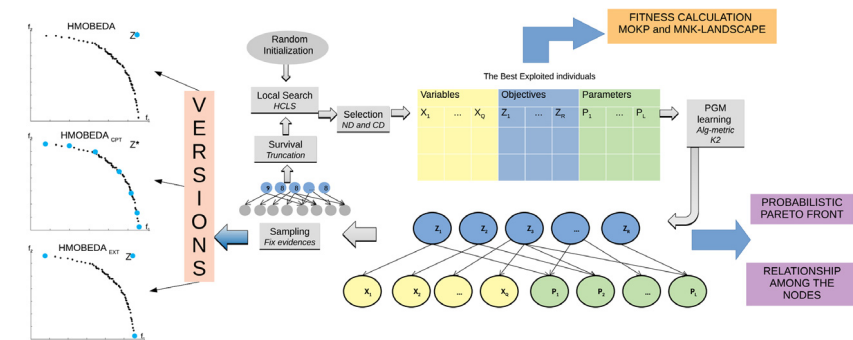
<sup>b</sup> University of the Basque Country, San Sebastián, Spain

<sup>c</sup> State University of the Midwest of Paraná, Guarapuava PR, Brazil

## HIGHLIGHTS

- An approach for multi and many-objective combinatorial optimization is explored.
- It is based on a joint probabilistic model with local optimizers as an online tuning.
- Versions with different sampling are analyzed from a probabilistic point of view.
- The best version outperforms other approaches when the number of objectives increases.
- Information can be extracted from the models to learn and explore dependencies.

## GRAPHICAL ABSTRACT



## ARTICLE INFO

## Article history:

Received 20 February 2018

Received in revised form 11 July 2018

Accepted 27 August 2018

Available online 1 September 2018

## Keywords:

Multi-objective optimization

Estimation of distribution algorithms

Automatic algorithm configuration

## ABSTRACT

The Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA) has shown to be very competitive for Many Objective Optimization Problems (MaOPs). The Probabilistic Graphic Model (PGM) of HMOBEDA expands the possibilities for exploration as it provides the joint probability of decision variables, objectives, and configuration parameters of an embedded local search. This work investigates different sampling mechanisms of HMOBEDA, applying the considered approaches to two different combinatorial MaOPs. Moreover, the paper provides a broad set of statistical analyses on its PGM model. These analyses have been carried out to evaluate how the interactions among variables, objectives and local search parameters are captured by the model and how information collected from different runs can be aggregated and explored in a Probabilistic Pareto Front. In experiments, two variants of HMOBEDA are compared with the original version, each one with a different set of evidences fixed during the sampling process. Results for instances of multi-objective knapsack problem with 2–5 and 8 objectives show that the best variant outperforms the original HMOBEDA in terms of convergence and diversity in the solution set. This best variant is then compared with five state-of-the-art evolutionary algorithms using the knapsack problem instances as well as a set of MNK-landscape instances with 2, 3, 5 and 8 objectives. HMOBEDA outperforms all of them.

## 1. Introduction

In many optimization problems, maximizing/minimizing two or more objective functions represents a challenge to a large number of optimizers [1]. This class of problems is known as Multi-objective Optimization Problems (MOPs), and solving MOPs has

\* Corresponding author.

E-mail address: [marcella@utfpr.edu.br](mailto:marcella@utfpr.edu.br) (M.S.R. Martins).

thus been established as an important field of research [2]. In the past few years, problems with more than three objectives are becoming usual. They are referred as Many Objective Optimization Problems (MaOPs) [3].

MOPs and especially MaOPs contain several, usually conflicting, objectives. This means, optimizing one objective does not necessarily optimize the others. Due to the objectives trade-off, a set called Pareto-optimal is generated at the decision variable space. Different approaches have been proposed to approximate the Pareto-optimal front (i.e. Pareto-optimal corresponding objectives) in the objective space in various scenarios [2].

Multi-objective evolutionary algorithms (MOEAs) are classical examples of these approaches. They have achieved good results on MOPs as they search multiple solutions in parallel with some advantages when compared with math programming-based approaches. Several MOEAs incorporating local search (LS) have been investigated on combinatorial optimization, and these hybrid approaches can often achieve good performance for many problems [4–6]. However, as discussed in [7] and [8], they still present challenges, such as the choice of suitable LS parameters e.g., the type, frequency and intensity of LS applied to a particular candidate solution.

Another strategy widely used in evolutionary optimization is the probabilistic modeling, which is the basis of Estimation of Distribution Algorithms (EDAs) [9]. EDAs are a class of Evolutionary Algorithms (EAs) that explores the search space by building a probabilistic model from a set with the current best candidate solutions [10]. Since new solutions are sampled from the probabilistic model, evolution is guided toward more promising areas of the search space.

Normally, Multi-objective Estimation of Distribution Algorithm (MOEDAs) [11] integrate both model building and sampling techniques into evolutionary multi-objective optimizers using special selection schemes [12]. Probabilistic Graphical Models (PGMs) combine graph and probability theory. They have been adopted to improve EDAs and MOEDAs performance [13]. Most of MOEDAs developed to deal with combinatorial MOPs adopt Bayesian Networks (BNs) as PGM. This way, these PGM-based MOEAs are able to properly modeling variables' dependencies, besides solving the optimization problem.

Furthermore, recently the role of the probabilistic model has been extended to model dependencies between variables and objectives [11]. In addition, MOEDAs for combinatorial optimization can be notably enhanced by adding a local optimizer that can refine solutions found by sampling the PGM [14,15].

This paper investigates the approach firstly presented in [7], called Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA), which is based on a joint probabilistic modeling of decision variables, objectives, and parameters of the local optimizers. Some recent MOEDA-based approaches model variables and objectives in the same PGM in order to explore their relationship, which means investigating how objectives influence variables and vice versa [11]. However, HMOBEDA also includes a Local Search (LS) parameters in its BN-based PGM. It means that appropriate LS parameters for different configurations of decision variables and objective values are sampled from PGM, which acts like an online parameter tuning approach. This paper aims to extend the work presented in [7] and further expanded in [8] by investigating different sampling mechanisms and also the results presented in [16] by expanding the statistical analysis and applying the considered approaches on two combinatorial MaOPs.

The main objective here is to explore, from a probabilistic point of view, the approximated Pareto Front (PF) using the final PGM structure. Two variants of HMOBEDA which use different ways to fix evidences during the sampling and were not explored in [7,8] are considered in this paper. However, differently from [16], we

provide a more profound investigation of the resulted probabilistic model. We evaluate how the interactions among variables, objectives and local search parameters are captured by the model for the considered MaOPs instances, by exploring the PGM structures learned during the evolutionary process. Besides Hypervolume [17,18] and Inverted Generational Distance (IGD) [19] indicators adopted in [16], this paper also considers a capacity metric called Error Ratio (ER) [19] on the statistical analysis of the results.

In [16], the embedded PGM is evaluated on the Multi-objective Knapsack Problem (MOKP) - the multi-objective version of the well known knapsack problem, which has been recently explored in other works in the literature [20,21]. MOEDAs that use different types of probabilistic models have already been applied to MOKP [14,15], particularly those based on BN [12]. However, these works do not consider the objectives and parameters structured all together in the same BN, as proposed in HMOBEDA. Therefore, HMOBEDA was chosen to be investigated in this work using the same instances of MOKP addressed in [16]. In addition, we extend the analysis to the multi-objective NK-landscape (MNK-landscape) problem [22,23]. This problem represents another combinatorial MOP that has been recently explored by [23–26]. The mono-objective version has also been explored by using EDAs [27–29]. However, these works do not consider objectives and LS parameters structured all together in the same BN.

In the MOKP case, many real-world applications are reported dealing with capital budgeting [30], selection of transportation investment alternatives [31], relocation issues arising in conservation biology [32], planning remediation of contaminated light-station sites [33] and action plan in the social and medico-social sector [34]. The NK model has found application in several fields, ranging from evolutionary biology studies [35,36] to complex systems, such as the representation of complexity in economics and organizational sciences [37]. Therefore, these problems have been chosen due to their combinatorial and multi-objective features and because they might model several other real-world problems.

This paper is organized as follows. The addressed MOPs are presented in Section 2. Section 3 provides a brief introduction to Bayesian Network and EDA PGM. Section 4 details the two variants of HMOBEDA. Results from numerical experiments are shown and discussed in Section 5 with conclusions and future directions presented in Section 6.

## 2. The multi and many-objective problems addressed in this paper

This section presents the problems addressed in this paper. First we investigate the MOKP, according to the multi-objective formulation proposed by [17,20] and [21]. Then we explore the MNK-landscape problem proposed by [22].

### 2.1. The multi-objective knapsack problem

The binary version 0/1 knapsack is a widely studied problem due to its practical importance. In the last years a general formulation has been well studied and a number of algorithms have been proposed. Evolutionary approaches for solving the multi-objective version (MOKP) are of great interest [38] with many works presented in the last years [17,20,21,39–42] and [43], many algorithms for solving the MOKP variant have been proposed using different techniques for real [30–34] and non-real world applications [38].

MOKP can be formulated as follows:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), \dots, z_R(\mathbf{x})) \\ \text{subject to} \quad & \sum_{q=1}^Q b_{rq}x_q \leq c_r, \quad r = 1, \dots, R \\ & \text{with } z_r(\mathbf{x}) = \sum_{q=1}^Q a_{rq}x_q, \quad \mathbf{x} \in \{0, 1\}^Q \end{aligned} \quad (1)$$

where,  $a_{rq}$  is the profit of item  $q$ , according to knapsack  $r$ ,  $b_{rq}$  denotes the weight of item  $q$  according to knapsack  $r$ , and  $c_r$  is the constraint capacity of knapsack  $r$ , given a total of  $R$  objective functions (knapsacks) and  $Q$  items. Decision variables are represented by  $\mathbf{x}$ , a  $Q$ -dimension binary vector, with  $x_q = 1$  indicating that item  $q$  has been selected to be in  $r$ th knapsack, and  $x_q = 0$ , otherwise.

## 2.2. The multi-objective NK-landscape problem

The single NK fitness landscapes is a family of combinatorial problems proposed in [44] aiming to explore 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  $\mathbf{X} = (X_1, \dots, X_N)$  denote a vector of discrete variables and  $\mathbf{x} = (x_1, \dots, x_N)$  an assignment to the variables.

An NK fitness landscape is defined by the following components [28]:

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

Both the subfunction  $f_q$  for each variable  $X_q$  and the neighborhood structure  $\Pi(X_q)$  are randomly set [28].

The mono-objective function  $z_{NK}$  to be maximized is defined as:

$$z_{NK}(\mathbf{X}) = \sum_{q=1}^N f_q(x_q, \Pi(x_q)). \quad (2)$$

For a set of given parameters, the problem consists in finding the global maximum of the function  $z_{NK}$  [45].

The MNK-landscape problem [22,23] is a multi-objective version of the NK fitness landscape model with  $R$  objectives,  $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), z_2(\mathbf{x}), \dots, z_R(\mathbf{x})) : \mathcal{B}^N \rightarrow \mathcal{R}^R$ . 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,<sup>1</sup>  $R$  is the number of objectives,  $z_r(\mathbf{x})$  is the  $r$ -ith objective function, and  $\mathcal{B} = \{0, 1\}$ .  $\mathbf{K} = \{K_1, \dots, K_R\}$  is a set of integers where  $K_r$  is the neighborhood size in the  $r$ th landscape.

MNK-landscape problem can be formulated as follows [23]:

$$\max_{\mathbf{x}} \mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), \dots, z_R(\mathbf{x}))$$

$$\text{subject to } \mathbf{x} \in \{0, 1\}^N,$$

$$\text{with} \quad (3)$$

$$z_r(\mathbf{x}) = \frac{1}{N} \sum_{q=1}^N f_{r,q}(x_q, \Pi_{r,q}(x_q)),$$

$$r \in \{1, \dots, R\},$$

$$q \in \{1, \dots, N\},$$

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

## 2.3. Related approaches

There are many related approaches to solve the problems addressed in this paper. Several probabilistic models can be used to represent the solution set for both 0/1 knapsack and NK-landscape problems. The main contributions of HMOBEDA are the inclusion of local search and the relationship between objectives, variables and local search parameters in the probabilistic model.

Considering the MOKP and starting with the approach presented in [46], we point out that in this work authors present a modified evaluation phase of the single BOA algorithm using the strength criterion concept adopted by the SPEA algorithm, and their experiments are for 2 objectives. The work proposed by [12] has been designed using a probabilistic model based on binary decision trees and a special selection scheme that uses  $\epsilon$ -archives, it was applied to 2, 3 and 4 objectives. Differently, HMOBEDA uses the non-dominated sorting scheme, tournament selection and crowding distance (CD), procedures also used in NSGA-II. Another difference, is that HMOBEDA adopts a local search, models the best solutions using a BN, considers objectives and local search parameters in addition to decision variables as nodes of the network, and can be applied to multi and many objective optimization.

In [14], for MOKP, the following techniques are used: random repair method; random neighborhood structure in local search; stochastic clustering method; a mixture-based UMDA, for 2, 3 and 4 objectives. The differences of this approach and HMOBEDA are the probabilistic model and the local search: in their work the local search is based on a weighted sum method, with weights calculated according to maximization and minimization functions of each objective. In our work the local search weights are uniformly distributed.

An EDA is proposed in [47] to find the solution of the Bounded Knapsack Problem using the same repair method we have used in our work. However their addressed problem considers a single objective, a vector as the probabilistic model (i.e., they assume no relationship among variables) and a mutation probability exchanging a bit from 0 to 1 or 1 to 0 is applied to the sampled solutions in order to overcome a local optimum.

The work presented in [6] proposed an EDA combining two local search strategies addressing the knapsack problem for a single objective. The main difference from this previous research is that HMOBEDA addresses the embedded local search parameters in an EDA framework for MOPs. In [6] the authors present a relevant EDA approach with local search that might be extended to the MOKP to be consider in our future research.

Regarding the MNK-landscape problem, the main difference is that none of the related works considers EDAs for the multi-objective NK-landscape problem and none of them includes LS parameters in PGM.

