



HMOBEDA: Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm

Marcella S. R. Martins
Federal University of
Technology - Paraná (UTFPR)
Av. Sete de Setembro, 3165
Curitiba PR, Brazil
marcella@utfpr.com.br

Myriam R. B. S. Delgado
Federal University of
Technology - Paraná (UTFPR)
Av. Sete de Setembro, 3165
Curitiba PR, Brazil
myriamdelg@utfpr.edu.br

Roberto Santana
University of the Basque
Country (UPV/EHU)
20018 Donostia
San Sebastián, Spain
roberto.santana@ehu.es

Ricardo Lüders
Federal University of
Technology - Paraná (UTFPR)
Av. Sete de Setembro, 3165
Curitiba PR, Brazil
luders@utfpr.edu.br

Richard Aderbal
Gonçalves
Midwest State University of
Parana (UNICENTRO)
Guarapuava, PR, Brazil
richard@unicentro.br

Carolina Paula de
Almeida
Midwest State University of
Parana (UNICENTRO)
Guarapuava, PR, Brazil
carol@unicentro.br

ABSTRACT

Probabilistic modeling of selected solutions and incorporation of local search methods are approaches that can notably improve the results of multi-objective evolutionary algorithms (MOEAs). In the past, these approaches have been jointly applied to multi-objective problems (MOPs) with excellent results. In this paper, we introduce for the first time a joint probabilistic modeling of (1) local search methods with (2) decision variables and (3) the objectives in a framework named HMOBEDA. The proposed approach is compared with six evolutionary methods (including a modified version of NSGA-III, adapted to solve combinatorial optimization) on instances of the multi-objective knapsack problem with 3, 4, and 5 objectives. Results show that HMOBEDA is a competitive approach. It outperforms the other methods according to the hypervolume indicator.

1. INTRODUCTION

Nowadays, a number of metaheuristics have been developed for efficiently solving multi-objective optimization problems. Among these techniques, evolutionary algorithms are particularly popular, due to their promise of using a population to move towards an entire set of good solutions in a single run. However, for many combinatorial problems, multi-objective optimization evolutionary algorithms (MOEAs), by themselves, are not able to successfully find good solutions. Two strategies have shown to be particularly suited to improve results from MOEAs: local optimizers and probabilistic modeling to capture and exploit the potential regularities that arise in the promising solutions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '16, July 20-24, 2016, Denver, CO, USA

© 2016 ACM. ISBN 978-1-4503-4206-3/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2908812.2908826>

Hybrid algorithms that combine traditional evolutionary operators with local search (LS) procedures have attracted considerable attention from the MOEA community [19]. As pointed out by [9], a number of researchers have acknowledged that these hybrid approaches can often achieve good performance for many MOPs. However, they still present challenges regarding local optimizers, like the choice of the right parameters (e.g., the type, frequency and intensity of LS applied on a specific candidate solution).

Frequency and intensity directly define the degree of evolution (exploration) versus exploitation (performed by LS) in these hybrid approaches [20]. Clearly, a more intense procedure provides greater chances of convergence to the local optima but limits the amount of exploration that could be performed without incurring in an excessive computational cost. Therefore, for a fixed computational budget, care should be taken when setting these two parameters. Moreover, when only a subset of individuals undergo local optimization, the issue of which will be chosen must also be considered. Finally, the type of LS performed favors different neighborhood structures. All previous discussed parameters greatly affect the algorithm's performance.

The other strategy is probabilistic modelling, widely used in Estimation of distribution algorithms (EDAs) [24] a computational approach that, instead of applying genetic operators, learn a probabilistic model from a set of promising solutions, and sample new candidate solutions from the model. The probabilistic model can capture relevant statistics about problem variables and important dependencies existing among these variables.

EDAs developed to solve multi-objective problems [16] are usually called Multi-objective Estimation of Distribution Algorithm (MOEDAs). Although in MOEDAs the probabilistic model generally represents only the relationships between decision variables, the work proposed in [16] shows the effectiveness of also modelling objectives.

At the same time, hybrid approaches have been developed by integrating LS methods as part of a MOEDA [23, 43]. Although hybrid MOEDAs which incorporate LS methods represent a powerful class of algorithms to deal with MOPs, they inherit from hybrid MOEAs the challenge of finding out

an optimal setting of LS parameters. This question is particularly relevant because at different phases of the search the requirements for the application of LS can be very different. Furthermore, in an ideal setting we would tune the parameters of the local optimizer according to the characteristics of the solution to which the local optimizer is applied (e.g. the current values of the different objectives provided by those variables). One alternative to deal with this question is to encode the LS parameters as part of the representation of problem solution as done in previous EAs [1, 39].

In this paper we propose a different approach aiming to benefit from a probabilistic modeling-based approach. The proposal is a joint probabilistic modeling of decision variables, objectives, and parameters of the local optimizer. The rationale is that the probabilistic model could learn which is the appropriate choice of the local optimization parameters for different configurations of decision variables and objective values.

The introduced approach is evaluated on a combinatorial MOP - the multi-objective version of the well known knapsack problem named Multi-objective Knapsack Problem (MOKP). The performance of MOEAs for this problem has been extensively investigated [12, 37, 44]. In particular, MOEDAs that use different types of probabilistic models have already been applied to MOKP. In [23], a hybrid MOEDA based on an univariate probabilistic model is used to solve the 0/1 MOKP. In [39], a hybrid EDA that uses an adaptive local search (HEDA) is proposed to enhance the exploitation ability of the EDA to solve the multidimensional knapsack problem. MOEDAs based on Bayesian networks have also been applied to MOKP in [22, 32].

