

FEDERAL UNIVERSITY OF TECHNOLOGY - PARANÁ
GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER
ENGINEERING

MARCELLA SCOCZYNSKI RIBEIRO MARTINS

**A HYBRID MULTI-OBJECTIVE BAYESIAN ESTIMATION OF
DISTRIBUTION ALGORITHM**

THESIS

CURITIBA

2017

MARCELLA SCOCZYNSKI RIBEIRO MARTINS

**A HYBRID MULTI-OBJECTIVE BAYESIAN ESTIMATION OF
DISTRIBUTION ALGORITHM**

Thesis submitted to the Graduate Program in Electrical and Computer Engineering of Federal University of Technology - Paraná in partial fulfillment for the degree of “Doutor em Ciências” (Computer Engineering).

Advisor: Myriam Regattieri Delgado

Co-Advisor: Ricardo Lüders

CURITIBA

2017

Dados Internacionais de Catalogação na Publicação

N244h
2017 Martins, Marcella Scoczynski Ribeiro
A hybrid multi-objective bayesian estimation of distribution
algorithm / Marcella Scoczynski Ribeiro Martins.-- 2017.
124 f. : il. ; 30 cm

Texto em inglês com resumo em português
Disponível também via World Wide Web
Tese (Doutorado) – Universidade Tecnológica Federal do Pa-
raná. Programa de Pós-graduação em Engenharia Elétrica e In-
formática Industrial, Curitiba, 2017
Bibliografia: f. 97-108

1. Algoritmos computacionais. 2. Probabilidade (Matemática).
3. Teoria bayesiana de decisão estatística. 4. Algoritmos heurísti-
cos. 5. Otimização matemática. 6. Engenharia elétrica – Teses. I.
Delgado, Myriam Regattieri de Biase da Silva. II. Lüders, Ricardo.
III. Universidade Tecnológica Federal do Paraná. Programa de
Pós-Graduação em Engenharia Elétrica e Informática Industrial.
IV. Título.

CDD: Ed. 23 – 621.3

Biblioteca Central da UTFPR, Câmpus Curitiba
Bibliotecário: Adriano Lopes CRB9/1429

TERMO DE APROVAÇÃO DE TESE Nº 163

A Tese de Doutorado intitulada “**A Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm**”, defendida em sessão pública pelo(a) candidato(a) **Marcella Scoczynski Ribeiro Martins**, no dia 11 de dezembro de 2017, foi julgada para a obtenção do título de Doutor em Ciências, área de concentração Engenharia de Computação, e aprovada em sua forma final, pelo Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial.

BANCA EXAMINADORA:

Prof(a). Dr(a). Myriam Regattieri de Biase da Silva Delgado -Presidente– (UTFPR)
Prof(a). Dr(a). Roberto Santana Hermida – (University of the Basque Country)
Prof(a). Dr(a). Gilberto ReynosoMeza – (PUC-PR)
Prof(a). Dr(a). Aurora Trinidad Ramirez Pozo – (UFPR)
Prof(a). Dr(a). Viviana Cocco Mariani - (PUC-PR)
Prof (a). Dr(a). Lucia Valeria Ramos de Arruda

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Curitiba, 11 de dezembro de 2017.

ACKNOWLEDGEMENTS

As all of these long and hard working years went by, I realize I would not be able to accomplish anything, if God was not by my side. I thank for all the prayers my mother, my family and my friends have said in my behalf. I specially thank my Professors, Myriam and Ricardo, which have always supported me. She is an example of wisdom, kindness and respect - probably she uses a bit of magic too. He does not spare efforts when helping others, and I seek to become a person and a Professor like them, and to have the opportunity to take care of my students like they have done to me: as friends. I would also like to thank Roberto, for the help he has provided for this work. His suggestions and opinions were very important. Plus, I extend my acknowledgment to the Professors who kindly accepted to be part of my presentation board, as well as their important contributions. I also thank Carol, Ledo and Richard, who always pointed out relevant ideas to this research.

I can not forget the great opportunity I had to attend extraordinary conferences due to help of my Graduate Program, the CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) government agency, the student grants by ACM (Association for Computing Machinery) and the Department of Electronics - UTFPR Ponta Grossa, where I work. I thank my colleagues Jeferson, Virginia, Claudinor and Frederic, for all the help with classes, laboratories and with the server, where I ran all the experiments.

Thank you, my dear parents Ary and Deonéia, grandparents Anna, Ary and Digui, brothers Rafael and Rodrigo, my friends Lilian, Fernando, Mohamed, Carlos Alexandre, Eloi, Lorena and Rodrigo, and my love Luís Guilherme, who gave me the most beautiful flower on our garden - Ana Rosa. Thank you, my little daughter, to understand my absence, and thank you, mom, to take care of her during this.

"Soñar el sueño imposible. Sufrir la angustia implacable. Pisar donde los bravos no osan. Notar el mal irreparable. Enfrentar el enemigo invencible. Intentar cuando las fuerzas se desvanecen. Alcanzar la estrella inatingible; esa es mi búsqueda..."

Miguel de Cervantes, *Don Quijote de La Mancha*.

ABSTRACT

MARTINS, Marcella Scoczynski Ribeiro. A HYBRID MULTI-OBJECTIVE BAYESIAN ESTIMATION OF DISTRIBUTION ALGORITHM. 124 f. Thesis – Graduate Program in Electrical and Computer Engineering, Federal University of Technology - Paraná. Curitiba, 2017.