In [27] and [28] the authors analyze the performance of hBOA, UMDA, and a genetic algorithm (GA) on the mono-objective NK-landscape problem. In [29] the authors also study the behavior of EDAs using UMDA, BMDA and EBNA with different parameter settings. Although they use probabilistic models based on BN, they consider only mono-objective NK-landscape instances.

The work presented in [23] compares NSGA-II and SPEA2 on the generated landscapes using hypervolume indicator, similar to our work, and analyzes the influence of MNK-landscape parameters on several features of the fitness landscape. However, the main difference between that work and ours is that we aim to study a PGM-based MOEDA (HMOBEDA), and compare its variants analyzing the probabilistic model.

The works proposed by [26] and [48] study which (and how) problem features impact the search performance for a global EMO strategy (GSEMO) and a neighborhood-based local search heuristic (PLS) in the  $\rho$ MNK-landscapes, a variant of MNK-landscape model. The differences of these works and ours are that we explore an online tuning of LS parameters, where these parameters are self-adapted as the optimization process evolves.

Table 1 summarizes the main characteristics of EDA-based approaches developed to solve knapsack and NK-landscape problems compared with HMOBEDA.

## 3. Bayesian networks: a brief introduction

One of the most general probabilistic models for discrete variables used in EDAs and MOEDAs is the Bayesian Network [49].

<sup>1</sup> The number of variables  $N$  is noted  $Q$  in Sections 4 and 5.

**Table 1**

Summarizing EDA approaches for the knapsack and NK-landscape problems.

EDA	Problem	Objectives	Fitness assignment	Probabilistic Model	Local Search	References
BOA	MOKP	2	Pareto technique	Bayesian Network	No	[46]
BMOA	MOKP	2, 3 and 4	$(\mu + \lambda, \epsilon)$ Operator	Bayesian Network	No	[12]
MOHEDA	MOKP	2, 3 and 4	Scalarizing Functions	Marginal factorization	Yes	[14]
Greedy EDA	KP	1	Objective function	Probability vector	No	[47]
Hybrid EDA	KP	1	Surrogate/Objective Function	Marginal factorization	Yes	[6]
hBOA	NK	1	Objective function	Bayesian Networks	No	[27]
UMDA	NK	1	Objective function	Marginal factorization	No	[27]
UMDA	NK	1	Objective function	Marginal factorization	No	[29]
BMDA	NK	1	Objective function	Tree-model	No	[29]
EBNA	NK	1	Objective function	Bayesian Networks	No	[29]
HMOBEDA	MOKP	2–5 and 8	Pareto technique	Bayesian Network	Yes	[7]

Bayesian Networks are directed acyclic graphs (DAG) whose nodes represent variables, and whose missing edges encode conditional independencies between variables. Each node is associated with a probability function that takes as input a particular set of values for the node's parent variables and gives the probability of the variable represented by the node [50,51].

Let  $\mathbf{Y} = (Y_1, \dots, Y_M)$  be a vector of  $M$  random variables, and let  $y_m$  be a value of  $Y_m$ , the  $m$ th component of  $\mathbf{Y}$ . The representation of a Bayesian model is given by two components [13]: a structure and a set of local parameters. The set of local parameters  $\Theta$  contains, for each variable, the conditional probability distribution of its values given different value settings for its parents, according to structure  $B$ .

The structure  $B$  for  $\mathbf{Y}$  is a DAG that describes a set of conditional dependencies of all variables in  $\mathbf{Y}$ .  $\mathbf{Pa}_m^B$  represents the set of parents (variables from which an arrow is coming out in  $B$ ) of the variable  $Y_m$  in the PGM whose structure is given by  $B$  [52]. This structure assumes that  $Y_m$  is independent from its non-descendants given  $\mathbf{Pa}_m^B$ ,  $m = 2, \dots, M$ , where  $Y_1$  is the root node.

Therefore, for discrete random variables, a Bayesian Network encodes a factorization for the probability mass function (pmf) as follows:

$$p(\mathbf{y}) = p(y_1, y_2, \dots, y_M) = \prod_{m=1}^M p(y_m | \mathbf{pa}_m^B) \quad (4)$$

In discrete domains, we can assume that  $Y_m$  has  $s_m$  possible values (states),  $y_m^1, \dots, y_m^{s_m}$ , therefore the particular conditional probability,  $p(y_m^k | \mathbf{pa}_m^{j,B})$ , can be defined as:

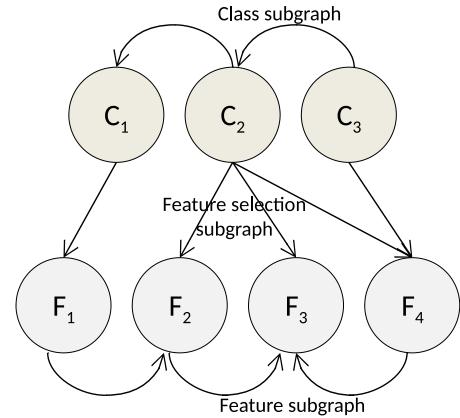
$$p(y_m^k | \mathbf{pa}_m^{j,B}) = \theta_{y_m^k | \mathbf{pa}_m^{j,B}} = \theta_{mjk} \quad (5)$$

where  $\mathbf{pa}_m^{j,B} \in \{\mathbf{pa}_m^{1,B}, \dots, \mathbf{pa}_m^{t_m,B}\}$  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  $\theta_{mjk}$  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  $\Theta = \{\theta_1, \dots, \theta_m, \dots, \theta_M\}$ , where  $\theta_m = (\theta_{m11}, \dots, \theta_{mjk}, \dots, \theta_{m,t_m,s_m})$  and  $M$  the total number of nodes in the BN.

BNs are often used for modeling multinomial data with discrete variables [53] and generate new solutions using the particular conditional probability described in Eq. (5) (probabilistic logic sampling [54]).

Generally, parameters in the whole set  $\Theta$  and the BN structure  $B$  are unknown. Two approaches to estimate the parameters are usually applied: Maximum Likelihood Estimate (MLE) and Bayesian Estimate [52], this last one is used in this work to estimate the parameters.

There are three main approaches to learn BN structures: score-based learning, constraint-based learning, and hybrid methods [55].

**Fig. 1.** An MBN classifier with 3 class variables and 4 feature variables.

The majority of the developed structure learning algorithms fall into the category of score-based approaches, which look for the network structure that maximizes the value of a scoring metric. Usual examples of such approaches are the simple greedy search algorithm, local hill-climbing, simulated annealing, tabu search and evolutionary computation [13], most of them use K2 as the scoring metric [50].

Constraint-based learning methods typically use statistical tests to identify conditional independence relations from data and build a BN structure that best fits those relations [53], such as Incremental Association Markov Blanket (IAMB) [56] and Semi-Interleaved Hiton-PC (SI-HITON-PC) [57]. Hybrid methods aim to integrate the two approaches, like Max-Min Hill Climbing (MMHC) [58].

### 3.1. Naive BN-based models

One of the most important fields of BN application is classification, when the BN nodes represent either classes or instance features. However, many application domains involve instances that are expected to be assigned to the most likely combination of classes [59]. A multidimensional BN (MBN) classifier is a Bayesian Network, with a restricted topology designed to address these applications, which includes one or more class variables and one or more feature variables in its PGM.

An MBN classifier has a set of random variables  $\mathbf{Y}$  partitioned into a set of feature variables  $\mathbf{Y}_F = \{F_1, \dots, F_m\}$ ,  $m \geq 1$ , and a set  $\mathbf{Y}_C = \{C_1, \dots, C_n\}$ ,  $n \geq 1$ , with the class variables. Subgraph  $B_C \subset \mathbf{Y}_C$  is called class subgraph; and  $B_F \subset \mathbf{Y}_F$  is called feature subgraph. Subgraph  $B_{CF}$ , which includes arcs from class to feature variables, is called feature selection subgraph, since it represents the selection of features that are deemed relevant for classification in view of class variables [60]. Fig. 1 illustrates an example of an MBN with  $n = 3$  and  $m = 4$ .



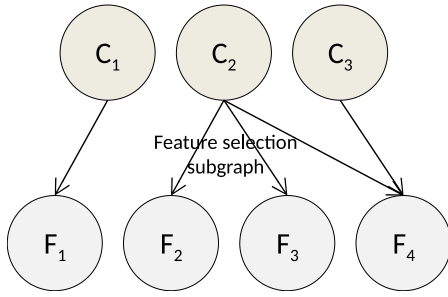


Fig. 2. A naive MBN classifier with 3 class variables and 4 feature variables.

According to [11], an MBN can infer the classes of given feature variables, the most probable feature values for a given combination of classes, and the most probable values for a subset of features or classes given the value of the others.

Different types of MBN classifiers can be distinguished based on their graphical structures. An example is the naive multidimensional classifier in which both the class subgraph and the feature subgraph are empty [59] (i.e. there are edges only in feature selection subgraph, as presented in Fig. 2). This structure is the base of HMOBEDA, therefore the structure learning process consists in finding, for root nodes defined as multiple objectives, a feature selection subgraph (encompassing decision variables and local search parameters in our case) that better fits the data.

Our interest in this work is to apply MBN concepts to MOPs, aiming to answer ((i) mainly (ii) possibly) the following questions: (i) what is the most probable solution resulting in a specific value-setting for the objectives? (ii) what are the estimated objectives values of a given solution?

MBN classifiers use different estimation methods or search algorithms to find out the optimal way to represent data [59], and most of them are score-based [61]. The presence of several classes adds complexity and increases the number of network parameters [62]. Therefore, some authors have suggested methods to overcome these shortcomings such as: K2 algorithms [50,61,63], the Greedy Search [61,64], the Greedy Hill Climber (BN-HC) and the Repeated Hill Climber (BN-RHC) [61,65].

The choice of the K2 algorithm is based on previous studies. In [61], K2 was compared with BN-HC and BN-RHC and again, it achieved a better trade-off between accuracy and computational cost.

The K2 algorithm, introduced by [50], is a score-based greedy local search technique which uses the K2 metric to select the next arc between  $B_C$  and  $B_F$ . It starts by assuming that a node, in an ordered list, does not have any parent, then it processes each node in turn, gradually adding edges from previously processed nodes to the current one. In each step it adds the edge that increases the K2 scoring metric the most. When no edge increases the metric anymore, it turns attention to the next node.

In this work we apply the K2 structure learning algorithm for all variants of HMOBEDA, which will be discussed in the next section.

#### 4. Comparing HMOBEDA versions

HMOBEDA is a hybrid EDA approach introduced in [7]. The term hybrid concerns a local search (LS) mechanism included into its PGM-based framework to improve the performance. This way, LS can be combined with sorting and selection techniques usually adopted in MOEAs and also be configured with suitable parameters during the search. The general scheme of the HMOBEDA is presented in Fig. 3.

As can be noticed, HMOBEDA uses a probabilistic model (based on BN) of objectives, variables and local search parameters to sample new individuals. Every solution is thus represented by a joint

vector with  $Q+R+L$  elements,  $\mathbf{y} = (\mathbf{x}, \mathbf{z}, \mathbf{p}) = (X_1, \dots, X_Q, Z_1, \dots, Z_R, P_1, \dots, P_L)$ , denoting the decision variables ( $X_1, \dots, X_Q$ ), objectives ( $Z_1, \dots, Z_R$ ) and LS parameters ( $P_1, \dots, P_L$ ).

At the first iteration, solutions are randomly generated in the *Initialization* process and the corresponding objectives are calculated through the objective functions of the addressed problem. A *Local Search* based on Hill Climbing procedure (HCLS) [66] generates a neighbor for each solution, then HMOBEDA calculates its fitness and updates the original solution in case the neighbor has a better fitness.

In order to select a total of  $N_{PGM}$  individuals from the current population, the Non-dominated Sorting (ND) [67] technique is applied. 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 (CD) [68].

Aiming to learn the probabilistic model, the BN structure and parameters are estimated in the *PGM Learning* block. Different algorithms can be considered for the *PGM Learning* block: in this work we use the K2 algorithm with K2 metric running over a set of  $N_{PGM}$  best individuals each one composed of  $Q$  decision variables,  $R$  objectives and  $L$  local search parameters, (i.e.  $\mathbf{Y} = (Y_1, \dots, Y_M) = (Z_1, \dots, Z_R, X_1, \dots, X_Q, P_1, \dots, P_L)$ ). In [69] some experiments were conducted and we concluded that K2-algorithm was a good option for the structure prediction of HMOBEDA, as well CD for tie-breaker criteria. This way the BN structure encodes a factorization of the joint probability distributions or the probability mass function (pmf) given by:

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

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  $\mathbf{pa}_q^B \subseteq \{Z_1, \dots, Z_R\}$ ,  $\mathbf{pa}_l^B \subseteq \{Z_1, \dots, Z_R\}$ ,  $\mathbf{pa}_r^B = \emptyset$ , which means  $p(z_r | \mathbf{pa}_r^B) = p(z_r)$  for  $r = 1, \dots, R$ .

In the *Sampling* block, the obtained PGM is used to sample the set of new individuals. As discussed in [16], 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. This naive Bayesian model (see Fig. 3) 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 ( $X_1, \dots, X_Q$ ) and LS parameters ( $P_1, \dots, P_L$ ) more related to the objectives fixed as evidences can be drawn for each new individual, and this paper aims at exploring different ways of fixing evidences.

Finally, objectives ( $Z_1, \dots, Z_R$ ) 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  $g + 1$ , and the main loop continues until the stop condition is achieved.

In this paper, we explore two variants of HMOBEDA named HMOBEDA<sub>EXT</sub> and HMOBEDA<sub>CPT</sub> which were proposed in [16] and represent different possibilities to guide the search.

As shown in Fig. 4(b), HMOBEDA<sub>CPT</sub> uses *priori* probabilities of  $Z_1, \dots, Z_R$  described in the Conditional Probability Table (CPT) to draw and further fix evidences. Fig. 4(c) shows that HMOBEDA<sub>EXT</sub>, besides considering  $Z^*$  like the HMOBEDA version, it also includes the extreme points of the approximated Pareto Front (PF) as candidates for the evidences in the root nodes. In HMOBEDA<sub>EXT</sub>, these values are fixed at each generation considering the same probabilities for all candidates:  $Z^*$  and the total number of objectives (each in every extreme of the current approximation Pareto Front).