Our work is built on previous results that investigate the application of LS to MOEDAs [30, 42]. It is also linked to the work presented in [16] in which a joint probabilistic model of objectives and variables is proposed, and related to [23] by considering a LS based on the weighted sum method. However, none of these previous works consider a joint probabilistic model of objectives, variables and LS parameters.

This paper is organized as follows. Section 2 presents the principles of the theory used in the paper. Section 3 details our proposed approach. Results from numerical experiments are shown and discussed in Section 4, with the conclusions and future directions presented in Section 5.

2. BACKGROUND

This section presents a background on multi-objective optimization, with a special attention dedicated to the multi-objective knapsack problem, and the basic concepts associated to EDAs.

2.1 Multi-objective optimization

A general multi-objective optimization problem (MOP) includes decision variables, objective functions, and constraints where objective functions and constraints are functions of the decision variables [33, 44]. Mathematically, a maximization MOP can be defined as:

$$\begin{aligned} \max_{\mathbf{x}} \mathbf{y} = \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{subject to } \mathbf{h}(\mathbf{x}) &= (h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_k(\mathbf{x})) \leq 0, \\ \mathbf{x} &= (x_1, x_2, \dots, x_n) \in X, \\ \mathbf{y} &= (y_1, y_2, \dots, y_m) \in Y, \end{aligned} \quad (1)$$

where \mathbf{x} is the decision vector, X is the variable space, \mathbf{y} is

the objective vector, Y is the objective space and $\mathbf{h}(\mathbf{x}) \leq 0$ is the set of constraints which determines a set of feasible solutions X_f .

The set of solutions of MOP includes decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another - these vectors are called Pareto optimal set. The idea of Pareto optimality is based on the Pareto dominance. A decision vector \mathbf{u} dominates a decision vector \mathbf{v} iff $f_i(\mathbf{u}) \geq f_i(\mathbf{v})$ for $i = 1, 2, \dots, m$ with $f_i(\mathbf{u}) > f_i(\mathbf{v})$ for at least one i . The vector \mathbf{u} is called Pareto optimal if there is no vector \mathbf{v} which dominates vector \mathbf{u} in the decision space X . In the objective space the set of nondominated solutions lies on a surface known as Pareto optimal front. The goal of the optimization is to find a representative sampling of solutions along the Pareto optimal front.

The 0/1 knapsack problem is a widely studied problem due to its practical importance. Recently, the multi-objective generalization of this problem has been well studied and many algorithms for solving it have been proposed [12, 37].

In this work we use the multi-objective formulation proposed in [12, 37, 44]. Given a total of R objective functions (knapsacks) and Q items, a_{rq} is the profit of item $q = 1, \dots, Q$, according to knapsack $r = 1, \dots, R$, b_{rq} denotes the weight of item q according to knapsack r , and c_r is the constraint capacity of knapsack r . The multi-objective knapsack problem can be formulated as follows:

$$\begin{aligned} \max_{\mathbf{x}} \mathbf{z}(\mathbf{x}) &= (z_1(\mathbf{x}), \dots, z_R(\mathbf{x})) \\ \text{subject to } \sum_{q=1}^Q b_{rq} x_q &\leq c_r, \quad r = 1, \dots, R \\ \text{with } z_r(\mathbf{x}) &= \sum_{q=1}^Q a_{rq} x_q, \quad \mathbf{x} \in \{0, 1\}^Q \end{aligned} \quad (2)$$

where \mathbf{x} is a Q -dimension binary vector, a_{rq} , b_{rq} and c_r are nonnegative coefficients, such that $x_q = 1$ means that item q is selected to be in all knapsacks.

2.2 Estimation of Distribution Algorithms

The main idea of EDAs [24, 27] is to extract and represent, using a probabilistic graphical model (PGM), the regularities shared by a subset of high-valued problem solutions. The PGM then samples new solutions biasing the search to areas where optimal solutions are more likely to be found.

Usually, an EDA starts by generating a random set of solutions. Solutions are then evaluated using one or more objective functions and a subset of them is selected according to a pre-defined criterion. The selected solutions are used to learn a PGM which is then applied to sample new solutions and the cycle of evaluation, selection, modeling, and sampling is repeated until a stop condition is fulfilled.

EDAs have also been applied to solve multi-objective problems. Normally they integrate both, the model building and sampling techniques, into evolutionary multi-objective optimizers using special selection schemes [17, 22]. However, recently the role of the probabilistic model has been extended to model the dependencies between variables and objectives [16]. In addition, MOEDAs can be notably enhanced by adding a local optimizer that can refine the solutions found by sampling from the PGM [23, 39, 43]. In this paper, we use this type of enhancement in the context of multi-objective optimization.

2.3 Bayesian Networks as Probabilistic Graphical Model

Bayesian networks (BN) are directed acyclic graphs whose nodes represent variables, and whose missing edges encode conditional independencies between variables. Random variables represented by nodes may be observable quantities, latent variables, unknown parameters or hypotheses. 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 [6, 18]. Mathematically, an acyclic BN with directed edges encodes a joint probability distribution. This can be written as in 3:

$$p(X) = \prod_{i=0}^{n-1} p(X_i | Pa_{X_i}) \quad (3)$$

where $X = (X_0, \dots, X_{n-1})$ is the vector of variables, Pa_{X_i} is the set of parents of X_i in the network (the set of nodes from which there exists an arc to X_i) and $p(X_i | Pa_{X_i})$ is the conditional probability of X_i given Pa_{X_i} . This distribution can be used to generate new instances using the marginal and conditional probabilities in a modeled data set. Therefore, BN's are considered the most prominent Probabilistic Graphical Models (PGMs) [21] and often are used for modeling multinomial data with discrete variables [25].