Nowadays, a number of metaheuristics have been developed for dealing with multiobjective optimization problems. Estimation of distribution algorithms (EDAs) are a special class of metaheuristics that explore the decision variable space to construct probabilistic models from promising solutions. The probabilistic model used in EDA captures statistics of decision variables and their interdependencies with the optimization problem. Moreover, the aggregation of local search methods can notably improve the results of multi-objective evolutionary algorithms. Therefore, these hybrid approaches have been jointly applied to multi-objective problems. In this work, a Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA), which is based on a Bayesian network, is proposed to multi and many objective scenarios by modeling the joint probability of decision variables, objectives, and configuration parameters of an embedded local search (LS). We tested different versions of HMOBEDA using instances of the multi-objective knapsack problem for two to five and eight objectives. HMOBEDA is also compared with five cutting edge evolutionary algorithms (including a modified version of NSGA-III, for combinatorial optimization) applied to the same knapsack instances, as well to a set of MNK-landscape instances for two, three, five and eight objectives. An analysis of the resulting Bayesian network structures and parameters has also been carried to evaluate the approximated Pareto front from a probabilistic point of view, and also to evaluate how the interactions among variables, objectives and local search parameters are captured by the Bayesian networks. Results show that HMOBEDA outperforms the other approaches. It not only provides the best values for hypervolume, capacity and inverted generational distance indicators in most of the experiments, but it also presents a high diversity solution set close to the estimated Pareto front.

Keywords: Multi-objective optimization, metaheuristic, bayesian network, estimation of distribution algorithm, local search, hybridization

RESUMO

MARTINS, Marcella Scoczynski Ribeiro. UM ALGORITMO DE ESTIMAÇÃO DE DISTRIBUIÇÃO HÍBRIDO MULTIOBJETIVO COM MODELO PROBABILÍSTICO BAYESIANO. 124 f. Thesis – Graduate Program in Electrical and Computer Engineering, Federal University of Technology - Paraná. Curitiba, 2017.

Atualmente, diversas metaheurísticas têm sido desenvolvidas para tratar problemas de otimização multiobjetivo. Os Algoritmos de Estimação de Distribuição são uma classe específica de metaheurísticas que exploram o espaço de variáveis de decisão para construir modelos de distribuição de probabilidade a partir das soluções promissoras. O modelo probabilístico destes algoritmos captura estatísticas das variáveis de decisão e suas interdependências com o problema de otimização. Além do modelo probabilístico, a incorporação de métodos de busca local em Algoritmos Evolutivos Multiobjetivo pode melhorar consideravelmente os resultados. Estas duas técnicas têm sido aplicadas em conjunto na resolução de problemas de otimização multiobjetivo. Nesta tese, um algoritmo de estimação de distribuição híbrido, denominado HMOBEDA (*Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm*), o qual é baseado em redes bayesianas e busca local é proposto no contexto de otimização multi e com muitos objetivos a fim de estruturar, no mesmo modelo probabilístico, as variáveis, objetivos e as configurações dos parâmetros da busca local. Diferentes versões do HMOBEDA foram testadas utilizando instâncias do problema da mochila multiobjetivo com dois a cinco e oito objetivos. O HMOBEDA também é comparado com outros cinco métodos evolucionários (incluindo uma versão modificada do NSGA-III, adaptada para otimização combinatória) nas mesmas instâncias do problema da mochila, bem como, em um conjunto de instâncias do modelo MNK-landscape para dois, três, cinco e oito objetivos. As fronteiras de Pareto aproximadas também foram avaliadas utilizando as probabilidades estimadas pelas estruturas das redes resultantes, bem como, foram analisadas as interações entre variáveis, objetivos e parâmetros de busca local a partir da representação da rede bayesiana. Os resultados mostram que a melhor versão do HMOBEDA apresenta um desempenho superior em relação às abordagens comparadas. O algoritmo não só fornece os melhores valores para os indicadores de hipervolume, capacidade e distância invertida geracional, como também apresenta um conjunto de soluções com alta diversidade próximo à fronteira de Pareto estimada.

Palavras-chave: Otimização Multi-objetivo, metaheurística, rede bayesiana, algoritmo de estimação de distribuição, modelo de probabilidade, busca local, hibridização

LIST OF FIGURES

FIGURE 1	– An example of Pareto front and Pareto dominance in the objective space. .	21
FIGURE 2	– Illustration of the boundary intersection approach (ZHANG; LI, 2007). . . .	26
FIGURE 3	– The general framework of an EDA.	33
FIGURE 4	– An MBN classifier with 3 class variables and 4 feature variables.	41
FIGURE 5	– A naive MBN classifier with 3 class variables and 4 feature variables. . . .	43
FIGURE 6	– The HMOBEDA Framework.	55
FIGURE 7	– An example of the PGM structure used by HMOBEDA.	61
FIGURE 8	– Different HMOBEDA versions considered in the experiments.	67
FIGURE 9	– Evidences (blue circles) from the approximated Pareto front with 2 objectives for (a) HMOBEDA _{IDEAL} , (b) HMOBEDA _{EXT} and (c) HMOBEDA _{CPT}	70
FIGURE 10	– A probabilistic view of the approximated Pareto front for 2 objectives and 100 items for HMOBEDA _{IDEAL} , HMOBEDA _{EXT} and HMOBEDA _{CPT} . . .	78
FIGURE 11	– A probabilistic view of the approximated Pareto front for 2 objectives and 250 items for HMOBEDA _{IDEAL} , HMOBEDA _{EXT} and HMOBEDA _{CPT} . . .	78
FIGURE 12	– The Euclidean distance between each solution and the ideal point for 2 objectives with (a) 100 items and (b) 250 items.	79
FIGURE 13	– The Euclidean distance between each solution and the ideal point for 3 objectives with (a) 100 items and (b) 250 items.	79
FIGURE 14	– The Euclidean distance between each solution and the ideal point for 4 objectives with (a) 100 items and (b) 250 items.	80
FIGURE 15	– The Euclidean distance between each solution and the ideal point for 5 objectives with (a) 100 items and (b) 250 items.	80
FIGURE 16	– The Euclidean distance between each solution and the ideal point for 8 objectives with (a) 100 items and (b) 250 items.	80
FIGURE 17	– 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 $X_q, q \in \{1, \dots, 100\}$	90
FIGURE 18	– Glyph representation of the three LS parameters (spokes) for each objective Z_1 to Z_8 of instance 8-100.	90
FIGURE 19	– 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 $X_q, q \in \{1, \dots, 20\}$	91
FIGURE 20	– Glyph representation of the three LS parameters (spokes) for each objective Z_1 to Z_3 of M3N50K6 instance.	91
FIGURE 21	– Glyph representation of the three LS parameters (spokes) for each objective Z_1 to Z_8 of M8N100K10 instance.	92
FIGURE 22	– An example of structure learned by K2 Algorithm.	114
FIGURE 23	– An overview of the proposed MBN-EDA by Karshenas et al. (2014). . . .	122