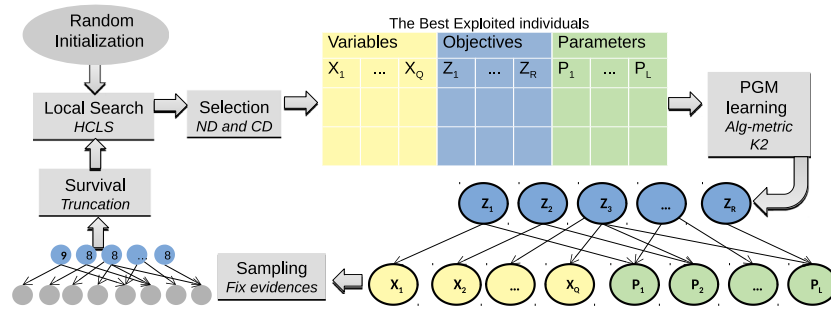
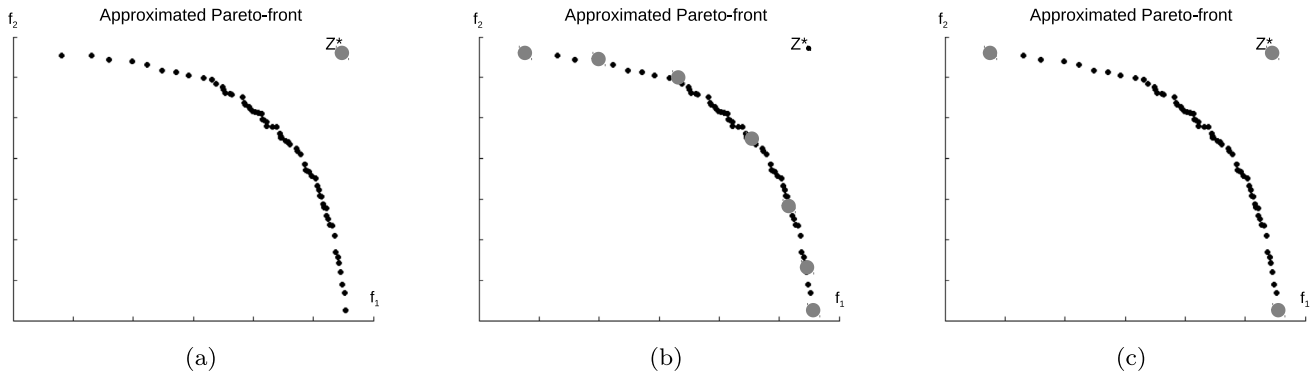


Fig. 3. The HMOBEDA Framework.

Fig. 4. Evidences (gray circles) from the approximated Pareto Front with 2 objectives for (a) HMOBEDA, (b) HMOBEDA<sub>CPT</sub> and (c) HMOBEDA<sub>EXT</sub>.

## 5. Experiments and results

First we compare the original version of HMOBEDA with the two considered variants over MOKP instances through the Hypervolume ( $HV^-$ ) [17,18] and Inverted Generational Distance (IGD) [19], which are usually adopted for measuring the quality of the optimal solution set for multi and many-objective optimization [70].

After that, we compare the best version of HMOBEDA with cutting edge approaches, over MOKP and MNK-landscape instances. In this case besides  $HV^-$  and IGD we also consider a capacity metric called Error Ratio (ER) [19].

Hypervolume considers the difference ( $HV^-$ ) between the hypervolume of the corresponding objectives from both the solution set of an algorithm and the reference set. The IGD metric is the average distance from point in the PF associated with the reference set to the nearest point in the approximated PF. So, smaller values of  $HV^-$  and IGD correspond to higher quality solutions in non-dominated sets indicating both better convergence and good coverage of the reference set. The Error Ratio considers the ratio of solutions from the approximated set that are not in the reference set, so smaller values also indicate better performances. We use the PISA framework [71] to compute  $HV^-$  and Matlab to compute the remaining metrics.

The optimal Pareto Front (PF) 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 approaches addressed in [7] in addition to those obtained by the HMOBEDA versions over all executions.

In the conducted experiments concerning MOKP  $a_q^r$  and  $b_q^r$  are specified as integers in the interval [10, 100]. The capacity  $c_r$  is defined as 50% of the sum of all weights related to each knapsack  $r$  as in [17]. The set of instances (characterized as  $R-Q$ ) is the union of those considered in [17,20,21]. We use thus instances with 100 and 250 items, 2 to 5 and 8 objectives. If an infeasible solution is generated, we apply the same greedy repair method used in [7,17]. This method repairs a solution by removing items in an ascending

order of the relation profit/weight until the constraints are satisfied.

We also considered MNK-landscape instances with  $K \in \{2, 4, 6, 8, 10\}$ , objective space dimension  $R \in \{2, 3, 5, 8\}$  and size  $N = Q \in \{20, 50, 100\}$ , 60 instances (one for each combination of  $R$ ,  $K$  and  $Q$ ).

As in previous works [7,8,16,69], we estimate the parameters through Bayesian Estimate, using the Dirichlet prior, and we adopt K2 metric as score-based technique, although any other method could be used as well. In a preliminary study on the HMOBEDA framework, constraint-based learning methods such as IAMB [56], SI-HITON-PC [57] and hybrid methods like MMHC [58] were also investigated. However, the experiments have shown that K2 metric as score-based technique is a good compromise between accuracy and computational cost.

The K2 algorithm is used here by setting parent nodes as objectives in the network. Therefore, fixing objective values as target evidences enables a straightforward estimation of their associated decision variables and LS parameters.

The parameters for all the HMOBEDA versions considered in this section are: population size ( $N = 100$ ), number of selected individuals for PGM building  $N_{PGM} = N/2$ , Number of sampled individuals  $N_{smp} = 10 * N$ . The LS online configuration adopted by HMOBEDA versions during the evolution assumes the following elements in the vector  $\mathbf{p}$ : the number of LS iterations  $N_{iter} \in \{5, 6, \dots, 20\}$ ; the type of neighbor fitness calculation  $T_{Fnbh} \in \{1, 2\}$ : with (1) representing Linear Combination of objectives and (2) Alternation of objectives (i.e., one by one for each LS iteration); the neighborhood type  $T_{nbh} \in \{1, 2\}$ : with (1) defining double bit-flip operator and (2) single bit-flip from 0 to 1. These parameters have been defined experimentally in previous work [7,8,16,69].

Due to heterogeneous computer hardware and to be fair enough, we define a stop condition based on maximum number of fitness evaluations ( $Max_{eval}$ ), which includes repair procedures and LS-iterations. Then, all algorithms stop when the total number of fitness computations achieves 200,000 evaluations. A total of 30

independent executions are conducted for each algorithm to get average performance metrics.

We use the Shapiro–Wilk normality test [72] to verify whether performance metric results are normally distributed. In case of non-normal distribution, Kruskal–Wallis [73] and Dunn–Sidak's post-hoc tests are applied for statistical analysis considering all approaches being compared. For normal distributions the analyses of variance test (ANOVA) can be applied [73]. All tests have been executed with a confidence level of 95% ( $\alpha = 5\%$ ).

### 5.1. Comparing HMOBEDA alternative versions

In this section, we compare the original HMOBEDA with two modified versions: HMOBEDA<sub>CPT</sub> and HMOBEDA<sub>EXT</sub>. As discussed in Section 4, the differences among them are mainly in the sampling step.

Based on the assumptions of each variant, we aim to investigate how the evidences may influence the distribution of non-dominated solutions along the approximated PF.

In addition to HMOBEDA performance in terms of HV<sup>−</sup> and IGD indicators, this section aims to evaluate the quality of the final approximation of the PF through the analysis of the final achieved PGM. This way, we want to explore one of the main advantages of using EDA: the possibility of scrutinizing its probabilistic model which encompasses relationship among variables encoded in the nodes.

Table 2 shows the statistical analysis of pairwise comparisons between HMOBEDA, HMOBEDA<sub>EXT</sub> and HMOBEDA<sub>CPT</sub> for each instance with respect to HV<sup>−</sup> and IGD values, respectively. The first number shows how many algorithms are better than the algorithm listed in the corresponding line, and the second number shows how many algorithms are worse. When entries present no statistically significant difference the background is emphasized in light blue. The entry related to the algorithm with the lowest (best) average metric is highlighted in bold.

Although Table 2 shows that there is no statistically significant differences between HMOBEDA, HMOBEDA<sub>CPT</sub> and HMOBEDA<sub>EXT</sub> regarding HV<sup>−</sup> indicator for all instances, there are statistically significant differences regarding the IGD metric for 5–100, 5–250, 8–100 and 8–250 instances. HMOBEDA and HMOBEDA<sub>EXT</sub> present better results than HMOBEDA<sub>CPT</sub> for 2–250, 3–100, 3–250, 5–100 and 5–250 with some advantage for HMOBEDA<sub>EXT</sub>. Besides, HMOBEDA<sub>EXT</sub> is the best approach for 8–100 and 8–250 instances.

The average run time for each algorithm is presented in Table 3, with the respective post-hoc test showed in Table 4. We can observe that the computational efforts for all algorithms are similar, as expected, keeping the run time at practical levels. However, increasing the number of objectives and variables severely impacts the average computational time of all approaches. The run times range from approximately 3 min (instance 2–100) to 66 min (instance 8–250) for the three HMOBEDA versions, with no statistically significant differences between them for the same instance.

In order to analyze the final approximated PF achieved by each HMOBEDA version taking into account the information available in the final PGM, we calculate, for each possible solution in the approximated PF, the pmf  $P(\mathbf{y})$ , defined by Eq. (6), from the final set (dominated and non-dominated solutions) achieved at the end of each execution. After that, the mean of pmf values along all executions is obtained for further calculating the marginal distribution  $P(Z_1 = z_1, \dots, Z_R = z_R)$ . The probabilistic view of the Pareto front is then defined by gathering all non-dominated solutions obtained over all 30 executions. Each non-dominated solution is represented by a circle, which is proportional to the corresponding marginal probability  $P(Z_1 = z_1, \dots, Z_R = z_R)$ .

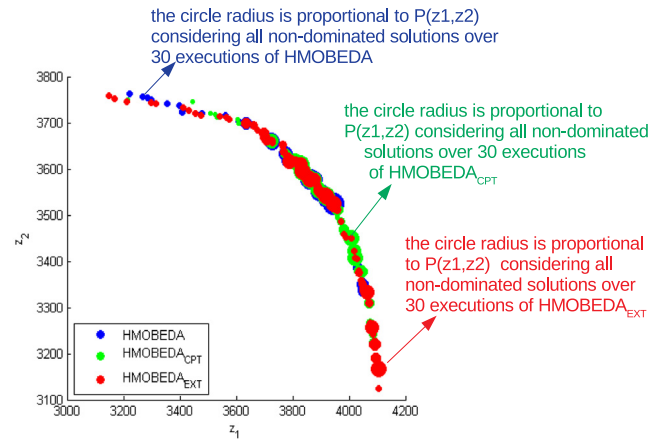


Fig. 5. A probabilistic view of the approximated PF for 2 objectives ( $z_1, z_2$ ) and 100 items for HMOBEDA, HMOBEDA<sub>EXT</sub> and HMOBEDA<sub>CPT</sub>.

Fig. 5 presents the analysis of the approximated PF of all HMOBEDA versions for the instance 2–100 considering the probabilistic information of the PGM. Although PFs seem similar in terms of convergence, we can observe that HMOBEDA and HMOBEDA<sub>CPT</sub> provide solutions concentrated on a particular region of the PF (around the ideal point), region named Pareto Front knee. However, for HMOBEDA<sub>EXT</sub> solutions are better distributed, as points located near the extreme point associated with objective  $Z_1$  present higher probabilities than those obtained by HMOBEDA and HMOBEDA<sub>CPT</sub>.

Since it is difficult to visualize the PF for more than two objectives, Figs. 6(a) and 6(b) illustrate a simplified way to view the approximation: the ordered Euclidean distance between each point from the approximated PF and the ideal point. We note that all the approaches present similar Euclidean distances in Fig. 6(a) for instance 2–100, indicating similar convergence for all the approaches, however, for instance 8–100, HMOBEDA presents the smoothest distance fluctuation in Fig. 6(b). This observation, associated with the IGD values in Table 5, indicates that HMOBEDA provides an approximation of the PF with a slightly high concentration around the ideal point.

Notice that in Fig. 6(a), it is easy to see that each solution/circle in the graph is proportional to its marginal probability  $P(z_1, z_2)$ . Therefore, in Fig. 6(a), we can conclude that for 2 objectives the solutions present different probabilities along the PF, with some points concentrated around the ideal point, as also noticeable in Fig. 5.

We note, in Fig. 6(b), that for 8 objectives, the solutions present similar probabilities since there are few large points plotted in the PF. However, HMOBEDA<sub>EXT</sub> and HMOBEDA<sub>CPT</sub> produce solutions with higher diversity and smaller mean distances from ideal point than HMOBEDA: some solutions might be close to the ideal point but others (near the extreme points) can be far away.

With these experiments, we can conclude that, examining BN structures to provide (after some additional calculations) the marginal distribution of the corresponding objectives values taken over all the algorithm executions, enables the analysis of the influence of fixing evidences in the HMOBEDA approach.

According to the results we note that fixing evidences along the evolutionary process can guide the search through specific regions of the PF, providing different convergence and diversity. HMOBEDA has fixed the highest values for objectives as evidences in the network, providing solutions concentrated around the ideal point. On the other hand, when the evidences are chosen based on a uniform distribution of ideal and extreme points (i.e. with

**Table 2**

Results for pairwise comparisons between HMOBEDA, HMOBEDA<sub>EXT</sub> and HMOBEDA<sub>CPT</sub> using Kruskal–Wallis and Dunn–Sidak's post-hoc tests with  $\alpha = 5\%$  for each problem instance.