There are three main approaches to learn these PGMs structures: score-based learning, constraint-based learning, and hybrid methods [40]. Score-based learning methods evaluate the quality of Bayesian network structures using a scoring function and selects the best one, like BD-metric [10] and K2-metric [6]. Constraint-based learning methods typically use statistical testing to identify conditional independence relations from the data and build a Bayesian network structure that best fits those relations [25]. Hybrid methods aim to integrate the previous two approaches, like what was done in [26, 38]. The authors used constraint-based learning to create a skeleton graph and then used score-based learning to find a high-scoring network structure that is a subgraph of the skeleton.

In our implementation, we have used a simple greedy algorithm with only arc additions allowed in order to maximize the value of K2-metric and we also assume a causal relation between objectives and variables/parameters, but different assumptions can be made in the HMOBEDA framework. In the implemented concept, objectives are parents in the network, and variables and parameters are child nodes with no inter-relation among each other.

3. THE PROPOSAL: HMOBEDA

In the proposed approach, the goal is to include as part of the probabilistic model not only decision variables, but also objectives and parameters that control the embedded LS. In this section we explain how this strategy can be implemented detailing the proposed algorithm named HMOBEDA.

3.1 Encoding scheme

In the EDA population, every individual is represented by a joint vector $(\mathbf{x}, \mathbf{z}, \mathbf{p}) = (x_1, \dots, x_Q, z_1, \dots, z_R, p_1 \dots p_L)$, of size $Q + R + L$, denoting the decision variables (i.e. the solution of the addressed problem instance), objectives, and

LS parameters, respectively. The components \mathbf{x} , \mathbf{z} and \mathbf{p} can be specified as:

- \mathbf{x} is a binary vector of items, with element $x_q \in \{0, 1\}$, $q = 1 \dots Q$, indicating the presence or absence of the associated item;
- \mathbf{z} is a vector of objectives, with element z_r , $r = 1 \dots R$, representing the discrete value of r^{th} objective.
- \mathbf{p} is a vector of elements regarding local search, with element p_l , $l = 1 \dots L$, indicating the l^{th} discrete value associated with the LS procedure - see Section 3.2.3. In this work we assume the following elements: (i) $N_{iter} \in \{5, 6, \dots, 20\}$: maximum number of iterations of LS; (ii) $T_{Fcal} \in \{1, 2\}$: the type of fitness calculation: (1) Linear Combination or (2) Alternation of included objectives in both cases; (iii) $N_{DAd/Ins} \in \{1, 2\}$: neighborhood type: drop-add (1) or insertion (2); (iv) $\mathbf{Obj}_{ls} = (b_1, \dots, b_m)$, where $b_i \in \{0, 1\}$: binary vector indicating whether objectives are included or not in the fitness computation of LS neighbors.

3.2 HMOBEDA Algorithm

The main steps performed by HMOBEDA are described in Algorithm 1. The proposed algorithm makes use of specific strategies described in the next subsections.

3.2.1 Initialization

The *Initialization process* randomly generates every vector \mathbf{x} and \mathbf{p} of the initial population Pop^1 , a total of N vector solutions are generated. For each solution in Pop^1 the values of all objectives are calculated based on the objective functions and further discretized to form vector \mathbf{z} . Therefore, N joint vectors $(\mathbf{x}, \mathbf{z}, \mathbf{p})$ are obtained to compose the population in the first generation.

3.2.2 Non-dominated Sorting

In order to sort individuals of the current population, the proposed approach uses Non-dominated Sorting [35]. This technique sorts the population according to the dominance relationships established among solutions, producing a total of Tot_F sub-populations or fronts, $F_1, F_2, \dots, F_{Tot_F}$. Front F_1 corresponds to the set of non-dominated solutions (the best front). Sets F_i , $i = 2, \dots, Tot_F$ contain the non-dominated solutions when sets F_1, \dots, F_{i-1} are removed from the whole population. After dominance calculation, a diversity criterion named crowding distance (CD) is computed. Within each F_i , CD estimates the density of solutions surrounding a particular point $\mathbf{z} \in F_i$. For this, the method considers each of the R objectives independently, and sums the distances between the nearest points that have respectively smaller and greater value for the objective [8]. Individuals are sorted taking at first the dominance criterion and secondly, the CD criterion. Finally truncation selection is applied, i.e. the best N solutions are selected¹.

3.2.3 Local search

In this work we implement the hill climbing (HC) local search [28, 36] at each generation, aiming to improve a set of solutions selected by the truncation procedure.

¹This implementation was adapted from [34].