LIST OF TABLES

TABLE 1	– MOEAs to MOKP main characteristics	49
TABLE 2	– MOEAs to MNK-landscape problem main characteristics	54
TABLE 3	– Summarizing EDAs approaches to the knapsack problem and NK-landscape problem	64
TABLE 4	– HMOBEDA _{<i>f-inst</i>} parameters.	69
TABLE 5	– Parameters of the MOEAs used for solving the MOKP and MNK-landscape problem.	70
TABLE 6	– Results for pairwise comparison between HMOBEDA _{<i>K2</i>} and HMOBEDA _{<i>BN-HC</i>} using Mann-Whitney-Wilcoxon test with $\alpha = 5\%$ for each problem instance.	72
TABLE 7	– Average Run time (minutes) for each algorithm and instance.	73
TABLE 8	– Results for pairwise run time comparisons between HMOBEDA _{<i>K2</i>} and HMOBEDA _{<i>BN-HC</i>} using Mann-Whitney-Wilcoxon test with $\alpha = 5\%$ for each problem instance.	73
TABLE 9	– Results for pairwise comparisons between HMOBEDA _{<i>CD</i>} and HMOBEDA _{<i>hype</i>} using Mann-Whitney-Wilcoxon test with $\alpha = 5\%$ for each problem instance.	74
TABLE 10	– Average run time (minutes) for each algorithm and instance.	74
TABLE 11	– Results for pairwise run time comparisons between HMOBEDA _{<i>CD</i>} and HMOBEDA _{<i>hype</i>} using Mann-Whitney-Wilcoxon test with $\alpha = 5\%$ for each problem instance.	74
TABLE 12	– Results for pairwise comparisons between HMOBEDA _{<i>online</i>} and its off-line versions using Kruskal-Wallis and Dunn-Sidak’s post-hoc tests with $\alpha = 5\%$ for each problem instance.	75
TABLE 13	– Average run time (minutes) for each algorithm and instance.	75
TABLE 14	– Results for pairwise run time comparisons among HMOBEDA _{<i>online</i>} and the off-line versions using Kruskal-Wallis and Dunn-Sidak’s post-hoc tests with $\alpha = 5\%$ for each problem instance.	75
TABLE 15	– Results for pairwise comparisons between HMOBEDA _{<i>IDEAL</i>} , HMOBEDA _{<i>EXT</i>} and HMOBEDA _{<i>CPT</i>} using Kruskal-Wallis and Dunn-Sidak’s post-hoc tests with $\alpha = 5\%$ for each problem instance.	76
TABLE 16	– Average run time (minutes) for each algorithm and instance.	77
TABLE 17	– Results for pairwise run time comparisons between HMOBEDA _{<i>IDEAL</i>} , HMOBEDA _{<i>CPT</i>} and HMOBEDA _{<i>EXT</i>} versions using Kruskal-Wallis and Dunn-Sidak’s post-hoc tests with $\alpha = 5\%$ for each problem instance.	77
TABLE 18	– Average HV ⁻ and IGD over 30 executions.	82
TABLE 19	– Capacity metrics over 30 executions for each algorithm. The best values are in bold	82
TABLE 20	– Average Run time (minutes) for each algorithm and instance.	83
TABLE 21	– Average HV ⁻ over 30 independent executions.	84
TABLE 22	– Average IGD over 30 executions landscapes.	85

TABLE 23	– Overall Non-dominated Vector Generation (ONVG) over 30 executions for each algorithm. The best values are in bold.	86
TABLE 24	– Error Ratio over 30 executions for each algorithm. The best values are in bold.	87
TABLE 25	– Average run time (minutes) for each algorithm and instance.	89
TABLE 26	– A given dataset for the K2 algorithm example	109

LIST OF ALGORITHMS

ALGORITHM 1	– K2 Algorithm	42
ALGORITHM 2	– BN-HC Algorithm	43
ALGORITHM 3	– HMOBEDA Framework	57
ALGORITHM 4	– Local Search (LS)	59
ALGORITHM 5	– Bit-flip Neighborhood	60
ALGORITHM 6	– Set Neighborhood	60
ALGORITHM 7	– NSGA-II	116
ALGORITHM 8	– NSGA-III	119
ALGORITHM 9	– MOEA/D	123