Algorithm	2-100	2-250	3-100	3-250	4-100	4-250	5-100	5-250	8-100	8-250
Hypervolume differences (HV <sup>-</sup> )										
HMOBEDA	(0, 0)	<b>(0,0)</b>	<b>(0,0)</b>	<b>(0,0)</b>	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
HMOBEDA <sub>EXT</sub>	(0, 0)	(0, 0)	(0, 0)	(0, 0)	<b>(0,0)</b>	<b>(0,0)</b>	<b>(0,0)</b>	<b>(0,0)</b>	<b>(0,0)</b>	<b>(0,0)</b>
HMOBEDA <sub>CPT</sub>	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
IGD metric										
HMOBEDA	(0, 0)	(0, 1)	<b>(0,1)</b>	<b>(0,1)</b>	(0, 0)	(0, 0)	(1, 1)	(1, 1)	(1, 0)	(1, 0)
HMOBEDA <sub>EXT</sub>	<b>(0,0)</b>	<b>(0,1)</b>	(0, 1)	(0, 1)	<b>(0,0)</b>	<b>(0,0)</b>	<b>(0,2)</b>	<b>(0,2)</b>	<b>(0,2)</b>	<b>(0,2)</b>
HMOBEDA <sub>CPT</sub>	(0, 0)	(2, 0)	(2, 0)	(2, 0)	(0, 0)	(0, 0)	(2, 0)	(2, 0)	(1, 0)	(1, 0)

**Table 3**

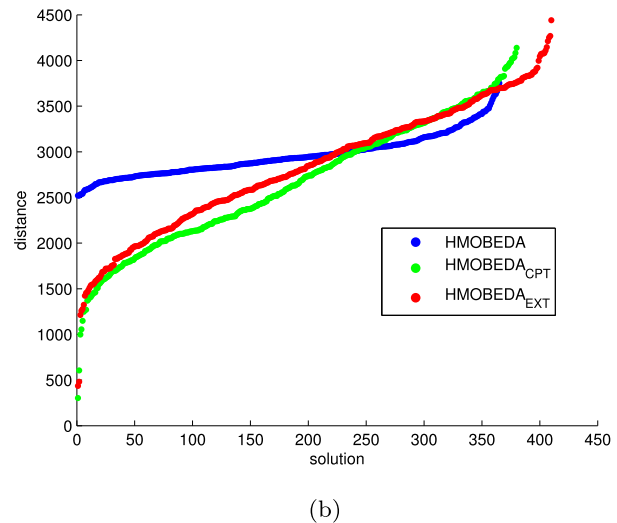
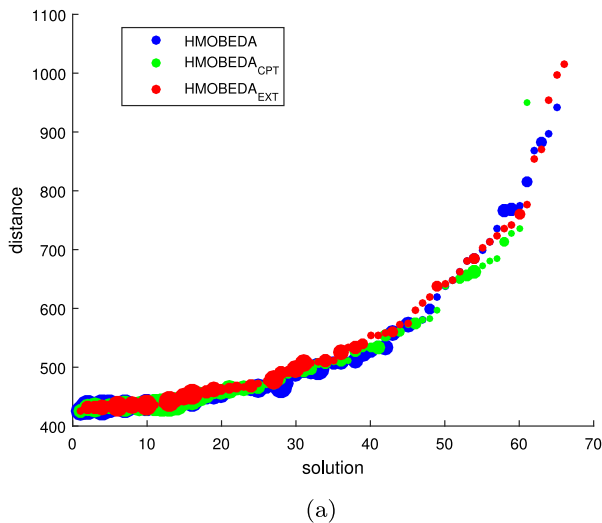
Average run time (min) for each algorithm and instance.

Algorithm	2-100	2-250	3-100	3-250	4-100	4-250	5-100	5-250	8-100	8-250
HMOBEDA	2.45	5.19	5.31	14.58	9.15	24.24	23.54	52.48	34.24	65.54
HMOBEDA <sub>EXT</sub>	2.42	5.34	5.39	14.27	9.84	24.51	23.49	52.54	34.27	65.13
HMOBEDA <sub>CPT</sub>	2.39	5.26	5.15	14.48	9.57	24.58	23.98	52.84	34.21	65.27

**Table 4**

Results for pairwise run time comparisons between HMOBEDA, HMOBEDA<sub>CPT</sub> and HMOBEDA<sub>EXT</sub> versions using Kruskal–Wallis and Dunn–Sidak's post-hoc tests with  $\alpha = 5\%$  for each problem instance.

Algorithm	2-100	2-250	3-100	3-250	4-100	4-250	5-100	5-250	8-100	8-250
HMOBEDA	(0, 0)	<b>(0,0)</b>	(0, 0)	(0, 0)	<b>(0,0)</b>	<b>(0,0)</b>	(0, 0)	<b>(0,0)</b>	(0, 0)	(0, 0)
HMOBEDA <sub>EXT</sub>	(0, 0)	(0, 0)	(0, 0)	<b>(0,0)</b>	(0, 0)	(0, 0)	<b>(0,0)</b>	(0, 0)	(0, 0)	<b>(0,0)</b>
HMOBEDA <sub>CPT</sub>	<b>(0,0)</b>	(0, 0)	<b>(0,0)</b>	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	<b>(0,0)</b>	(0, 0)

**Fig. 6.** The Euclidean distance between each solution and the ideal point for (a) 2 objectives and (b) 8 objectives.

equally likely chances of occurrence for each of them), like in HMOBEDA<sub>EXT</sub>, the solutions are better distributed along the front.

These observations are based on the probability of the solutions along the PF and Euclidean distance fluctuations to the ideal point presented in Figs. 6(a) and 6(b), and support the results presented in Table 2, which show that HMOBEDA<sub>EXT</sub> provides higher diversity (lowest IGD values) especially when the number of objectives increases. Therefore, in the next section we compare HMOBEDA<sub>EXT</sub> with cutting edges approaches on both MOKP and MNK-landscape instances.

## 5.2. Comparing HMOBEDA<sub>EXT</sub> with cutting edge approaches

In this section we compare the HMOBEDA<sub>EXT</sub> version with MBN-EDA [11], NSGA-II [68], S-MOGLS [74] (NSGA-II with local search), MOEA/D [75] and NSGA-III [76].

All algorithms used in the comparison are the original ones found in the literature. The exception is NSGA-III that has been

adapted for combinatorial optimization. As discussed by [74], for S-MOGLS we set the probabilities  $P_{ls}$  and bit-flip operation in LS as 0.1 and 4/500, respectively; the number of neighbors ( $N_{ls}$ ) to be examined as 20. For NSGA-III, we adopt the same configuration used by [76], i.e., the number of reference points ( $H$ ) defines the population size  $N$ :  $H = C_{R+p-1}^p$ . In MOEA/D, the number of subproblems equals the population size  $N$  and the weight vectors  $\lambda^1, \dots, \lambda^N$  are controlled by the configuration parameter  $W$ , calculated as proposed by [75]. As discussed by [76], the size of the neighborhood for each weight vector is  $T = 10$ . MOEA/D considers the weighted sum approach.<sup>2</sup>

All algorithms are run on an AMD Opteron Processor 6378 server, CPU 2.40 GHz machine with 125 GB of RAM, running Linux.

In the tables, the best values are highlighted in bold and results with no statistically significant differences with the best values

<sup>2</sup> This approach is usual for MOKP [75], and can be downloaded from <http://dces.essex.ac.uk/staff/zhang/webofmoead.htm>. It is also suggested by [20].



**Table 5**Average HV<sup>-</sup> and IGD over 30 executions.

Instance	2-100	2-250	3-100	3-250	4-100	4-250	5-100	5-250	8-100	8-250
Average hypervolume differences (HV <sup>-</sup> )										
	x10 <sup>7</sup>	x10 <sup>8</sup>	x10 <sup>10</sup>	x10 <sup>11</sup>	x10 <sup>14</sup>	x10 <sup>15</sup>	x10 <sup>17</sup>	x10 <sup>19</sup>	x10 <sup>29</sup>	x10 <sup>33</sup>
HMOBEDA <sub>EXT</sub>	<b>1.5091</b>	1.0550	<b>2.4284</b>	<b>4.1649</b>	<b>1.7502</b>	<b>4.5288</b>	<b>5.2559</b>	<b>5.4919</b>	<b>8.0724</b>	<b>1.3427</b>
MBN-EDA	1.5801	1.0970	3.1139	4.9970	2.1104	5.7474	6.7614	6.4183	8.2010	1.4549
NSGA-II	1.5572	1.1006	2.9675	5.0176	2.0044	5.6352	6.4577	6.2135	8.1963	1.4601
S-MOGLS	1.6866	1.1006	2.9886	5.0135	1.9956	5.6412	6.3791	6.2463	8.1951	1.4537
NSGA-III	1.5121	1.0866	2.4461	4.2254	1.7815	5.4369	6.3693	6.0640	8.1820	1.4423
MOEA/D	1.5174	<b>0.7219</b>	2.4487	4.2432	1.8215	5.2503	6.4815	5.8908	8.1914	1.4467
Average IGD metric										
	x10 <sup>3</sup>	x10 <sup>3</sup>	x1	x1	x1	x1	x1	x1	x10 <sup>3</sup>	x10 <sup>4</sup>
HMOBEDA <sub>EXT</sub>	2.4960	6.5290	<b>21.7847</b>	<b>44.2044</b>	18.2645	<b>26.3493</b>	<b>10.2432</b>	<b>16.3464</b>	<b>8.0231</b>	<b>1.3235</b>
MBN-EDA	2.5449	6.7565	32.9048	53.4658	23.8543	30.2950	14.8650	21.9795	8.2153	1.6470
NSGA-II	2.5286	6.7907	32.1872	55.1838	23.8436	30.0147	15.1695	22.8114	8.4397	1.7084
S-MOGLS	2.7496	6.7745	32.4780	54.6336	23.6078	29.9242	15.2941	22.5989	8.3997	1.7059
NSGA-III	<b>2.4931</b>	6.7602	22.5150	45.8885	<b>18.1235</b>	29.5950	14.1986	20.9275	8.2402	1.5197
MOEA/D	2.4992	<b>5.0465</b>	23.0015	45.9216	20.3025	30.2128	14.6657	21.9553	8.1993	1.5268

**Table 6**

Capacity metrics over 30 executions for each algorithm. The best values are in bold.

Algorithm	2-100	2-250	3-100	3-250	4-100	4-250	5-100	5-250	8-100	8-250
Ref	72	72	408	220	182	619	452	957	1133	1120
Error Ratio (ER)										
HMOBEDA <sub>EXT</sub>	0.7222	0.6944	<b>0.6627</b>	<b>0.5927</b>	<b>0.5330</b>	<b>0.4911</b>	<b>0.2898</b>	<b>0.4619</b>	<b>0.1882</b>	<b>0.1295</b>
MBN-EDA	0.9114	0.9411	0.9115	0.9409	0.9286	0.9600	0.9712	0.9697	0.9982	0.9982
NSGA-II	0.9083	0.9317	0.9755	0.9091	0.9743	0.9919	0.9956	1.0000	1.0000	1.0000
S-MOGLS	0.9444	0.9097	0.9755	0.9768	0.9725	0.9935	0.9889	0.9801	1.0000	1.0000
NSGA-III	<b>0.6697</b>	0.9000	0.7304	0.9000	0.6484	0.7932	0.8761	0.7868	0.9153	0.9330
MOEA/D	0.8400	<b>0.6250</b>	0.7451	0.6818	0.9451	0.7706	0.8783	0.8015	0.8985	0.9393

**Table 7**

Average Run time (min) for each algorithm and instance.

Algorithm	2-100	2-250	3-100	3-250	4-100	4-250	5-100	5-250	8-100	8-250
HMOBEDA <sub>EXT</sub>	2.42	5.34	5.39	14.27	9.84	24.51	<b>23.49</b>	52.54	34.27	65.13
MBN-EDA	1.54	5.21	3.87	12.90	6.41	25.76	26.15	55.89	<b>31.81</b>	<b>62.45</b>
NSGA-II	<b>0.87</b>	<b>2.21</b>	<b>1.57</b>	<b>5.87</b>	<b>5.89</b>	<b>20.13</b>	26.51	<b>50.32</b>	35.27	66.18
S-MOGLS	1.76	3.53	4.12	11.10	8.91	22.67	30.67	56.76	37.18	67.03
NSGA-III	2.91	5.18	5.41	14.90	9.41	24.90	22.85	54.23	34.54	67.04
MOEA/D	2.82	5.01	5.44	14.50	9.34	23.70	23.76	52.87	34.61	66.55

are emphasized in light blue for each instance, using Kruskal–Wallis and Dunn–Sidak's post-hoc tests with a significance level of  $\alpha = 5\%$ .

### 5.2.1. Experiments with MOKP

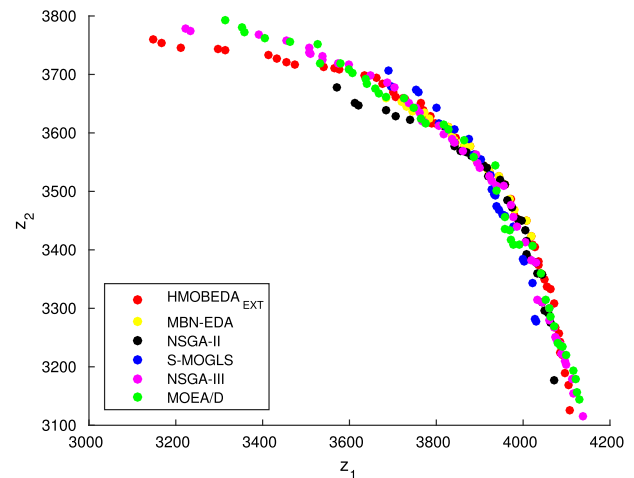
Table 5 shows the hypervolume difference (HV<sup>-</sup>) and IGD metric, both averaged over 30 executions for HMOBEDA<sub>EXT</sub> and cutting edge evolutionary approaches.

We note that there is no statistically significant differences between HMOBEDA<sub>EXT</sub>, NSGA-III and MOEA/D regarding both HV<sup>-</sup> indicator and IGD metric for the 2-100, 3-100 and 3-250 instances. However, HMOBEDA<sub>EXT</sub> presents the lowest values among all other approaches for 4-250, 5-100, 5-250, 8-100 and 8-250 instances.