Algorithm 1 HMOBEDA framework

INPUT: N , population size;
 N_{PGM} , number of individuals selected to learn
 the probabilistic model;
 N_{smp} , number of individuals sampled from the
 probabilistic model;
 β , window of occurrence for the local search;
 Max_{ger} , maximum number of generations of
 EDA;
OUTPUT: ND , the final set of non-dominated solutions;
 {Initialization}
1: $Pop^1(\mathbf{x}) = \text{RandomVector}(N, Q)$; {repair if necessary}
2: $Pop^1(\mathbf{z}) = \text{Fitness}(Pop^1(\mathbf{x}))$;
3: $Pop^1(\mathbf{p}) = \text{RandomVector}(N, L)$;
4: $g = 1$;
 {EDA: main loop}
5: **while** $g \leq Max_{ger}$ **do**
6: **if** $g > 1$ {EDA sampling} **then**
7: $Pop_{smp} = \text{Sampling}(BNPM)$; {repair if necessary}
8: $Pop_{smp}(\mathbf{z}) = \text{Fitness}(Pop_{smp}(\mathbf{x}))$; {not surrogate as-
 sisted}
9: $Pop^g = \{Pop^{g-1} \cup Pop_{smp}\}$; {EDA survival}
10: **end if**
 {Non-dominated Sorting}
 {Defines k Pareto fronts from the best ($i = 1$) to the
 worst, and assigns a crowding distance}
11: $F_1 \dots F_k = \text{ParetoDominance}(Pop^g)$;
12: $Pop^g = \text{Select}(N, F_1 \cup \dots \cup F_k)$; {Truncation Selection}
 {Local search: performed at every β generations}
13: **if** $((g = 1) \text{ or } (g \bmod \beta = 0))$ **then**
14: $Pop_I = (\text{LS}(Pop^g))$; {repair if necessary}
15: $Pop^g = \text{ParetoDominance}(Pop_I)$;
16: **else**
17: $Pop_I = \emptyset$;
18: **end if**
 {EDA: learning the probabilistic model}
19: $Pop_{PGM}^g = \text{Selection}(N_{PGM}, Pop^g)$; {binary tourna-
 ment}
20: $BNPM = \text{ProbabilisticModelEstimation}(Pop_{PGM}^g)$;
21: $g = g + 1$;
22: **end while**
23: $ND = Pop^{g-1}(\mathbf{x})$;

For every solution in Pop^g (population at generation g), the HC-based local search (HCLS) generates a neighbor, calculates its fitness and compares it with the original solution fitness. If the neighbor is better it takes the original solution's place in Pop^g . In this work HCLS is applied at every β generations (steps 13 to 18 in Algorithm 1).

Neighbors are generated by swapping (drop-add) or inserting an item at a random position of the sequence depending on the $N_{DA/Ins}$ element of \mathbf{p} .

When LS compares the original solution with its neighbor, the fitness computation is based on \mathbf{Obj}_{js} and T_{Fcal} . It uses a linear combination (aggregation or one objective at a time for all active objectives in the corresponding neighbor and original solution (i.e., \forall objective $i \mid b_i = 1, b_i \in \mathbf{Obj}_{js}$). The decision between the two types of fitness calculation is set by the \mathbf{p} element T_{Fcal} . In the second case, at each iteration (of a total of N_{iter} performed by HCLS at the current generation

g) a different objective is chosen for the fitness calculation. This method of fitness calculation is based on [1, 23, 29, 39].

3.2.4 Probabilistic Modeling

After local search, HMOBEDA starts the PGM construction phase. First, $N/2$ individuals of Pop^g are selected through a binary tournament in the same way used by NSGA-II [8]. The procedure randomly selects two solutions from Pop^g . If both solutions belong to front F_i , the solution with greatest crowding distance is chosen. Otherwise, the solution that lies in the best front is selected. Then, Pop_{PGM}^g is obtained encompassing $N/2$ good individuals.

In order to learn the probabilistic model, the objective values are discretized providing t discrete values. The discretization process converts objective value into t discrete states considering the maximum possible value for each objective (Max^i). For each objective i , its discrete value is calculated as $\lceil objective^i / (Max^i / t) \rceil$. In the experiments, we set $t = 10$. BN is modeled using K2-metric and \mathbf{z} data collected from Pop_{PGM}^g . BN structure (encompassing local search parameters and objectives, in addition to decision variables) and BN parameters (joint probability distributions) are estimated based on [16].

Figure 1 presents an example of a PGM obtained. The model encodes the joint probabilistic model of Q decision variables, R objectives and L local search parameters. In this figure, Z represents the objectives, P the parameters and X the decision variables. The advantage of HMOBEDA over traditional EDA-based approaches is that besides providing good decision variables (based on the model captured from good solutions present in Pop_{PGM}^g) it can also provide variables and LS parameters more related with good values of objectives fixed as evidence. This naive Bayesian model is conceived to facilitate the sampling process: fixing objective values enables the estimation of their associated decision variables and LS parameters.

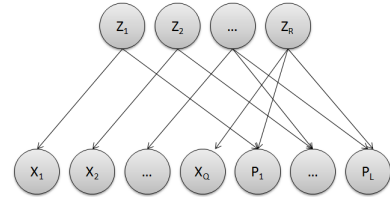


Figure 1: An example of a PGM used by HMOBEDA.

The probabilistic model (BNPM) is used to sample the set of new individuals (Pop_{smp}). In this case, not only decision variables \mathbf{x} , but also, local search parameters \mathbf{p} can be sampled. Vector \mathbf{z} is calculated based on the fitness function (in the case of surrogate assisted approaches, BNPM can also be used to sample the objective values). New individuals are generated from the joint distribution encoded by the network using probabilistic logic sampling. In this process, vector \mathbf{z} can be fixed as evidence, set with high values (for maximization problems). Therefore, it is possible to obtain a set of solutions and parameters associated with high objective values.

A union of the sampled population (Pop_{smp}) and Pop^g is used to create the new population for the next generation g , and the main loop HMOBEDA continues until the stop condition is fulfilled. At any stage of the evolutionary process, if

an infeasible solution is generated, HMOBEDA applies the same greedy repair method as in [44]. This method repairs a solution by removing items in an ascending order of the relation profit/weight until the constraints are satisfied.

4. EXPERIMENTS AND RESULTS

To analyze the performance of the proposed approach in multi-objective combinatorial optimization, HMOBEDA is evaluated in a set of MOKP instances with different sizes and number of objectives. These instances are considered because they are well explored in many works in the literature and have been previously addressed in [44]. We use six different instances, with 100 and 250 items, for 3, 4 and 5 objectives. We characterize each instance as R - Q , where R is the number of objectives and Q the number of items.