LIST OF ACRONYMS

AffEDA	Affinity Propagation EDA
ALS	Adaptive Local Search
BI	Boundary Intersection
BKP	Bounded Knapsack Problem
BMDA	Bivariate Marginal Distribution Algorithm
BMOA	Bayesian Multi-objective Optimization Algorithm
BN	Bayesian Network
BN-HC	Bayesian Network Greedy Hill Climber Algorithm
BOA	Bayesian Optimization Algorithm
CD	Crowding Distance
GA	Genetic Algorithm
cGA	Compact Genetic Algorithm
COMIT	Combining Optimizer with Mutual Information Tree
DAG	Directed Acyclic Graph
DR	Dominance Rank
EA	Evolutionary Algorithm
ER	Error Ratio
EBNA	Estimation of Bayesian Network Algorithm
ECGA	Extended Compact Genetic Algorithm
EDA	Estimation of Distribution Algorithm
FDA	Factorized Distribution Algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
hBOA	Hierarchical Bayesian Optimization Algorithm
HC	Hill Climbing
HMEA	Hybrid Multi-objective Evolutionary Algorithm
HMOBEDA	Hybrid Multi-objective Bayesian Estimation of Distribution Algorithms
HypE	Fast Hypervolume-based Many-objective Optimization
IBMOLS	Iterated Local Search based on R2 Quality Indicator
IGD	Inverted Generational Distance
ILS	Iterated Local Search
IMMOGLS	Ishibuchi's and Murata's Multiple-objective Genetic Local Search
MaOP	Many Objective Optimization Problem
MBN	Multidimensional Bayesian Network
MIMIC	Mutual Information Maximization for Input Clustering
MLE	Maximum Likelihood Estimate
MNK	Multi-objective NK
MOA	Markovianity-based Optimization Algorithm
MOEA/D	Multi-objective Evolutionary Algorithm Based on Decomposition
MOEA	Multi-objective Evolutionary Algorithm
MOEDA	Multi-objective Estimation of Distribution Algorithm
MOGLS	Multiple-objective Genetic Local Search

MOHEDA	Hybrid Estimation of Distribution Algorithm
MOKP	Multi-objective Knapsack Problem
MoMad	Memetic Algorithm for Combinatorial Multi-objective
MOP	Multi-objective Optimization Problem
M-PAES	Memetic-Pareto Archive Evolution Strategy
NPGA	Niched Pareto Genetic Algorithm
NSGA	Non-dominated Sorting in Genetic Algorithm
ONVG	Overall Non-dominated Vector Generation
PBIL	Population-based Incremental Learning
PESA	Pareto Envelope-Based Selection Algorithm
PGM	Probabilistic Graphical Model
PLS	Pareto Local Search
pmf	probability mass function
S-MOGLS	Simple Multi-Objective Genetic Local Search Algorithm
SMS-EMOA	Multi-objective selection based on dominated hypervolume
SPEA	Strength Pareto Evolutionary Algorithm
TTP	Travelling Thief Problem
UMDA	Univariate Marginal Distribution Algorithm

LIST OF VARIABLES

AP	Approximated Pareto-front
a_{rq}	Profit of item $q = 1, \dots, Q$, according to knapsack $r = 1, \dots, R$
b_{rq}	Weight of item q according to knapsack r
\mathcal{B}	Set of binary numbers
c_r	Constraint capacity of knapsack r
$f(\mathbf{x})$	Mono-objective function
$E(\theta_{mjk} \mathbf{N}_{mj}, B)$	Expected value of θ_{mjk} given Bayesian Estimate
$\boldsymbol{\lambda}$	Non-negative weight vector
Max_{eval}	Maximum number of solutions evaluation
Max_{eval}	Maximum number of solutions evaluation
N	Population size
N_{best}	Maximum of parents selected to provide the offspring population
N_{iter}	Maximum number of iterations of the considered local search
N_{ls}	Total number of neighbors
N_{mj}	Number of individuals in Pop with the parents of Y_m are instantiated to its j -th combination
N_{mjk}	Number of observations for which Y_m assumes the k -th value given the j -th combination of values from its parents
N_{PGM}	Number of selected individuals to compose Pop_{PGM}
ND	Set of non-dominated solutions in the variable space
\mathbf{p}	Vector of elements associated with the local search parameters
P_{ls}	Probability of occurrence of Local Search
Pop^g	Population of generation g
Pop_{merged}^g	Joint of current population and offspring population
Pop_{best}^g	Population of selected individuals from generation g
Pop_I	Population of improved individuals
$Pop_{offspring}^g$	Population generated by genetic operations
Pop_{PGM}	Population of N_{PGM} individuals to support the PGM
Pop_{smp}	Population of Sampled individuals
Q	Decision variable vector dimension
R	Number of objectives
Ref	Reference front
\mathcal{R}	Real numbers set
$\hat{\theta}_{mjk}$	Maximum Likelihood Estimate parameter for θ_{mjk}
T_{Fnbh}	Function type used by local search to compute the neighbor fitness
T_{nbh}	Local Search neighborhood type: drop-add (DAd) or insert(Ins)
Tot_F	Total number of Pareto fronts
\mathbf{x}	Decision variable vector
X	Decision variable space
\mathbf{z}	Vector of objective functions
Z	Objective Space

z^{bi}	Boundary Intersection scalarizing function
z^{te}	Tchebycheff scalarizing function
z^{ws}	Weighted sum scalarizing function