Table 6 presents the capacity metric ER, which used an approximated front of each algorithm calculated over 30 executions and the reference defined as all non-dominated solutions obtained by all algorithms over all executions.

We can observe that HMOBEDA<sub>EXT</sub> is better than the other approaches for almost all instances regarding ER (lowest values), except for 2-100 and 2-250 instance. Therefore, we can affirm that HMOBEDA<sub>EXT</sub> is a competitive approach for the MOKP particularly for many-objective instances and can be considered a state-of-the-art algorithm for the instances considered here.

Fig. 7 presents the approximated PFs for 2-100 instances for HMOBEDA<sub>EXT</sub>, MBN-EDA, NSGA-II, S-MOGLS, MOEA/D and NSGA-III. Each approximated PF is composed gathering all non-dominated solutions over the 30 executions. Note that this representation is different from the corresponding HV<sup>-</sup> and IGD metrics which are averaged over 30 executions.



**Fig. 7.** The approximated PFs for 2 objectives ( $z_1, z_2$ ) and 100 items for HMOBEDA<sub>EXT</sub>, MBN-EDA, NSGA-II, S-MOGLS, MOEA/D and NSGA-III.

We can observe that, for 2-100 instance, HMOBEDA<sub>EXT</sub>, MOEA/D and NSGA-III present the solutions well distributed along the entire front.

The average run time for each algorithm is presented in Table 7 only as guidance of the corresponding computational effort.

We can observe from Table 7 that the computational efforts required by each algorithm for the same instance are similar to

**Table 8**  
Average HV<sup>-</sup> over 30 independent executions.

Algorithm	Average hypervolume differences (HV <sup>-</sup> )					
	HMOBEDA <sub>EXT</sub>	MBN-EDA	NSGA-II	S-MOGLS	NSGA-III	MOEA/D
M2N20K2	0.11601	0.47900	0.49704	0.51590	0.11463	<b>0.10534</b>
M2N20K4	0.09880	0.52351	0.54506	0.55554	<b>0.09534</b>	0.09612
M2N20K6	0.08375	0.52550	0.51571	0.52295	<b>0.08066</b>	0.08121
M2N20K8	<b>0.08369</b>	0.51777	0.51541	0.51834	0.08548	0.09259
M2N20K10	<b>0.08086</b>	0.49878	0.49253	0.50265	0.08111	0.08344
M2N50K2	<b>0.06026</b>	0.44380	0.45590	0.47286	0.09033	0.11873
M2N50K4	<b>0.06188</b>	0.50516	0.50135	0.52737	0.10549	0.13829
M2N50K6	<b>0.04358</b>	0.47313	0.47007	0.49150	0.10547	0.12471
M2N50K8	<b>0.05705</b>	0.44096	0.43702	0.45979	0.11891	0.12500
M2N50K10	<b>0.04334</b>	0.33153	0.33919	0.35620	0.04874	0.04851
M2N100K2	<b>0.05269</b>	0.42464	0.43374	0.44931	0.10483	0.12004
M2N100K4	<b>0.03939</b>	0.45249	0.45964	0.48457	0.13176	0.14831
M2N100K6	<b>0.07622</b>	0.43678	0.44017	0.45551	0.15507	0.16818
M2N100K8	<b>0.05054</b>	0.30400	0.30569	0.32625	0.05351	0.05209
M2N100K10	0.04264	0.27860	0.27225	0.29621	<b>0.03932</b>	0.04080
M3N20K2	<b>0.12614</b>	0.38089	0.38218	0.40413	0.12656	0.14033
M3N20K4	0.09641	0.39954	0.39400	0.40857	0.10170	<b>0.09151</b>
M3N20K6	0.10178	0.41152	0.40721	0.41239	<b>0.09852</b>	0.10329
M3N20K8	<b>0.08725</b>	0.38262	0.37600	0.39207	0.09217	0.09333
M3N20K10	<b>0.07339</b>	0.35889	0.35904	0.36278	0.07660	0.07656
M3N50K2	<b>0.07459</b>	0.34824	0.32923	0.34722	0.11470	0.12905
M3N50K4	0.07618	0.37666	0.35558	0.38003	<b>0.12625</b>	0.13629
M3N50K6	<b>0.05955</b>	0.35323	0.33518	0.35396	0.12372	0.13490
M3N50K8	<b>0.09064</b>	0.32901	0.31501	0.33111	0.13303	0.13344
M3N50K10	<b>0.03908</b>	0.22928	0.21717	0.23668	0.04214	0.04620
M3N100K2	<b>0.04747</b>	0.31130	0.28710	0.30722	0.10605	0.11367
M3N100K4	<b>0.04230</b>	0.32941	0.30443	0.32996	0.12814	0.13998
M3N100K6	<b>0.04848</b>	0.32150	0.29838	0.31977	0.14241	0.15180
M3N100K8	<b>0.09785</b>	0.26909	0.24519	0.27049	0.11340	0.11646
M3N100K10	<b>0.03386</b>	0.17547	0.15512	0.17938	0.03660	0.03616
M5N20K2	<b>0.11121</b>	0.21689	0.20636	0.20273	0.18351	0.23320
M5N20K4	<b>0.09549</b>	0.23163	0.21516	0.23760	0.18757	0.24871
M5N20K6	<b>0.08491</b>	0.22297	0.19679	0.20196	0.17860	0.23994
M5N20K8	<b>0.08248</b>	0.20895	0.20493	0.18128	0.17412	0.23374
M5N20K10	<b>0.08266</b>	0.20812	0.20240	0.22737	0.17212	0.22745
M5N50K2	<b>0.05579</b>	0.16104	0.18568	0.16104	0.11668	0.16580
M5N50K4	<b>0.05293</b>	0.17219	0.16786	0.17127	0.11978	0.17970
M5N50K6	<b>0.05678</b>	0.15587	0.13926	0.17006	0.10730	0.16460
M5N50K8	<b>0.03715</b>	0.12062	0.13992	0.12782	0.07574	0.13010
M5N50K10	<b>0.03619</b>	0.11462	0.12881	0.11858	0.07201	0.12377
M5N100K2	<b>0.03723</b>	0.13228	0.12925	0.14883	0.08685	0.13329
M5N100K4	<b>0.03321</b>	0.13203	0.11220	0.14432	0.08119	0.13286
M5N100K6	<b>0.03226</b>	0.10459	0.09999	0.10714	0.05507	0.10780
M5N100K8	<b>0.02874</b>	0.08882	0.08745	0.08272	0.04487	0.09522
M5N100K10	<b>0.02874</b>	0.08676	0.08921	0.08107	0.04178	0.09107
M8N20K2	<b>0.00373</b>	0.00584	0.00572	0.00649	0.00639	0.00669
M8N20K4	<b>0.00464</b>	0.00741	0.00770	0.00627	0.00811	0.00874
M8N20K6	<b>0.00524</b>	0.00762	0.00792	0.00759	0.00891	0.00963
M8N20K8	<b>0.00546</b>	0.00768	0.00671	0.00616	0.00929	0.00906
M8N20K10	<b>0.00473</b>	0.00641	0.00583	0.00704	0.00779	0.00751
M8N50K2	<b>0.00068</b>	0.00157	0.00143	0.00176	0.00148	0.00190
M8N50K4	<b>0.00078</b>	0.00197	0.00187	0.00198	0.00170	0.00232
M8N50K6	<b>0.00083</b>	0.00191	0.00187	0.00187	0.00160	0.00230
M8N50K8	<b>0.00078</b>	0.00182	0.00187	0.00132	0.00140	0.00194
M8N50K10	<b>0.00055</b>	0.00112	0.00077	0.00121	0.00097	0.00137
M8N100K2	<b>0.00018</b>	0.00056	0.00055	0.00044	0.00049	0.00068
M8N100K4	<b>0.00030</b>	0.00092	0.00099	0.00066	0.00071	0.00105
M8N100K6	<b>0.00024</b>	0.00075	0.00066	0.00077	0.00053	0.00084
M8N100K8	<b>0.00025</b>	0.00073	0.00077	0.00066	0.00050	0.00081
M8N100K10	<b>0.00023</b>	0.00060	0.00066	0.00066	0.00044	0.00070

each other, keeping the run time at practical levels. However, increasing the number of objectives and variables severely impacts the average computational time of all approaches.

The results also show that NSGA-II presents the lowest computational time for all instances with 2, 3 and 4 objectives. However, HMOBEDA<sub>EXT</sub> presents competitive results in comparison with other approaches for instances with 5 and 8 objectives.

### 5.2.2. Experiments with MNK-landscape problem

In this section we aim to compare the results of HMOBEDA<sub>EXT</sub>, MBN-EDA, NSGA-II, S-MOGLS, MOEA/D and NSGA-III for solving MNK-landscape instances.

Tables 8 and 9 show the hypervolume difference (HV<sup>-</sup>), IGD metric respectively, both averaged over 30 executions of each

algorithm. There is no statistically significant differences between HMOBEDA<sub>EXT</sub>, NSGA-III and MOEA/D regarding both HV<sup>-</sup> indicator and IGD metric for several instances with 2 and 3 objectives. However, there are statistically significant differences for all the remaining instances for both HV<sup>-</sup> indicator and IGD metric, where HMOBEDA<sub>EXT</sub> presents the best values in comparison with all other approaches.

Table 10 presents the capacity metrics which, as in the previous case, uses the approximated front of each algorithm calculated over 30 executions and the reference defined as all non-dominated solutions obtained by all algorithms over all executions.

We can observe that HMOBEDA<sub>EXT</sub> is better than the other approaches for all instances regarding ER. Another important observation is that all solutions from the reference set of M5N20K6,

**Table 9**  
Average IGD over 30 executions landscapes.

Algorithm	Average IGD metric					
	HMOBEDA <sub>EXT</sub>	MBN-EDA	NSGA-II	S-MOGLS	NSGA-III	MOEA/D
M2N20K2	0.02191	0.12140	0.12525	0.12884	0.02190	<b>0.01864</b>
M2N20K4	0.02522	0.18211	0.18724	0.18704	0.02696	<b>0.02440</b>
M2N20K6	0.02110	0.12357	0.12049	0.12019	0.01846	<b>0.01790</b>
M2N20K8	0.02384	0.17916	0.17589	0.17343	<b>0.02219</b>	0.02503
M2N20K10	<b>0.01833</b>	0.15700	0.15227	0.15145	0.01869	0.01954
M2N50K2	<b>0.00492</b>	0.05747	0.05892	0.06136	0.00686	0.00912
M2N50K4	<b>0.00767</b>	0.09249	0.09122	0.09582	0.01236	0.01583
M2N50K6	<b>0.00656</b>	0.08513	0.08337	0.08722	0.01237	0.01426
M2N50K8	<b>0.01519</b>	0.13499	0.13000	0.13702	0.02639	0.02816
M2N50K10	<b>0.01006</b>	0.09981	0.09834	0.10278	0.01058	0.01063
M2N100K2	<b>0.00378</b>	0.05643	0.05645	0.05918	0.00855	0.00921
M2N100K4	<b>0.00510</b>	0.07402	0.07357	0.07857	0.01352	0.01560
M2N100K6	<b>0.00903</b>	0.08162	0.07975	0.08383	0.02013	0.02141
M2N100K8	<b>0.01570</b>	0.13131	0.12708	0.13694	0.01680	0.01618
M2N100K10	0.00926	0.08682	0.08100	0.08955	<b>0.00894</b>	0.00900
M3N20K2	0.00775	0.03609	0.03614	0.03841	<b>0.00767</b>	0.00828
M3N20K4	0.00763	0.04474	0.04382	0.04568	0.00789	<b>0.00747</b>
M3N20K6	0.00642	0.04677	0.04576	0.04587	<b>0.00632</b>	0.00647
M3N20K8	0.00720	0.04580	0.04478	0.04538	0.00719	<b>0.00716</b>
M3N20K10	<b>0.00581</b>	0.04074	0.03941	0.03869	0.00576	0.00598
M3N50K2	<b>0.00323</b>	0.02511	0.02368	0.02550	0.00416	0.00456
M3N50K4	<b>0.00310</b>	0.02748	0.02539	0.02814	0.00451	0.00478
M3N50K6	<b>0.00411</b>	0.03678	0.03435	0.03702	0.00704	0.00776
M3N50K8	<b>0.00573</b>	0.03933	0.03685	0.03927	0.00884	0.00895
M3N50K10	<b>0.00326</b>	0.03229	0.03014	0.03280	0.00356	0.00354
M3N100K2	<b>0.00284</b>	0.02864	0.02632	0.02904	0.00522	0.00547
M3N100K4	<b>0.00230</b>	0.03280	0.02958	0.03368	0.00671	0.00793
M3N100K6	<b>0.00253</b>	0.03304	0.02967	0.03351	0.00864	0.00929
M3N100K8	<b>0.01013</b>	0.04524	0.04071	0.04675	0.01250	0.01293
M3N100K10	<b>0.00287</b>	0.02874	0.02523	0.03010	0.00293	0.00292
M5N20K2	<b>0.00457</b>	0.00954	0.00814	0.01012	0.00808	0.01142
M5N20K4	<b>0.00423</b>	0.01049	0.01045	0.01045	0.00831	0.01253
M5N20K6	<b>0.00432</b>	0.01161	0.01100	0.01034	0.00934	0.01388
M5N20K8	<b>0.00436</b>	0.01073	0.01309	0.01001	0.00927	0.01366
M5N20K10	<b>0.00489</b>	0.01208	0.01342	0.01243	0.01045	0.01486
M5N50K2	<b>0.00390</b>	0.01208	0.00902	0.01089	0.00862	0.01383
M5N50K4	<b>0.00334</b>	0.01265	0.01320	0.01386	0.00849	0.01476
M5N50K6	<b>0.00404</b>	0.01441	0.01243	0.01353	0.01008	0.01729
M5N50K8	<b>0.00272</b>	0.00929	0.00902	0.01100	0.00657	0.01140
M5N50K10	<b>0.00283</b>	0.00926	0.00946	0.00825	0.00652	0.01127
M5N100K2	<b>0.00253</b>	0.01239	0.01331	0.01265	0.00821	0.01365
M5N100K4	<b>0.00310</b>	0.01621	0.01573	0.01375	0.01002	0.01768
M5N100K6	<b>0.00380</b>	0.01329	0.01540	0.01232	0.00758	0.01529
M5N100K8	<b>0.00248</b>	0.00843	0.00803	0.00792	0.00496	0.01043
M5N100K10	<b>0.00224</b>	0.00753	0.00781	0.00781	0.00429	0.00897
M8N20K2	<b>0.00409</b>	0.00523	0.00561	0.00572	0.00448	0.00695
M8N20K4	<b>0.00369</b>	0.00478	0.00484	0.00495	0.00414	0.00687
M8N20K6	<b>0.00364</b>	0.00471	0.00517	0.00451	0.00408	0.00717
M8N20K8	<b>0.00359</b>	0.00462	0.00473	0.00495	0.00406	0.00688
M8N20K10	<b>0.00370</b>	0.00466	0.00429	0.00506	0.00414	0.00694
M8N50K2	<b>0.00233</b>	0.00396	0.00319	0.00396	0.00290	0.00543
M8N50K4	<b>0.00230</b>	0.00418	0.00495	0.00440	0.00297	0.00566
M8N50K6	<b>0.00230</b>	0.00397	0.00374	0.00418	0.00295	0.00565
M8N50K8	<b>0.00229</b>	0.00401	0.00407	0.00363	0.00287	0.00547
M8N50K10	<b>0.00222</b>	0.00360	0.00385	0.00385	0.00274	0.00522
M8N100K2	<b>0.00189</b>	0.00439	0.00440	0.00407	0.00256	0.00567
M8N100K4	<b>0.00178</b>	0.00415	0.00374	0.00385	0.00243	0.00538
M8N100K6	<b>0.00166</b>	0.00391	0.00374	0.00396	0.00223	0.00494
M8N100K8	<b>0.00169</b>	0.00375	0.00374	0.00363	0.00224	0.00490
M8N100K10	<b>0.00169</b>	0.00360	0.00418	0.00352	0.00222	0.00480