In this section, we compare HMOBEDA with HMOBEDA_f (i.e., HMOBEDA with fixed parameters), MBN-EDA [16], NSGA-II [8], S-MOGLS (a NSGA-II with local search) [13, 14], MOEA/D [41], and NSGA-III [7, 15]. HMOBEDA_f has no LS parameter encoded as a node in the Bayesian network model. It has been included in the comparison to evaluate the benefits of incorporating LS parameters into the probabilistic model. The difference between MBN-EDA and HMOBEDA_f is the LS applied to HMOBEDA_f. All algorithms have run in the same computer using MatLab, excepted MOEA/D and NSGA-III which have been adapted from [41] and [4, 7], respectively, using C++.

As commented in Section 2.1, we use the same problem formulation used in [12, 37, 44]. The values of a_q^r and b_q^r are specified as integers in the interval [10, 100]. Like in [44], the capacity c_r is specified as 50% of the sum of all weights related to each knapsack r .

4.1 Algorithms Settings

The parameters for each algorithm are shown in Table 1.

Table 1: Parameters of algorithms.

	Description	Value	Algorithm
N	Population size	100	HMOBEDA, HMOBEDA _f , MBN-EDA, NSGA-II, S-MOGLS
β	Local Search occurrence	1	HMOBEDA, HMOBEDA _f , S-MOGLS
N_{PGM}	Pop_{PGM}^g size	$N/2$	HMOBEDA, HMOBEDA _f , MBN-EDA, NSGA-II, S-MOGLS
N_{smp}	Pop_{smp}^g size	$10 * N$	HMOBEDA, HMOBEDA _f , MBN-EDA
p_c	Uniform Crossover probability	0.8	NSGA-II, S-MOGLS, NSGA-III, MOEA/D
p_m	Bit Flip mutation probability	1/500	NSGA-II, S-MOGLS, NSGA-III, MOEA/D

HMOBEDA_f assumes $T_{Fcal} = 0$, $N_{iter} = 20$ and $N_{DAd/Ins} = 0$ (these parameters have been empirically defined). For S-MOGLS we set the probability of LS (P_{ls}) as 0.1, the probability of Bit-Flip operation in LS as 4/500, and $N_{ls} = 20$, where N_{ls} is the number of neighbors to be examined [13].

For NSGA-III, we use the same configuration used in [7], i.e., the number of reference points (H) defines the population size N . For an R -objective problem, if p divisions are considered along each objective, [7] defines $H = C_{R+p-1}^p$ (e.g., $H = N = 126$, for 5 objectives and $p = 5$).

In MOEA/D, the number of the subproblems equals the population size N and the weight vectors $\lambda^1, \dots, \lambda^N$ is con-

trolled by the W configuration parameter, calculated as proposed in [41], which is same procedure used to generate the reference points in NSGA-III. The size of the neighborhood of each weight vector is $T = 10$ [7]. The weighted sum approach was used for the MOEA/D experiments².

All the algorithms run a total of $Max_{ger} = 100$ generations (LS iterations are not considered as generations); 20 independent executions are conducted for each algorithm and then the performance indicators are computed.

4.2 Performance Metrics

Since for the addressed problem the optimal Pareto front for each instance is not known, we use a reference set which is constructed by gathering all non-dominated solutions obtained by all algorithms over all executions. The reference set (the best known Pareto front) is denoted by Ref . The performance of an algorithm is measured by the difference HV^- between the hypervolume of its solutions set and the hypervolume of the reference set. So, smaller values correspond to higher quality solutions in non-dominated sets and it indicates both a better convergence as well as a good coverage of the reference set [2].

4.3 Numerical results

In this section, we aim to compare HMOBEDA with HMOBEDA_f, MBN-EDA, NSGA-II, S-MOGLS, NSGA-III and MOEA/D. The sets provided by these algorithms will denote the non-dominated solutions found among all the executions (the reference set). Table 2 presents the percentage of non-dominated solutions of each algorithm that composes the reference set (cardinality).

Table 2: Percentage of $|Ref|$ corresponding to each algorithm.

Problem	3-100	4-100	5-100	3-250	4-250	5-250
$ Ref $	502	1320	1112	601	687	1439
HMOBEDA	30.78	31.89	43.65	50.98	49.76	50.01
HMOBEDA _f	23.09	20.02	21.82	19.89	20.43	20.65
MBN-EDA	8.76	6.82	8.95	9.65	9.08	10.02
NSGA-II	4.98	4.01	5.45	1.02	3.02	3.92
S-MOGLS	2.65	5.59	6.10	0.54	4.07	6.45
NSGA-III	22.65	26.02	23.87	40.32	29.02	28.56
MOEA/D	28.87	27.37	25.12	42.76	30.14	29.56

Note that some non-dominated solutions are found by more than one algorithm. We conclude that for all instances HMOBEDA finds the greatest number of solutions in the reference of the reference set. For instances 3 – 250 and 5 – 250 more than half of the solutions in the reference set are originated from solutions produced by HMOBEDA.

Average run times are presented in Table 3.

We observed that for smaller search spaces, NSGA-II presents better results (an average of total runs). However, increasing the number of objectives and variables severely impacts the computational time of all approaches, mainly for NSGA-II. Also, HMOBEDA, NSGA-III and MOEA/D execution times are similar for all instances.