CONTENTS

1 INTRODUCTION	13
1.1 OBJECTIVES	15
1.2 CONTRIBUTIONS	15
1.3 ORGANIZATION	19
2 MULTI-OBJECTIVE OPTIMIZATION	20
2.1 BASIC CONCEPTS	20
2.2 PARETO DOMINANCE-BASED APPROACHES	22
2.3 SCALARIZING FUNCTION-BASED METHODS AND OTHERS APPROACHES	23
2.3.1 Weighted Sum approach	24
2.3.2 Tchebycheff approach	25
2.3.3 Boundary Intersection approach	25
2.4 SOLUTION QUALITY INDICATORS	26
2.4.1 Hypervolume Indicator - HV	30
2.4.2 Inverted Generational Distance - IGD	30
2.4.3 Capacity metrics	30
2.5 SUMMARY	31
3 ESTIMATION OF DISTRIBUTION ALGORITHM - EDA	32
3.1 BASIC CONCEPTS	33
3.2 CLASSIFICATION	34
3.3 BAYESIAN NETWORK	35
3.3.1 Basic concepts	35
3.3.2 Parameter estimation	36
3.3.3 Structure	39
3.3.4 Naive BN	40
3.4 SUMMARY	44
4 THE MULTI-OBJECTIVE PROBLEMS ADDRESSED IN THIS WORK	45
4.1 THE MULTI-OBJECTIVE KNAPSACK PROBLEM	45
4.1.1 General formulation	45
4.1.2 MOEA to solve MOKP	47
4.1.3 EDA approaches for MOKP	49
4.2 THE MULTI-OBJECTIVE NK-LANDSCAPE PROBLEM	51
4.2.1 General formulation	51
4.2.2 MOEA approaches for MNK-landscape	53
4.3 SUMMARY	54
5 HYBRID MULTI-OBJECTIVE BAYESIAN ESTIMATION OF DISTRIBUTION ALGORITHM	55
5.1 GENERAL SCHEME	55
5.2 SOLUTION ENCODING	56
5.3 HMOBEDA FRAMEWORK	57
5.3.1 Initialization	58
5.3.2 Non-dominated sorting	58

5.3.3	Local search	58
5.3.3.1	Bit-flip and Set Mechanisms	59
5.3.4	Probabilistic model	60
5.4	DIFFERENCES FROM THE LITERATURE	62
5.5	SUMMARY	64
6	EXPERIMENTS AND RESULTS	65
6.1	SETUP OF EXPERIMENTS	66
6.1.1	Problem instances	66
6.1.2	HMOBEDA alternative versions	66
6.1.3	Cutting edge evolutionary approaches from the literature	70
6.2	RESULTS	71
6.2.1	Comparing alternative versions of HMOBEDA	72
6.2.1.1	BN Structure Estimation: HMOBEDA _{K2} x HMOBEDA _{BN-HC}	72
6.2.1.2	Tie-breaker criterion: HMOBEDA _{CD} x HMOBEDA _{hype}	73
6.2.1.3	LS parameter tuning: Online x Off-line versions	74
6.2.1.4	Setting evidence: HMOBEDA _{IDEAL} x HMOBEDA _{EXT} x HMOBEDA _{CPT}	76
6.2.2	Comparing HMOBEDA _{EXT} with cutting edge approaches	81
6.2.2.1	Experiments with MOKP	82
6.2.2.2	Experiments with MNK-landscape problem	83
6.2.3	Analyzing the Probabilistic Graphic Model	88
6.3	SUMMARY	92
7	CONCLUSION	94
7.1	FUTURE WORK	95
	REFERENCES	97
	Appendix A – EXAMPLE OF THE K2 ALGORITHM	109
	Appendix B – CUTTING EDGE EVOLUTIONARY ALGORITHMS	115
B.1	NSGA-II	115
B.2	S-MOGLS	117
B.2.1	S-MOGLS Local Search Procedure	117
B.3	NSGA-III	118
B.4	MBN-EDA	121
B.5	MOEA/D	122

1 INTRODUCTION

In many optimization problems, maximizing/minimizing two or more objective functions represents a challenge to a large number of optimizers (LUQUE, 2015). This class of problems is known as Multi-objective Optimization Problems (MOP), and solving MOPs has thus been established as an important field of research (DEB, 2001). In the past few years, problems with more than three objectives are becoming usual. They are referred as Many Objective Optimization Problems (MaOP) (ISHIBUCHI et al., 2008).

MOPs and specially 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 (DEB, 2001).

Evolutionary Algorithms (EA) and other population-based metaheuristics have been widely used for solving multi and many objective optimization, mainly due to their ability to find multiple solutions in parallel and to handle the complex features of such problems (COELLO, 1999). For combinatorial optimization, local optimizers can also be aggregated to capture and exploit the potential regularities that arise in the promising solutions.

Several Multi-objective Evolutionary Algorithms (MOEA) incorporating local search (LS) have been investigated, and these hybrid approaches can often achieve good performance for many problems (LARA et al., 2010; ZHOU et al., 2011, 2015). However, as discussed in Martins et al. (2016) and Martins et al. (2017a), 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.

Frequency and intensity directly influence the degree of exploration versus exploitation in these hybrid approaches (KRASNOGOR; GUSTAFSON, 2002). Clearly, a more intense procedure increases chances for converging into a local optimum nevertheless it limits a suitable search space exploration. Therefore, unless we work with unrestricted budget, special attention

is necessary when setting these two parameters. Moreover, when only a subset of individuals undergo local optimization, their choice becomes an issue. Finally, the type of LS favors different neighborhood structures. All these parameters affect the algorithm's performance making auto-adaptation an important topic for Hybrid EAs and MOEAs.

Another strategy widely used in evolutionary optimization is the probabilistic modeling, which is the basis of an Estimation of Distribution Algorithm (EDA) (MÜHLENBEIN; PAAB, 1996). The main idea of EDAs (LARRAÑAGA; LOZANO, 2002) is to extract and represent, using a probabilistic model, the regularities shared by a subset of high-valued problem solutions. New solutions are then sampled from the probabilistic model guiding the search toward areas where optimal solutions are more likely to be found. Normally, a Multi-objective Estimation of Distribution Algorithm (MOEDA) (KARSHENAS et al., 2014) integrates both model building and sampling techniques into evolutionary multi-objective optimizers using special selection schemes (LAUMANNNS; OCENASEK, 2002). A promising probabilistic model, called Probabilistic Graphical Model (PGM) that combines graph and probability theory, has been adopted to improve EDAs and MOEDAs performance (LARRAÑAGA et al., 2012). Most of MOEDAs developed to deal with combinatorial MOPs adopt Bayesian Networks as their PGM.