M5N20K10 and M5N50K2 instances have been generated using HMOBEDA<sub>EXT</sub>, as shown by  $ER = 0$  for this approach on these instances.

Table 11 shows that the run time required by each algorithm for the same instance are similar to each other. In addition, the run times do not present statistically significant differences, except for instances with 2 and 3 objectives, where NSGA-II has the lowest computational time. However, as in MOKP, increasing the number of objectives and variables severely impacts the average computational time of all approaches.

### 5.3. Analyzing the final bayesian network

Aiming to proceed with the exploratory analysis of the PGM structures, Fig. 8 presents three hypothetical examples for  $R = 2$

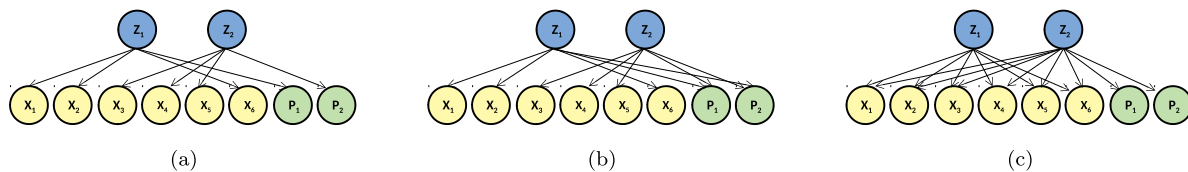
objectives,  $Q = 6$  decision variables and  $L = 2$  LS parameters, illustrating how the analysis of the graphical structure of the BN can unearth specific characteristics of the MOPs.

In the first example, Fig. 8(a), we can observe that from each objective ( $Z_1$  and  $Z_2$ ) there are three related decision variables and each one of the LS parameter ( $P_1$  or  $P_2$ ) depend on a single objective; however in Fig. 8(b), the two LS parameters ( $P_1$  and  $P_2$ ) depend on the two objectives ( $Z_1$  and  $Z_2$ ). Another different scenario can be shown in 8(c), where the six decision variables depend on the two objectives and the LS parameters ( $P_1$  and  $P_2$ ) depend on a single objective ( $Z_2$ ). These graphs analysis can help us to understand about the characteristics of the MOP. The comparison between the graphical structures can be used to characterize or classify the MOPs according to the patterns of interactions between variables, objectives and LS parameters.

**Table 10**

Error Ratio over 30 executions for each algorithm. The best values are in bold.

Instance	Ref	HMOBEDA <sub>EXT</sub>	MBN-EDA	NSGA-II	S-MOGLS	NSGA-III	MOEA/D
M2N20K2	19	<b>0.5000</b>	1.0000	1.0000	1.0000	0.6974	0.8026
M2N20K4	11	<b>0.5000</b>	1.0000	1.0000	1.0000	0.6591	0.8409
M2N20K6	20	<b>0.6000</b>	1.0000	1.0000	1.0000	0.7500	0.6500
M2N20K8	10	<b>0.4000</b>	1.0000	1.0000	1.0000	0.6000	1.0000
M2N20K10	12	<b>0.5000</b>	1.0000	1.0000	1.0000	0.6667	0.8333
M2N50K2	75	<b>0.5067</b>	1.0000	1.0000	1.0000	0.7400	0.7533
M2N50K4	38	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7237	0.7763
M2N50K6	42	<b>0.4762</b>	1.0000	1.0000	1.0000	0.7143	0.8095
M2N50K8	16	<b>0.5000</b>	1.0000	1.0000	1.0000	0.6875	0.8125
M2N50K10	18	<b>0.6111</b>	1.0000	1.0000	1.0000	0.7500	0.6389
M2N100K2	86	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7384	0.7616
M2N100K4	55	<b>0.5455</b>	1.0000	1.0000	1.0000	0.7545	0.7000
M2N100K6	46	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7283	0.7717
M2N100K8	10	<b>0.4000</b>	1.0000	1.0000	1.0000	0.6000	1.0000
M2N100K10	19	<b>0.4211</b>	1.0000	1.0000	1.0000	0.6579	0.9211
M3N20K2	161	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7314	0.7686
M3N20K4	114	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7719	0.7193
M3N20K6	116	<b>0.4914</b>	1.0000	1.0000	1.0000	0.7198	0.7888
M3N20K8	104	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7212	0.7788
M3N20K10	126	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7262	0.7738
M3N50K2	337	<b>0.5104</b>	1.0000	1.0000	1.0000	0.7478	0.7418
M3N50K4	323	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7407	0.7593
M3N50K6	194	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7345	0.7655
M3N50K8	156	<b>0.4936</b>	1.0000	1.0000	1.0000	0.7276	0.7788
M3N50K10	141	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7287	0.7713
M3N100K2	289	<b>0.5052</b>	1.0000	1.0000	1.0000	0.7612	0.7336
M3N100K4	248	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7379	0.7621
M3N100K6	248	<b>0.5040</b>	1.0000	1.0000	1.0000	0.7399	0.7560
M3N100K8	109	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7225	0.7775
M3N100K10	125	<b>0.5000</b>	1.0000	1.0000	1.0000	0.7260	0.7740
M5N20K2	1114	<b>0.0036</b>	1.0000	1.0000	1.0000	1.0000	0.9964
M5N20K4	1115	<b>0.0009</b>	1.0000	1.0000	1.0000	1.0000	0.9991
M5N20K6	911	<b>0.0000</b>	1.0000	1.0000	1.0000	1.0000	1.0000
M5N20K8	888	<b>0.0034</b>	1.0000	1.0000	1.0000	1.0000	0.9966
M5N20K10	703	<b>0.0000</b>	1.0000	1.0000	1.0000	1.0000	1.0000
M5N50K2	815	<b>0.0000</b>	1.0000	1.0000	1.0000	1.0000	1.0000
M5N50K4	849	<b>0.0047</b>	1.0000	1.0000	1.0000	1.0000	0.9953
M5N50K6	559	<b>0.0197</b>	1.0000	1.0000	1.0000	1.0000	0.9803
M5N50K8	943	<b>0.0286</b>	1.0000	1.0000	1.0000	1.0000	0.9714
M5N50K10	876	<b>0.0308</b>	1.0000	1.0000	1.0000	1.0000	0.9692
M5N100K2	842	<b>0.0036</b>	1.0000	1.0000	1.0000	1.0000	0.9964
M5N100K4	592	<b>0.0186</b>	1.0000	1.0000	1.0000	1.0000	0.9814
M5N100K6	539	<b>0.1466</b>	1.0000	1.0000	1.0000	1.0000	0.8534
M5N100K8	925	<b>0.1286</b>	1.0000	1.0000	1.0000	1.0000	0.8714
M5N100K10	1100	<b>0.1382</b>	1.0000	1.0000	1.0000	1.0000	0.8618
M8N20K2	3096	<b>0.1395</b>	1.0000	1.0000	1.0000	0.9984	0.8614
M8N20K4	3611	<b>0.1399</b>	1.0000	1.0000	1.0000	0.9958	0.8643
M8N20K6	3525	<b>0.1245</b>	1.0000	1.0000	1.0000	0.9983	0.8772
M8N20K8	3445	<b>0.1170</b>	1.0000	1.0000	1.0000	0.9942	0.8888
M8N20K10	3233	<b>0.1222</b>	1.0000	1.0000	1.0000	0.9954	0.8825
M8N50K2	4077	<b>0.1921</b>	1.0000	1.0000	1.0000	1.0000	0.8079
M8N50K4	3980	<b>0.1937</b>	1.0000	1.0000	1.0000	1.0000	0.8063
M8N50K6	3888	<b>0.2063</b>	1.0000	1.0000	1.0000	1.0000	0.7937
M8N50K8	3797	<b>0.2054</b>	1.0000	1.0000	1.0000	1.0000	0.7946
M8N50K10	3970	<b>0.2000</b>	1.0000	1.0000	1.0000	0.9997	0.8003
M8N100K2	3249	<b>0.1847</b>	1.0000	1.0000	1.0000	1.0000	0.8153
M8N100K4	3728	<b>0.2154</b>	1.0000	1.0000	1.0000	1.0000	0.7846
M8N100K6	4133	<b>0.2354</b>	1.0000	1.0000	1.0000	1.0000	0.7646
M8N100K8	4011	<b>0.2468</b>	1.0000	1.0000	1.0000	1.0000	0.7532
M8N100K10	3906	<b>0.2348</b>	1.0000	1.0000	1.0000	1.0000	0.7652

**Fig. 8.** Three different situations of the relationships that can be captured by the BN for  $R = 2$  objectives,  $Q = 6$  decision variables and  $L = 2$  LS parameters.



**Table 11**

Average run time (min) for each algorithm and instance.

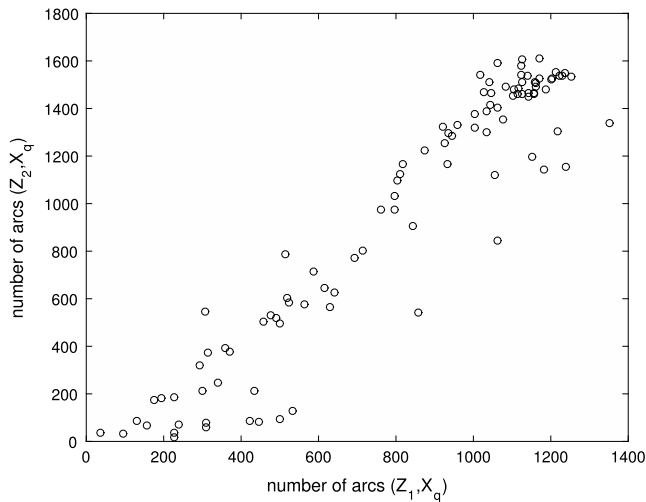
Algorithm	Average run time (minutes)					
	HMOBEDA <sub>EXT</sub>	MBN-EDA	NSGA-II	S-MOGLS	NSGA-III	MOEA/D
M2N20K2	0.59	1.17	<b>0.51</b>	0.89	1.06	0.92
M2N20K4	0.68	1.09	<b>0.54</b>	0.84	0.97	0.80
M2N20K6	0.96	0.90	<b>0.47</b>	0.91	0.95	0.88
M2N20K8	0.81	0.77	<b>0.56</b>	0.78	0.83	0.87
M2N20K10	0.86	0.67	<b>0.52</b>	0.64	0.87	1.02
M2N50K2	1.82	1.82	<b>0.85</b>	1.69	1.78	1.83
M2N50K4	1.84	1.86	<b>0.95</b>	1.90	1.80	1.90
M2N50K6	1.89	1.72	<b>0.81</b>	2.03	1.86	1.76
M2N50K8	1.60	1.69	<b>0.88</b>	1.75	1.97	1.86
M2N50K10	1.72	1.85	<b>0.92</b>	1.73	1.88	2.04
M2N100K2	2.69	2.60	<b>1.31</b>	2.67	2.74	2.75
M2N100K4	2.71	2.66	<b>1.46</b>	2.72	2.71	2.59
M2N100K6	2.59	2.54	<b>1.34</b>	2.58	2.62	2.82
M2N100K8	2.89	2.70	<b>1.32</b>	2.71	2.67	2.71
M2N100K10	2.70	2.75	<b>1.50</b>	2.62	2.69	2.48
M3N20K2	1.15	1.35	<b>0.69</b>	1.35	1.43	1.41
M3N20K4	1.23	1.38	<b>0.73</b>	1.18	1.58	1.42
M3N20K6	1.30	1.49	<b>0.69</b>	1.09	1.43	1.37
M3N20K8	1.25	1.22	<b>0.65</b>	1.67	1.19	1.30
M3N20K10	1.23	1.46	<b>0.70</b>	1.32	1.41	1.36
M3N50K2	2.45	2.95	<b>1.45</b>	2.76	2.66	2.83
M3N50K4	2.64	2.66	<b>1.45</b>	2.84	2.78	2.81
M3N50K6	2.83	2.56	<b>1.35</b>	2.83	2.67	2.59
M3N50K8	2.80	2.67	<b>1.41</b>	2.74	2.46	2.63
M3N50K10	3.01	2.63	<b>1.33</b>	2.61	2.38	2.73
M3N100K2	4.59	4.37	<b>2.30</b>	4.45	4.26	4.17
M3N100K4	4.38	4.65	<b>2.30</b>	4.39	4.47	4.46
M3N100K6	4.35	4.29	<b>2.18</b>	4.46	4.60	4.65
M3N100K8	4.49	4.45	<b>2.28</b>	4.98	4.42	4.52
M3N100K10	4.47	4.30	<b>2.29</b>	4.96	4.57	4.51
M5N20K2	<b>4.38</b>	4.52	4.51	4.62	4.47	4.56
M5N20K4	4.23	4.28	4.39	4.84	4.43	<b>4.10</b>
M5N20K6	4.89	4.72	5.01	<b>4.01</b>	4.44	4.55
M5N20K8	<b>4.22</b>	4.58	4.56	4.62	4.25	4.77
M5N20K10	5.02	4.43	5.29	4.64	<b>4.41</b>	4.47
M5N50K2	9.05	8.91	9.19	9.18	<b>8.51</b>	8.97
M5N50K4	<b>8.38</b>	9.14	9.55	8.98	8.93	8.53
M5N50K6	9.49	8.96	9.05	9.40	8.82	<b>8.56</b>
M5N50K8	9.37	8.83	8.91	<b>8.59</b>	8.79	9.30
M5N50K10	8.83	8.87	8.86	9.04	9.01	<b>8.78</b>
M5N100K2	18.34	17.90	<b>17.69</b>	18.33	18.29	18.20
M5N100K4	18.17	17.62	17.94	18.23	18.31	<b>17.60</b>
M5N100K6	18.16	17.83	17.88	<b>17.63</b>	18.22	18.41
M5N100K8	17.58	17.79	18.15	17.83	<b>17.35</b>	17.74
M5N100K10	17.72	<b>17.66</b>	17.71	17.82	17.84	18.01
M8N20K2	8.04	8.40	8.05	8.11	<b>7.97</b>	8.35
M8N20K4	<b>7.86</b>	8.20	8.19	8.13	8.04	8.19
M8N20K6	<b>8.00</b>	8.08	8.36	8.01	8.12	8.03
M8N20K8	8.25	8.14	8.54	8.22	8.13	<b>8.08</b>
M8N20K10	8.27	8.11	8.15	<b>7.96</b>	8.07	8.06
M8N50K2	13.54	<b>13.29</b>	13.62	13.71	13.39	13.84
M8N50K4	13.63	13.36	13.72	13.42	13.40	<b>13.14</b>
M8N50K6	13.56	<b>13.27</b>	13.84	13.40	13.60	13.57
M8N50K8	13.61	13.42	13.51	<b>13.35</b>	13.63	13.50
M8N50K10	13.48	<b>13.32</b>	13.53	13.46	13.90	13.38
M8N100K2	27.03	<b>26.67</b>	27.18	27.09	27.15	26.94
M8N100K4	<b>26.94</b>	27.09	27.29	27.08	27.06	27.02
M8N100K6	27.21	27.32	26.95	27.02	27.00	<b>26.69</b>
M8N100K8	27.22	27.36	27.00	<b>26.95</b>	27.07	27.25
M8N100K10	26.91	26.92	27.08	<b>26.85</b>	27.18	27.09