Table 4 shows the hypervolume difference (HV^-) averaged over 20 executions of each algorithm. We use PISA framework [31] to calculate HV^- . Due to the normality of the

²This approach is used in [41] implementation for MOKP, which can be downloaded from <http://http://dces.essex.ac.uk/staff/zhang/webofmoad.htm>.

Table 3: Average run time (min) of each algorithm.

Algorithm	3-100	4-100	5-100	3-250	4-250	5-250
HMOBEDA	5.22	9.71	23.39	13.76	23.48	50.08
HMOBEDA _f	4.79	8.35	20.94	10.36	20.21	46.08
MBN-EDA	3.26	6.12	15.94	12.25	15.06	35.01
NSGA-II	0.92	5.41	25.21	4.96	19.90	50.02
S-MOGLS	4.25	8.12	30.40	5.71	21.74	56.68
NSGA-III	5.03	10.04	21.64	15.21	24.45	54.46
MOEA/D	4.94	8.16	21.60	15.18	23.09	51.71

HV⁻ values, which has been verified through the Shapiro-Wilk normality test [5], the Anova test is applied to perform a statistical analysis on the results [3]. All tests have been executed with a confidence level of 95% ($\alpha = 0.05$) and 20 replications (executions of each algorithm).

Table 4: Average hypervolume differences (HV⁻).

Instance	3-100 $\times 10^{10}$	4-100 $\times 10^{14}$	5-100 $\times 10^{17}$	3-250 $\times 10^{11}$	4-250 $\times 10^{15}$	5-250 $\times 10^{19}$
HMOBEDA	2.16	1.81	4.82	3.62	4.14	5.37
HMOBEDA _f	2.55	1.99	5.42	4.18	4.90	6.05
MBN-EDA	2.75	2.11	5.76	4.54	5.19	6.36
NSGA-II	2.60	2.00	5.46	4.56	5.09	6.15
S-MOGLS	2.62	1.99	5.37	4.55	5.08	6.19
NSGA-III	2.38	1.83	5.37	4.04	4.69	6.00
MOEA/D	2.26	1.86	5.36	4.04	4.56	5.81

Table 5 shows the statistical analysis of the results for the algorithms for each instance, with respect to HV⁻ values. The null hypothesis that all the algorithms have HV⁻ values with no statistical difference can be rejected for all instances. A post-hoc analysis is performed to evaluate which algorithms present statistical difference. The entry related to the algorithm with the best average HV⁻ is shown in bold. The numbers in parentheses show the results of pairwise comparisons using Tukey’s posthoc test [11] with a significance level of $\alpha = 0.05$. The first number shows how many algorithms are significantly worse than the algorithm listed in the corresponding line, and the second number shows how many algorithms are significantly better.

Table 5: Results of post-hoc statistical test.

Algorithm	3-100	4-100	5-100	3-250	4-250	5-250
HMOBEDA	(3,0)	(4,0)	(6,0)	(6,0)	(6,0)	(6,0)
HMOBEDA _f	(0,0)	(0,2)	(0,1)	(0,1)	(0,1)	(0,1)
MBN-EDA	(0,2)	(0,3)	(0,1)	(0,3)	(0,3)	(0,3)
NSGA-II	(0,1)	(0,2)	(0,1)	(0,3)	(0,3)	(0,1)
S-MOGLS	(0,1)	(0,2)	(0,1)	(0,3)	(0,3)	(0,2)
NSGA-III	(0,0)	(4,0)	(0,1)	(3,1)	(3,1)	(1,1)
MOEA/D	(1,0)	(1,0)	(0,1)	(3,1)	(3,1)	(2,1)

Figures 2 and 3 illustrate the Tukey’s post-hoc test for HMOBEDA in comparison with other approaches: (a) instances with 3, (b) instances with 4 and (c) instances with 5 objectives. The continuous bold line represents the HV⁻ values obtained by HMOBEDA; the continuous line identifies the HV⁻ values obtained by algorithms with no significant difference from HMOBEDA; and the dashed line represents HV⁻ values with significant difference from HMOBEDA.

As we can see from these graphical results, the proposed approach presents statistical difference in comparison with almost all other algorithms. For Tukey’s post-hoc test, when confidence intervals do not overlap, HV⁻ values can be compared (and thus algorithms), otherwise they have similar

performances. An example can be seen in Figures 2 (a) and (b): for instances 3 – 100 the performances of HMOBEDA_f, MBN-EDA, NSGA-II and S-MOGLS are not statistically different and all of them are worse than HMOBEDA (which has similar performance to NSGA-III and MOEA/D); also, for instances 4 – 100 the performances of HMOBEDA, NSGA-III and MOEA/D are not statistically different; NSGA-III and HMOBEDA are better than HMOBEDA_f, MBN-EDA, NSGA-II and S-MOGLS. For the remaining instances we clearly see that confidence intervals do not overlap, so we can conclude that, based on HV⁻, HMOBEDA has the best performance among the algorithms compared.

Therefore, we can affirm that HMOBEDA is a competitive approach for the instances considered in this work. We have observed that it is statistically superior to the compared approaches, except for instances of small size where HMOBEDA, NSGA-III and MOEA/D are not statistically different. Also, HMOBEDA is better than HMOBEDA_f, which evidences that the proposed approach is capable of setting good values for the LS parameters.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a new approach for integrating local search methods and probabilistic modeling in the context of multi-objective optimization. The backbone of our proposal is a MOEDA-based approach that uses a Bayesian Network as its probabilistic model. Additionally, it is embedded with a hill climbing procedure as local optimizer. The distinguished characteristic of HMOBEDA is that it adopts joint probabilistic model of objectives, variables and LS parameters to generate new individuals through the sample process. The Bayesian network is learned using the K-2 method and individuals selected by non-dominated sort. Current solutions are exploited by the LS procedure according to the encoded LS parameters. Thus, the algorithm can explore in a proper way other solutions in the decision space, contributing to the diversity of the non-dominated final set. As the LS parameters are part of the network, they are adapted as the algorithm evolves.