In this work, we propose a different approach called Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA) based on a joint probabilistic modeling of decision variables, objectives, and parameters of a local optimizer. Some recent MOEDA-based approaches model a joint distribution of variables and objectives in order to explore their relationship, which means investigating how objectives influence variables and vice versa. However, our approach also includes an LS parameter tuning in the same model. The rationale of HMOBEDA is that the embedded PGM can be structured to sample appropriate LS parameters for different configurations of decision variables and objective values.

The proposed approach is evaluated by means of 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 (ISHIBUCHI et al., 2015; KE et al., 2014; TAN; JIAO, 2013; TANIGAKI et al., 2014; VIANNA; VIANNA, 2013). In particular, MOEDAs that use different types of probabilistic models have already been applied to MOKP (LI et al., 2004; WANG et al., 2012), especially those based on Bayesian Networks (BN) (LAUMANNNS; OCENASEK, 2002; SCHWARZ; OCENASEK, 2001a). We also extend our analysis to the multi-objective NK-landscape (MNK-landscape) problem (AGUIRRE; TANAKA, 2004, 2007), another multi-objective combinatorial optimization problem that has

been recently explored by Aguirre and Tanaka (2007), Verel et al. (2011), Santana et al. (2015b), Daolio et al. (2015), including works for the mono-objective version using EDAs (PELIKAN, 2008; PELIKAN et al., 2009; LIAW; TING, 2013). However, these works do not consider objectives and LS parameters structured all together in the same BN, as proposed in this work.

1.1 OBJECTIVES

This work aims to provide a new hybrid MOEDA-based approach named HMOBEDA (Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm) to solve multi and many objective combinatorial optimization problems with automatic setting of LS mechanisms (and their associated parameters) suitable to each stage of the evolutionary process.

The specific objectives include:

1. To implement this new hybrid MOEDA-based approach whose automatic LS setting is provided by a Bayesian Network;
2. To develop a framework capable of learning a joint probabilistic model of objectives, variables and local search parameters, all together;
3. To evaluate several variants in order to investigate the Bayesian Network learning structure, the tie-breaker criterion from the selection scheme, the online versus off-line versions of LS parameter tuning, and the different ways to set the evidences during the sampling, in the context of HMOBEDA;
4. To analyze the Pareto front approximation provided by HMOBEDA from a probabilistic point of view;
5. To scrutinize information present in the probabilistic model, analyzing the relations between objectives, variables and local search parameters;
6. To compare HMOBEDA with other recent approaches for solving instances of MOKP and MNK-Landscape problems.

1.2 CONTRIBUTIONS

Although other hybrid MOEDAs have been developed using joint probabilistic modeling of decision variables and objectives, none of them has considered local search parameters in the same model. Thus, this work proposes a joint probabilistic modeling of (i)

local search with (ii) decision variables and (iii) objectives, in a framework named HMOBEDA. The advantage is that a probabilistic model can learn local optimization parameters for different configurations of decision variables and objective values at different stages of the evolutionary process. Our main contribution refers to the inclusion of local search parameters in the probabilistic model providing an auto-adapting parameter tuning approach.

Due the fact that the PGM structure learning is an important process of our proposal, we investigate two methods: a Hill-Climbing technique (BN-HC Algorithm) and the K2 Algorithm, providing two versions: HMOBEDA_{BN-HC} and HMOBEDA_{K2} . An important criterion for the evolutionary approaches is the tie-breaker for the selection procedure. In order to evaluate both crowding distance and hypervolume strategies, we define two variants of HMOBEDA : HMOBEDA_{CD} and HMOBEDA_{hype} .

Another gap observed in the literature concerns the test and analysis of parameter tuning for hybrid MOEDAs accomplished before versus during optimization (named off-line and online parameter tuning). In this context HMOBEDA, which is considered an online parameter tuner, is modified into three other variants with LS parameters being pre-determined and kept fixed during the search (off-line configuration): HMOBEDA_f , HMOBEDA_{f-inst} and HMOBEDA_{irace} . HMOBEDA_f considers the most frequent LS-parameters achieved by the original HMOBEDA in non-dominated solutions, of all instances and executions. HMOBEDA_{f-inst} considers the most frequent LS parameters found in all HMOBEDA executions by instance. HMOBEDA_{irace} considers the parameters tuned by I/F-Race (BIRATTARI et al., 2010).

Since providing relationships among solution variables is one of the advantages of using EDA, this work scrutinizes information present in the final PGM. We explore, from a probabilistic point of view, the approximated Pareto front using the final PGM structure. Three versions, which use different sampling techniques, are tested: HMOBEDA_{IDEAL} , HMOBEDA_{CPT} and HMOBEDA_{EXT} . These versions represent different possibilities to guide the search. HMOBEDA_{IDEAL} considers evidences fixed as maximum values for all objectives (i.e. an estimated solution named ideal point). HMOBEDA_{CPT} considers evidences fixed according to the parameters and probabilities estimated from the Conditional Probability Table (CPT). HMOBEDA_{EXT} considers evidences fixed as the ideal point and combinations of maximum and minimum values for the objectives, all of them with the same probability of occurrence.