In this section we provide an interpretation of the probabilistic model learned during the evolutionary process for some MOKP and MNK-landscape instances (examples of easy bi-objective and difficult many objective optimization instances) achieved by the HMOBEDA<sub>EXT</sub> version.

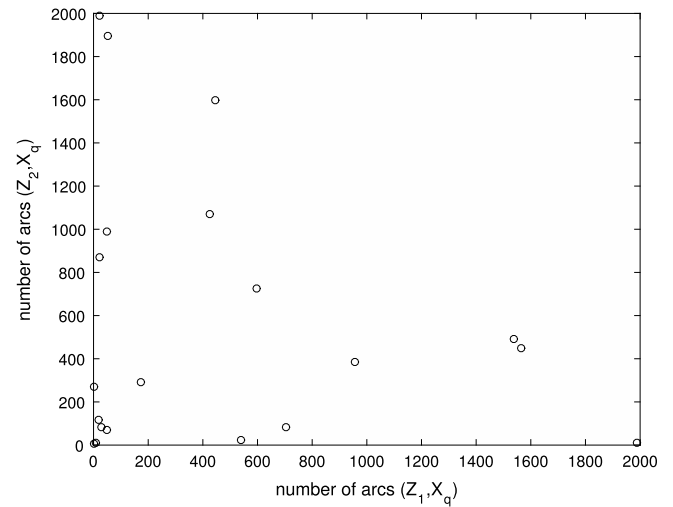
The arcs captured by the BN model represent the interactions learned from the data from the beginning until the end of the evolutionary process. For instance, an arc between objective  $z_1$  and variable  $x_1$  can indicate that the relationship between them is stronger than the relationship between objective  $z_1$  and another variable  $x_2$  that is not linked to  $z_1$  in the network. Similarly, arcs between objectives and parameters of the local optimizers reveal the existence of relationships between the behavior of local optimizers and the changes in the objectives. By computing the frequencies of the arcs learned by the Bayesian Networks we can determine

which are the most stable interactions and extract other useful information about the particular effect of the objectives on the different local optimizers.

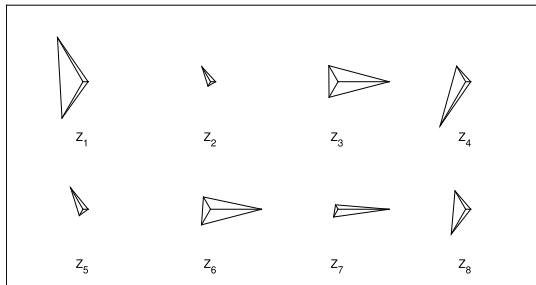
Fig. 9 shows the interactions (circles) between each decision variable  $X_q$ ,  $q \in \{1, \dots, 100\}$ , and the objectives  $Z_1$  and  $Z_2$ , learned by the BN for the MOKP instance 2–100. Each circle has coordinates indicating the number of times an arc  $(Z_1, X_q)$  has been captured along the evolutionary process for all executions versus a similar measure for arc  $(Z_2, X_q)$ . Note that the interaction is quite similar for the two objectives, since most of the points are located nearby the +1 slope line. In other words, based on Fig. 9, we can conclude that variables (especially if the number of interactions for a given variable is either very low or very high) are equally affected by both objectives.



**Fig. 9.** For instance 2–100, number of times arc  $(Z_1, X_q)$  has been captured in the BN versus a similar measure for arc  $(Z_2, X_q)$ . Each circle corresponds to one decision variable  $X_q, q \in \{1, \dots, 100\}$ .



**Fig. 11.** For M2N20K2 instance, number of times arc  $(Z_1, X_q)$  has been captured in the BN versus a similar measure for arc  $(Z_2, X_q)$ . Each circle corresponds to one decision variable  $X_q, q \in \{1, \dots, 20\}$ .



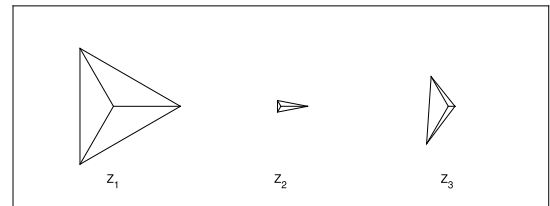
**Fig. 10.** Glyph representation of the three LS parameters (spokes) for each objective  $Z_1$  to  $Z_8$  of instance 8–100.

Fig. 10 focuses on the analysis of BN structure concerning the relations between objectives and LS parameters for the MOKP instance 8–100. The relations between each objective and the three LS parameters considered in this paper are illustrated by star glyphs. In such representation, each spoke represents one parameter  $P_l$  and it is proportional to the number of times the arc  $(Z_r, P_l), l \in \{1, 2, 3\}, r \in \{1, \dots, 8\}$ , has been captured along the evolutionary process for all executions. The glyphs allow us to visualize which is the relative strength of the relations. For example, it can be seen in Fig. 10 that objectives  $Z_2$  and  $Z_5$  have small influence on the way the parameters are instantiated. On the other hand,  $Z_1, Z_3$  and  $Z_6$  have great influence, although  $Z_3$  and  $Z_6$  seem to have similar balance among the parameters.

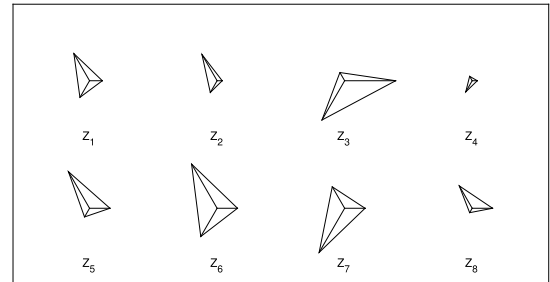
We extend this analysis for the MNK-landscape problem. Fig. 11 shows the interactions between each decision variable  $X_q, q \in \{1, \dots, 20\}$ , and the objectives  $Z_1$  and  $Z_2$ , learned by the BN for the M2N20K2 instance. The analysis of Fig. 11 reveals that different from 2–100 MOKP, for the M2N20K2 instance of the MNK landscape problem, variables are differently affected by each objective. Notice that some variables are almost exclusively affected by objective  $Z_1$  and others by objective  $Z_2$  (as in the extreme points depicted in Fig. 11 for example).

Figs. 12 and 13 present the relations between objectives and LS parameters for M3N50K6 and M8N100K10 instances, respectively.

We can observe, in Fig. 12, that objective  $Z_1$  has great influence on the parameters and it is better balanced among them. From Fig. 13 we can conclude that  $Z_4$  has the least influence on the parameters.



**Fig. 12.** Glyph representation of the three LS parameters (spokes) for each objective  $Z_1$  to  $Z_3$  of M3N50K6 instance.



**Fig. 13.** Glyph representation of the three LS parameters (spokes) for each objective  $Z_1$  to  $Z_8$  of M8N100K10 instance.

We have shown with these examples that the HMOBEDA<sub>EXT</sub> approach allows a step forward. First, it is possible to estimate the relation between variables and objectives from the analysis of how frequent objective-variable interactions are. Second, it is possible to determine how strong the interaction between objective and LS parameter is from the analysis of objective-parameters interactions in the PGM along the evolution.

## 6. Conclusion

In this paper we have explored, from a probabilistic point of view, the approximated Pareto-front, using the final PGM structure, for a hybrid EDA named Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA). We have investigated the influence of using different types of evidence in the behavior of this algorithm. For this, we have extended the original

HMOBEDA to build two other versions with different sample techniques: HMOBEDA<sub>CPT</sub>, which considers *priori* probabilities for the objectives nodes available in the PGM conditional probability table (CPT); and HMOBEDA<sub>EXT</sub>, with extreme or ideal points considered as evidences values.

These versions are considered in the context of multi and many-objective combinatorial optimization in order to analyze how using evidences in the sampling process during the evolution can influence both convergence and diversity along the approximated Pareto Front. This analysis can be useful from a general point of view, defining which strategy is better than other. In addition, we aimed to explore one of the main advantages of using EDA: the possibility of analyzing its probabilistic model which in this work encompasses relationship among objectives, decision variables and configuration parameters of the local search procedure.

We have evaluated the quality of the final Pareto Front approximations of HMOBEDA, HMOBEDA<sub>CPT</sub> and HMOBEDA<sub>EXT</sub> through the analysis of the final achieved PGM. The Pareto Front as well as the probability of each solution captured by the PGM for all executions have been shown in a single plot. For more than two objectives, the Euclidean distance from each solution to the ideal point has been shown. It is important to point out that this is an important contribution research since it enables to join in a single graphic average information of PFs resulting from its different executions.

In addition, we have analyzed the performance of HMOBEDA versions considering HV<sup>−</sup> and IGD, concluding that uniformly distributing the evidences among ideal and extreme points of the Pareto Front (HMOBEDA<sub>EXT</sub>) in the sampling process is beneficial for HMOBEDA. Further we compared HMOBEDA<sub>EXT</sub> with other techniques on the two addressed problems: MOKP and MNK-landscape. We concluded that, for the instances considered in this work, HMOBEDA<sub>EXT</sub> is a competitive approach for small number of objectives and outperformed the others when the number of objectives increases. In addition, the average run times for all algorithms have been kept in practical levels. These conclusions are based on the statistical analysis made on hypervolume indicator, IGD and a capacity (ER) metrics. Finally, an analysis of the resulting BN structures has been presented in order to evaluate how the interactions among variables, objectives and local search parameters are captured by this type of PGM. Aiming to illustrate the types of information that can be extracted from the models, we have shown that the frequency of arcs in the BNs can indicate how the variables and local search parameters are influenced by the objectives.

Based on the results we concluded that probabilistic modeling arises as a sensitive and feasible way to learn and explore dependencies between variables and objectives and it can be also used for controlling the application of local search operators, as in a parameter tuning approach. Another important conclusion is that HMOBEDA<sub>EXT</sub> is less sensitive to the increasing number of objectives, figuring as a good candidate for many objective optimization.

In the future, MOEA techniques other than Pareto-based approaches should be investigated, such as aggregative/decomposition based approaches, for example. These new approaches will be investigated with more than eight objectives and applied to other problems. Additionally, an interesting research direction is the use of other types of PGM that can learn and explore dependencies between variables, objectives and automatically control the application of local search operators. Moreover, the BN used to model parameters of the local search, can be extended to model other parameters (especially those associated with individual solutions) as well as those relevant to other search strategies.

## Acknowledgments

M. Delgado acknowledges CNPq, Brazil grant 309935/2017-2. M. Martins acknowledges CAPES/Brazil. R. Santana acknowledges

support from the IT-609-13 program (Basque Government, Spain) and TIN2016-78365-R (Spanish Ministry of Economy, Industry and Competitiveness, Spain).