Local search is commonly applied as a method to improve the results of evolutionary optimization methods as the ones analyzed in our paper. The main question for us was to determine whether using auto-adapted LS parameters, and letting the Bayesian network represent and set these parameters, would improve the search which can be answered in an affirmative way based on the experimental results. The algorithm was tested in six instances of the multi-objective knapsack problem. We concluded that the proposed hybrid EDA provides better performance when compared with the other evolutionary algorithms investigated. The explanation is based on the fact that since the LS parameters are in the same probabilistic model, HMOBEDA provides an automatic and informed decision at the time of setting these parameters, thus a variety of non-dominated solutions could be found attending different stages of the evolutionary process. We showed that the better performance of the proposed approach is more related with the probabilistic model than with LS: HMOBEDA_f did not provide the same trade-off even using LS with tuned parameters.

In the future, different repair methods and LS procedures can also be tested. Finally, we intend to compare HMOBEDA with a baseline method commonly used in hyperheuristics contexts which randomly generates LS parameters.

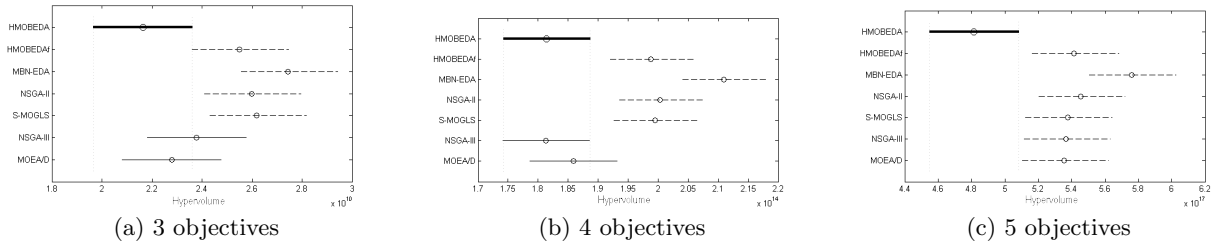


Figure 2: Graphical results of post-hoc test for HMOBEDA versus other approaches - MOKP with 100 items

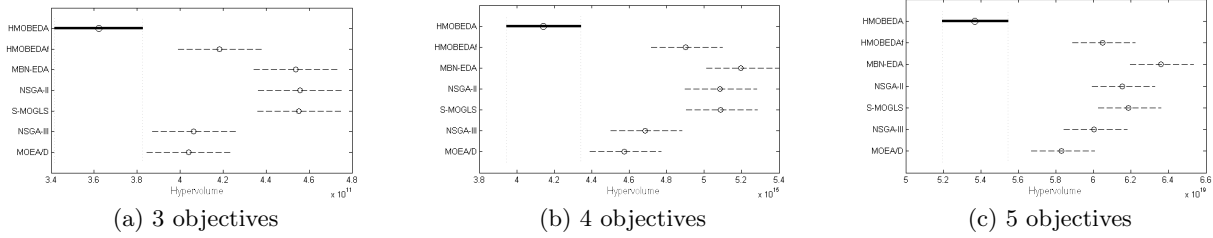


Figure 3: Graphical results of post-hoc test for HMOBEDA versus other approaches - MOKP with 250 items

6. ACKNOWLEDGEMENTS

R. Santana acknowledges support by: IT-609-13 program (Basque Government), TIN2013-41272P (Spanish Ministry of Science and Innovation) and CNPq Program Science Without Borders Nos.: 400125/2014-5 (Brazil Government).

M.Martins acknowledges CAPES/Brazil.

7. REFERENCES

- [1] C. W. Ahn, E. Kim, H.-T. Kim, D.-H. Lim, and J. An. A hybrid multiobjective evolutionary algorithm: Striking a balance with local search. In *Mathematical and Computer Modelling*, volume 52, pages 2048–2059, 2010.
- [2] J. Arroyo, R. dos Santos Ottoni, and A. de Paiva Oliveira. Multi-objective variable neighborhood search algorithms for a single machine scheduling problem with distinct due windows. In *Electronic Notes in Theoretical Computer Science*, volume 281, pages 5–19, 2011.
- [3] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury, USA, second edition, 2001.
- [4] T.-C. Chiang. nsga3cpp: A C++ implementation of NSGA-III. <http://web.ntnu.edu.tw/tcchiang/publications/nsga3cpp/nsga3cpp.htm>, 2014. Access date: 12 july.2015.
- [5] W. Conover. *Practical Nonparametric Statistics*. Wiley, third edition, 1999.
- [6] G. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.
- [7] K. Deb and H. Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.
- [8] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. In *IEEE Transactions on Evolutionary Computation*, volume 6, pages 182–197, 2002.
- [9] D. Garrett and D. Dasgupta. An Empirical Comparison of Memetic Algorithm Strategies on the Multiobjective Quadratic Assignment Problem. In *IEEE Symposium on Computational Intelligence in Multi-criteria Decision-Making*, pages 80–87, Nashville, TN, 2009.
- [10] D. Heckerman, D. Geiger, and D. Chickering. Learning bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.
- [11] Y. Hochberg and A. C. Tamhane. *Multiple Comparison Procedures*. John Wiley and Sons, Hoboken, NJ, 1987.
- [12] H. Ishibuchi, N. Akedo, and Y. Nojima. Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems. *IEEE Transactions on Evolutionary Computation*, 19(2):264–283, 2015.
- [13] H. Ishibuchi, Y. Hitotsuyanagi, and Y. Nojima. Scalability of Multiobjective Genetic Local Search to Many-Objective Problems: Knapsack Problem Case Studies. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*, pages 3586–3593, June 2008.
- [14] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima. Implementation of Multiobjective Memetic Algorithms for Combinatorial Optimization problems: A Knapsack Problem Case Study. 171:27–49, 2009.
- [15] H. Jain and K. Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, Aug 2014.
- [16] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga. Multiobjective Estimation of Distribution Algorithm Based on Joint Modeling of Objectives and Variables. 18:519–542, 2014.
- [17] N. Khan, D. E. Goldberg, and M. Pelikan.