After the comparison between the HMOBEDA variants on the MOKP instances, a standard version is defined and compared with algorithms in the literature often used

to solve MOPs: MBN-EDA (KARSHENAS et al., 2014), NSGA-II (DEB et al., 2002), S-MOGLS (ISHIBUCHI et al., 2008) (NSGA-II with local search), MOEA/D (ZHANG; LI, 2007) and NSGA-III (DEB; JAIN, 2014). This analysis points out HMOBEDA's performance using not only hypervolume (HV^-) but also Inverted Generational Distance (IGD) and capacity indicators to provide a more complete picture of HMOBEDA behavior considering a set of MOKP instances with 2 to 5 and 8 objectives, and of MNK-landscape instances with 2, 3, 5 and 8 objectives. Besides, we present the influence that objectives have on variables, from the analysis of how frequently objective-variable interactions occur. After that we determine how sensitive is the influence of objectives on LS parameters from the analysis of the frequency of the objective-parameters interactions. Few works consider MOEDAs to solve MOKP (SCHWARZ; OCENASEK, 2001a, 2001b; LI et al., 2004) and as far as we know, this is the first work using a PGM as part of a Hybrid MOEDA in the MOKP context and the first work using MOEDA to solve the MNK-landscape problem.

Results of this work and associated research were published in the following journal and conference papers.

MARTINS, M. S.; DELGADO, M. R.; SANTANA, R.; LÜDERS, R.; GONÇALVES, R. A.; ALMEIDA, C. P. d. HMOBEDA: Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm. In: **Proceedings of the Genetic and Evolutionary Computation Conference**. New York, NY, USA: ACM, 2016. (GECCO'16), p. 357–364. ISBN 978-1-4503-4206-3.

In this paper, the new approach HMOBEDA is firstly presented and compared with six evolutionary methods including a discrete version of NSGA-III, using instances of the MOKP with 3, 4, and 5 objectives. Results have shown that HMOBEDA is a competitive approach that outperforms the other methods according to hypervolume indicator.

MARTINS, M. S. R.; DELGADO, M. R. B. S.; LÜDERS, R.; SANTANA, R.; GONÇALVES, R. A.; ALMEIDA, C. P. de. Hybrid multi-objective Bayesian estimation of distribution algorithm: a comparative analysis for the multi-objective knapsack problem. **Journal of Heuristics**, Sep 2017. p. 1–23.

Here, an additional investigation of off-line versions of LS parameter tuning was published, comparing $HMOBEDA_f$, $HMOBEDA_{f_{inst}}$ and $HMOBEDA_{irace}$ with HMOBEDA (the online version). This paper also compared HMOBEDA with traditional approaches for multi and many objective optimization based on Inverted Generational Distance (IGD) quality indicator in addition to the hypervolume metric. Besides, an analysis of the resulting BN structures has also been carried out to evaluate how the interactions among

variables, objectives and local search parameters are captured by the BNs. The results showed that probabilistic modeling arises as a significant and feasible way not only to learn and explore dependencies between variables and objectives but also to automatically control the application of local search operators.

MARTINS, M. S. R.; DELGADO, M. R. B. S.; LÜDERS, R.; SANTANA, R.; GONÇALVES, R. A.; ALMEIDA, C. P. de. 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**. Uberlândia, (BRACIS'17). p. 1–8.

This paper explored, from a probabilistic point of view, the approximated Pareto front using the final PGM structure. Two variants of HMOBEDA that use different sampling techniques were then proposed: HMOBEDA_{CPT} and HMOBEDA_{EXT}. Results concluded that uniform distribution of evidences among ideal and extreme points of the Pareto front (EXT version) in the sample process is beneficial for HMOBEDA .

Although this thesis addresses MOEDAs-based multi-optimization techniques, in the next two papers we explored a relatively new single-objective NP-hard benchmark problem - Travelling Thief Problem (TTP) - in a hyperheuristic context using Genetic Programming (GP) and EDA.

YAFRANI, M. E.; MARTINS, M.; WAGNER, M.; AHIOD, B.; DELGADO, M.; LÜDERS, R. A hyperheuristic approach based on low-level heuristics for the travelling thief problem. **Genetic Programming and Evolvable Machines**, Jul 2017. p. 1–30.

MARTINS, M. S. R.; YAFRANI, M. E.; DELGADO, M. R. B. S.; WAGNER, M.; AHIOD, B.; LÜDERS, R. HSEDA: A heuristic selection approach based on estimation of distribution algorithm for the travelling thief problem. In: **Proceedings of the Genetic and Evolutionary Computation Conference**. New York, NY, USA: ACM, (GECCO '17), p. 361–368. ISBN 978-1-4503-4920-8.

Both approaches used well known low-level-heuristics in order to evolve combinations of these heuristics aiming to find a good model for the instance at hand. The main contributions of the last two studies are in the PGM used in hyperheuristic way and an attempt to work with TTP problem to be further extended to a multi-objective context, exploring other probabilistic models which are able to represent permutation problems as well.

1.3 ORGANIZATION

This work is organized as follows. After this introduction, Chapters 2 and 3 present an overview of multi-objective optimization concepts and estimation of distribution algorithms, respectively. In Chapter 4, the MOKP and the MNK-landscape problems considered in this work are discussed, in addition some previous work and contributions in the context of MOEAs and EDAs are presented. Chapter 5 details the proposed approach, presenting the encoding scheme and the developed framework and algorithm (HMOBEDA). Computational experiments, results and discussions using instances of MOKP and MNK-landscape problem are reported in Chapter 6 including a PGM analysis. Chapter 7 concludes this thesis and presents new directions to future work.

2 MULTI-OBJECTIVE OPTIMIZATION

Real-world problems are generally characterized by several competing objectives. While in the case of single-objective optimization one optimal solution is usually required to solve the problem, this is not true in multi-objective optimization. The standard approach to solve this difficulty lies in finding all possible trade-offs among the multiple, competing objectives. MOP (as a rule) presents a possibly uncountable set of solutions, which when evaluated, produce vectors whose components represent trade-offs in the objective space. In the past few years, MOPs with more than three objectives became a trend in the field of multi-objective optimization, being a field by its own referred to as many objective optimization. Once a set of Pareto solutions is determined, a decision maker can implicitly choose an acceptable solution (or solutions) by selecting one or more of these vectors.