## References

- [1] M. Luque, Modified interactive Chebyshev algorithm (MICA) for non-convex multiobjective programming, *Optim. Lett.* 9 (1) (2015) 173–187.
- [2] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley and Sons, New York, 2001.
- [3] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: A short review, in: IEEE Conference on Evolutionary Computation, in: CEC'2008, IEEE, Piscataway, New Jersey, 2008, pp. 2419–2426, <http://dx.doi.org/10.1109/CEC.2008.4631121>.
- [4] A. Lara, G. Sanchez, C.A.C. Coello, O. Schutze, HCS: A new local search strategy for memetic multiobjective evolutionary algorithms, *IEEE Trans. Evol. Comput.* 14 (1) (2010) 112–132.
- [5] A. Zhou, B.Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, *Swarm Evol. Comput.* (1) (2011) 32–49.
- [6] A. Zhou, J. Sun, Q. Zhang, An estimation of distribution algorithm with cheap and expensive local search methods, *IEEE Trans. Evol. Comput.* 19 (6) (2015) 807–822.
- [7] M.S. Martins, M.R. Delgado, R. Santana, R. Lüders, R.A. Gonçalves, C.P.d. Almeida, HMOBEDA: Hybrid multi-objective bayesian estimation of distribution algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'16, ACM, New York, NY, USA, 2016, pp. 357–364.
- [8] M.S.R. Martins, M.R.B.S. Delgado, R. Lüders, R. Santana, R.A. Gonçalves, C.P. de Almeida, Hybrid multi-objective Bayesian estimation of distribution algorithm: a comparative analysis for the multi-objective knapsack problem, *J. Heuristics* (2017) 1–23.
- [9] H. Mühlenbein, G. Paab, From recombination of genes to the estimation of distributions I. Binary parameters, in: Parallel Problem Solving from Nature, in: PPSN IV - Lecture Notes in Computer Science, vol. 1411, Springer-Verlag, London, UK, 1996, pp. 178–187.
- [10] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, vol. 2, Springer, Netherlands, 2002.
- [11] H. Karshenas, R. Santana, C. Bielza, P. Larrañaga, Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables, *IEEE Trans. Evol. Comput.* 18 (2014) 519–542.
- [12] M. Laumanns, J. Ocenasek, Bayesian optimization algorithms for multi-objective optimization, in: Parallel Problem Solving from Nature, in: PPSN VII - Lecture Notes in Computer Science, vol. 2439, Springer-Verlag, London, UK, 2002, pp. 298–307.
- [13] P. Larrañaga, H. Karshenas, C. Bielza, R. Santana, A review on probabilistic graphical models in evolutionary computation, *J. Heuristics* 18 (2012) 795–819.
- [14] H. Li, Q. Zhang, E. Tsang, J.A. Ford, Hybrid estimation of distribution algorithm for multiobjective Knapsack problem, *Evol. Comput. Combin. Opt.* (2004) 145–154.
- [15] L. Wang, S. Wang, Y. Xu, An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem, *Expert Syst. Appl.* 39 (2012) 5593–5599.
- [16] M.S. Martins, M.R. Delgado, R. Lüders, R. Santana, Ricardo, Gonçalves, C.P.d. Almeida, 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, 2017, pp. 384–389.
- [17] E. Zitzler, L. Thiele, Multiple objective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Trans. Evol. Comput.* 3 (1999) 257–271.
- [18] J.M. Bader, Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods (Ph.D. thesis), ETH Zurich, Zurich, 2009.
- [19] D.A. D. A. van Veldhuizen, G.B. Lamont, Multiobjective evolutionary algorithm test suites, in: Proceedings of the 1999 ACM Symposium on Applied Computing, SAC'99, ACM, New York, NY, USA, 1999, pp. 351–357.
- [20] H. Ishibuchi, N. Akedo, Y. Nojima, Behavior of multiobjective evolutionary algorithms on many-objective Knapsack problems, *IEEE Trans. Evol. Comput.* 19 (2) (2015) 264–283.
- [21] Y. Tanigaki, K. Narukawa, Y. Nojima, H. Ishibuchi, Preference-based NSGA-II for many-objective Knapsack problems, in: 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and Advanced Intelligent Systems (ISIS), IEEE, Kita-Kyushu, Japan, 2014, pp. 637–642.
- [22] H.E. Aguirre, K. Tanaka, Insights on properties of multiobjective MNK-landscapes, in: Proceedings of the 2004 Congress on Evolutionary Computation, vol. 1, IEEE, Portland, OR, USA, 2004, pp. 196–203.
- [23] H.E. Aguirre, K. Tanaka, Working principles, behavior, and performance of MOEAs on MNK-landscapes, *European J. Oper. Res.* 181 (3) (2007) 1670–1690.

- [24] S. Verel, A. Liefvooghe, L. Jourdan, C. Dhaenens, Pareto local optima of multi-objective NK-landscapes with correlated objectives, *Evol. Comput. Combin. Opt.* (2011) 226–237.
- [25] R. Santana, A. Mendiburu, J.A. Lozano, Multi-objective NM-landscapes, in: *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO'15*, ACM, Orlando, Florida, USA, 2015, pp. 1477–1478.
- [26] F. Daolio, A. Liefvooghe, S. Verel, H. Aguirre, K. Tanaka, Global vs local search on multi-objective NK-landscapes: contrasting the impact of problem features, in: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO'15*, ACM, 2015, pp. 369–376.
- [27] M. Pelikan, Analysis of estimation of distribution algorithms and genetic algorithms on NK landscapes, in: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO'08*, ACM, Atlanta, Georgia, 2008, pp. 1033–1040.
- [28] M. Pelikan, K. Sastry, D.E. Goldberg, M.V. Butz, M. Hauschild, Performance of evolutionary algorithms on NK landscapes with nearest neighbor interactions and tunable overlap, in: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO'09*, ACM, Montreal, Canada, 2009, pp. 851–858.
- [29] R.-T. Liaw, C.-K. Ting, Effect of model complexity for estimation of distribution algorithm in NK landscapes, in: *Symposium on Foundations of Computational Intelligence, FOCI'13*, IEEE, 2013, pp. 76–83.
- [30] M. Rosenblatt, Z. Sinuany-Stern, Generating the discrete efficient frontier to the capital budgeting problem, *Oper. Res.* 37 (1989) 384–394.
- [31] J. Teng, G. Tzeng, A multiobjective programming approach for selecting non-independent transportation investment alternatives, *Transp. Res. B* 30 (1996) 201–207.
- [32] M. Kostreva, W. Ogryczak, D. Tonkyn, Relocation problems arising in conservation biology, *Comput. Math. Appl.* 37 (1999) 135–150.
- [33] L. Jenkins, A bicriteria knapsack program for planning remediation of contaminated lightstation sites, *European J. Oper. Res.* 140 (2) (2002) 427–433.
- [34] B. Chabane, M. Basseur, J.-K. Hao, A practical case of the multiobjective knapsack problem: Design, modelling, tests and analysis, in: *International Conference on Learning and Intelligent Optimization*, Springer, Lille, France, 2015, pp. 249–255.
- [35] S. Kauffman, S. Levin, Towards a general theory of adaptive walks on rugged landscapes, *J. Theoret. Biol.* 128 (1) (1987) 11–45.
- [36] S.A. Kauffman, E.D. Weinberger, The NK model of rugged fitness landscapes and its application to maturation of the immune response, *J. Theoret. Biol.* 141 (2) (1989) 211–245.
- [37] M. Valente, An NK-like model for complexity, *J. Evol. Econ.* 24 (1) (2014) 107–134.
- [38] Y.-Y. Tan, Y.-C. Jiao, MOEA/D with Uniform Design for Solving Multiobjective Knapsack Problems, *J. Comput.* 8 (2013) 302–307.
- [39] A. Jaskiewicz, Comparison of local search-based metaheuristics on the multiple-objective knapsack problem, *Found. Comput. Decis. Sci.* 26 (2001) 99–120.
- [40] A. Jaskiewicz, On the performance of multiple objective genetic local search on the 0/1 knapsack problem: a comparative experiment, *IEEE Trans. Evol. Comput.* 6 (2002) 402–412.
- [41] D.S. Vianna, M. de Ftima Dianin Vianna, Local search-based heuristics for the multiobjective multidimensional knapsack problem, *Produo* 23 (2013) 478–487.
- [42] T. Lust, J. Teghem, The multiobjective multidimensional knapsack problem: a survey and a new approach, *Int. Trans. Oper. Res.* 19 (2012) 495–520.
- [43] X. Gandibleux, A. Frville, Tabu search based procedure for solving the 0-1 multiobjective knapsack problem: The two objectives case, *J. Heuristics* 6 (2000) 361–383.
- [44] S.A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, USA, 1993.
- [45] R. Santana, A. Mendiburu, J.A. Lozano, Evolving MNK-landscapes with structural constraints, in: *IEEE Congress on Evolutionary Computation, CEC'15*, IEEE, Sendai, Japan, 2015, pp. 1364–1371.
- [46] J. Schwarz, J. Ocenasek, Pareto Bayesian optimization algorithm for the multiobjective 0/1 Knapsack problem, in: *7th International Mendel Conference on Soft Computing, MENDEL'2001*, IEEE, Piscataway, New Jersey, 2001, pp. 131–136.
- [47] S. Gupta, D. Batra, P. Verma, Greedy estimation of distributed algorithm to solve bounded knapsack problem, *Int. J. Comput. Sci. Inf. Technol.* 5 (2014) 4313–4316.
- [48] A. Liefvooghe, S. Verel, F. Daolio, H. Aguirre, K. Tanaka, A feature-based performance analysis in evolutionary multiobjective optimization, in: *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, Guimaraes, Portugal, 2015, pp. 95–109.
- [49] D. Koller, N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, The MIT Press, Cambridge, MA, 2009.
- [50] G. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, *Mach. Learn.* 9 (4) (1992) 309–347.
- [51] K.B. Korb, A.E. Nicholson, *Bayesian Artificial Intelligence*, second ed., CRC Press, Boca Raton, Florida, 2010.
- [52] E. Bengoetxea, *Inexact Graph Matching Using Estimation of Distribution Algorithms* (Ph.D. thesis), University of the Basque Country, Basque Country, 2002.
- [53] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA, 1988.
- [54] M. Henrion, Propagating uncertainty in Bayesian networks by probabilistic logic sampling, in: *Machine Intelligence and Pattern Recognition*, vol. 5, Elsevier, 1988, pp. 149–163.
- [55] C. Yuan, B. Malone, Learning optimal Bayesian networks: A shortest path perspective, *J. Artificial Intelligence Res.* 48 (1) (2013) 23–65.
- [56] I. Tsamardinos, C.F. Aliferis, A.R. Statnikov, E. Statnikov, Algorithms for large scale Markov blanket discovery, in: *FLAIRS Conference*, vol. 2, AAAI Press, St. Augustine, Florida, USA, 2003, pp. 376–380.
- [57] C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, X.D. Koutsoukos, Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: algorithms and empirical evaluation, *J. Mach. Learn. Res.* 11 (Jan) (2010) 171–234.
- [58] I. Tsamardinos, L.E. Brown, C.F. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, *Mach. Learn.* 65 (1) (2006) 31–78.
- [59] L. van der Gaag, P. de Waal, Multi-dimensional Bayesian network classifiers, in: *Proceedings of the Third European Workshop in Probabilistic Graphical Models, Action M Agency*, Prague, Czech Republic, 2006, pp. 107–114.
- [60] P.R. de Waal, L.C. van der Gaag, Inference and learning in multi-dimensional Bayesian network classifiers, in: *Proceedings of the 9th European Conference in Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'07*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 501–511, [http://dx.doi.org/10.1007/978-3-540-75256-1\\_45](http://dx.doi.org/10.1007/978-3-540-75256-1_45).
- [61] S. Moran, Y. He, K. Liu, Choosing the best Bayesian classifier: An empirical study, *IAENG Int. J. Comput. Sci.* 36 (4) (2009) 322–331.
- [62] H. Bouhamed, A. Masmoudi, T. Lecroq, A. Rebaï, A new approach for Bayesian classifier learning structure via K2 algorithm, in: *Proceedings of the 8th International Conference in Emerging Intelligent Computing Technology and Applications, ICIC'12*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 387–393, [http://dx.doi.org/10.1007/978-3-642-31837-5\\_56](http://dx.doi.org/10.1007/978-3-642-31837-5_56).
- [63] J.A. Carta, S. Velázquez, J. Matías, Use of Bayesian networks classifiers for long-term mean wind turbine energy output estimation at a potential wind energy conversion site, *Energy Convers. Manage.* 52 (2) (2011) 1137–1149.
- [64] P. Spirtes, C.N. Glymour, R. Scheines, *Causation Prediction, and Search*, MIT press, USA, 2000.
- [65] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, fourth ed., Morgan Kaufmann, 2016.
- [66] S.J. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, second ed., Prentice Hall, Upper Saddle River, New Jersey, USA, 2003.
- [67] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, *Evol. Comput.* 2 (1994) 221–248.
- [68] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2002) 182–197.
- [69] M.S.R. Martins, *A Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm* (Ph.D. thesis), Federal University of Technology - Paraná, Curitiba, 2017.
- [70] S. Jiang, Y.S. Ong, J. Zhang, L. Feng, Consistencies and contradictions of performance metrics in multiobjective optimization, *IEEE Trans. Cybern.* 44 (12) (2014) 2391–2404.
- [71] S. Bleuler, M. Laumanns, L. Thiele, E. Zitzler, PISA - A platform and programming language independent interface for search algorithms, in: *Evolutionary Multi-Criterion Optimization, EMO'2003*, Springer, Berlin, 2003, pp. 494–508.
- [72] W. Conover, *Practical Nonparametric Statistics*, third ed., Wiley, New York, 1999.
- [73] G. Casella, R.L. Berger, *Statistical Inference*, second ed., Duxbury, Pacific Grove, CA, 2002.
- [74] H. Ishibuchi, Y. Hitotsuyanagi, Y. Nojima, Scalability of multiobjective genetic local search to many-objective problems: Knapsack problem case studies, in: *IEEE Conference on Evolutionary Computation, CEC'2008*, IEEE, Piscataway, New Jersey, 2008, pp. 3586–3593.
- [75] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11 (6) (2007) 712–731.
- [76] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints, *IEEE Trans. Evol. Comput.* 18 (4) (2014) 577–601.