- Multi-objective Bayesian optimization algorithm. IlliGAL Report No. 2002009, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 2002.
- [18] K. B. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence*. CRC Press, second edition, 2010.
- [19] N. Krasnogor, B. P. Blackburne, E. K. Burke, and J. D. Hirst. Multimeme algorithms for Protein Structure Prediction. In *Parallel Problem Solving from Nature - PPSN VII*, volume 2439 of *Lecture Notes in Computer Science*, pages 769–778, Spain, 2002.
- [20] N. Krasnogor and S. Gustafson. Toward truly "memetic" memetic algorithms: discussion and proof of concepts. In *Advances in Nature-Inspired Computation: the PPSN VII Workshops*, 2002.
- [21] P. Larrañaga, H. Karshenas, C. Bielza, and R. Santana. A review on probabilistic graphical models in evolutionary computation. In *Journal of Heuristics*, volume 18, pages 795–819, 2012.
- [22] M. Laumanns and J. Ocenasek. Bayesian Optimization Algorithms for Multi-Objective Optimization. In *Parallel Problem Solving from Nature-PPSN VII*, volume 2439 of *Lecture Notes in Computer Science*, pages 298–307, 2002.
- [23] H. Li, Q. Zhang, E. Tsang, and J. A. Ford. Hybrid Estimation of Distribution Algorithm for Multiobjective Knapsack Problem. In *Evolutionary Computation in Combinatorial Optimization*, pages 145–154. Springer Berlin Heidelberg, 2004.
- [24] H. Muhlenbein and G. Paab. From Recombination of Genes to the Estimation of Distributions I. Binary parameters. In *Parallel Problem Solving from Nature-PPSN IV*, *Lecture Notes in Computer Science* 1411, pages 178–187, 1996.
- [25] J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, San Mateo CA, 1988.
- [26] E. Perrier, S. Imoto, S. Miyano, and M. Chickering. Finding Optimal Bayesian Network Given a Super-Structure. *Journal of Machine Learning Research*, 9:2251–2286, 2008.
- [27] P. Larrañaga and J. A. Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer, Netherlands, 2002.
- [28] S. J. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, second edition, 2003.
- [29] B. Samir, L. Yacine, and B. Mohamed. Local Search Heuristic for Multiple Knapsack Problem. *International Journal of Intelligent Information Systems*, 4(2):35–39, 2015.
- [30] R. Santana, P. Larrañaga, and J. A. Lozano. Combining variable neighborhood search and estimation of distribution algorithms in the protein side chain placement problem. *Journal of Heuristics*, 14:519–547, 2008.
- [31] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. Pisa - A Platform and Programming Language Independent Interface for Search Algorithms. In *Evolutionary multi-criterion optimization (EMO 2003)*, *Lecture notes in computer science*, pages 494–508, Berlin, 2003.
- [32] J. Schwarz and J. Ocenasek. Multiobjective Bayesian Optimization Algorithm for Combinatorial Problems: Theory and Practice. *Neural Network World*, 11(5):423–441, 2001.
- [33] J. Schwarz and J. Ocenasek. Pareto Bayesian Optimization Algorithm for the Multiobjective 0/1 Knapsack problem. In *7th International Mendel Conference on Soft Computing*, pages 131–136, 2001.
- [34] A. Seshadri. Multi-Objective Optimization using Evolutionary Algorithm. <http://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm>, 2009. Access date: 2 feb.2016.
- [35] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. In *Evolutionary Computation*, volume 2, pages 221–248, 1994.
- [36] E.-G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009.
- [37] Y. Tanigaki, K. Narukawa, Y. Nojima, and 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)*, pages 637–642, 2014.
- [38] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The Max-min Hill-climbing Bayesian Network Structure Learning Algorithm. *Machine Learning*, 65(1):31–78, 2006.
- [39] L. Wang, S. yao Wang, and Y. Xu. An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem. In *Expert Systems with Applications*, volume 39, pages 5593–5599, 2012.
- [40] C. Yuan and B. Malone. Learning Optimal Bayesian Networks: A Shortest Path Perspective. *Journal of Artificial Intelligence Research*, 48(1):23–65, 2013.
- [41] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.
- [42] Q. Zhang, J. Sun, and E. P. K. Tsang. Evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE Transactions on Evolutionary Computation*, 9(2):192–200, 2005.
- [43] A. Zhou, J. Sun, and Q. Zhang. An Estimation of Distribution Algorithm with Cheap and Expensive Local Search. In *IEEE Transactions on Evolutionary Computation*, accepted, 2015.
- [44] E. Zitzler and L. Thiele. Multiple objective evolutionary algorithms: A comparative case study and the strength Pareto approach. In *IEEE Transactions on Evolutionary Computation*, volume 3, pages 257–271, 1999.