This chapter introduces the multi-objective optimization basic concepts in Section 2.1. Pareto dominance-based, Scalarizing function-based and others approaches are discussed in Sections 2.2 and 2.3. Finally, Section 2.4 discusses the statistical tests usually applied to compare multi-objective algorithms.

2.1 BASIC CONCEPTS

A general MOP includes decision variables, objective functions, and constraints, where objective functions and constraints are functions of the decision variables (ZITZLER; THIELE, 1999; SCHWARZ; OCENASEK, 2001b). Mathematically, a maximization MOP can be defined as:

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

where $\mathbf{x} = (x_1, \dots, x_Q)$ is a Q -dimensional decision variable vector defined in a universe X ; \mathbf{z} is the objective vector, with R objectives, where each $f_r(\mathbf{x})$ is a single-objective function, Z is the objective space and $\mathbf{h}(\mathbf{x}) \leq 0$ is the set of constraints which determines a set of feasible solutions X_f .

The set of MOP solutions includes decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another - these decision vectors are called Pareto optimal set. The idea of Pareto optimality is based on the Pareto dominance.

For any two decision vectors \mathbf{u}, \mathbf{v} it holds

\mathbf{u} dominates \mathbf{v} iff $\mathbf{f}(\mathbf{u}) > \mathbf{f}(\mathbf{v})$,

\mathbf{u} weakly dominates \mathbf{v} iff $\mathbf{f}(\mathbf{u}) \geq \mathbf{f}(\mathbf{v})$,

\mathbf{u} is indifferent to \mathbf{v} iff \mathbf{u} and \mathbf{v} are not comparable.

A decision vector \mathbf{u} dominates a decision vector \mathbf{v} iff $f_r(\mathbf{u}) \geq f_r(\mathbf{v})$ for $r = 1, 2, \dots, R$ with $f_r(\mathbf{u}) > f_r(\mathbf{v})$ for at least one r . 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 . The set of non-dominated solutions lies, in the objective space, on a surface known as Pareto optimal front. The goal of the optimization is to find a representative set of solutions with the corresponding objective vectors along the Pareto optimal front.

Examples of the concept of Pareto dominance and Pareto front are depicted in a graphical form in Figure 1.

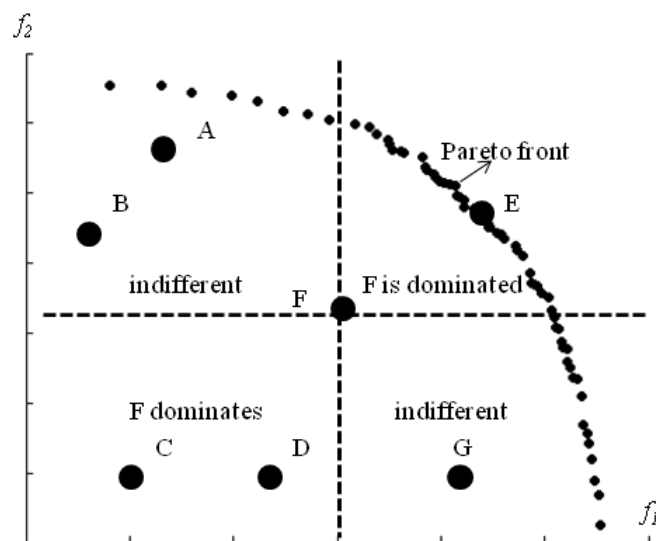


Figure 1: An example of Pareto front and Pareto dominance in the objective space.

It can be noticed that solution F dominates solution C and D , solution F is dominated by E , F is not comparable with A , B and G , solution E is non-dominated and Pareto optimal. Note that a solution is a decision vector that has its Pareto dominance criteria based on the objective space (objective vector corresponding values).

Generating the Pareto set can be computationally expensive and it is often infeasible, because most of the time the complexity of underlying application prevents exact methods from being applicable. For this reason, a number of stochastic search strategies such as EAs, tabu search, simulated annealing, and ant colony optimization have been developed: they usually do not guarantee the identification of optimal trade-offs instead they try to find a good approximation, i.e., a set of solutions whose objective vectors are not too far from the Pareto optimal front (ZITZLER et al., 2004). Due to their population-based nature, EAs are able to approximate the whole Pareto front of a MOP in a single run, and these EAs are called MOEAs (ZHOU et al., 2011).

MOEAs have been well established as effective approaches to deal with MOPs (JIANG et al., 2015). After multiple trade-off and non-dominated points are found, higher-level information can be used to choose one of the obtained trade-off points (DEB, 2001).

Based on various acceptance rules to accomplish fitness assignment and selection to guide the search toward the Pareto-optimal set, and to maintain a diverse population achieving a well distributed Pareto front, classical MOEAs can be generally divided in Pareto dominance-based approaches, Scalarizing function-based methods and other approaches (JIANG et al., 2015).

2.2 PARETO DOMINANCE-BASED APPROACHES

There are many papers presenting various approaches to find a Pareto front, most of them based on classical MOEAs. The balance between convergence and diversity is the most important aspect while solving a MOP. Unlike single-objective-optimization, fitness calculation in MOPs is usually related to the whole population. The multi-objective optimization is a typical multimodal search aiming to find multiple different solutions in a single run.

The main representatives of the Pareto optimization algorithms will be shortly mentioned. The Niche Pareto Genetic Algorithm (NPGA) combines tournament selection and the concept of Pareto dominance (HORN et al., 1994). An interesting approach using Non-dominated Sorting in Genetic Algorithm (NSGA) was proposed by Srinivas and Deb (1994). This technique calculates the Dominance Rank (DR) (ZITZLER et al., 2000),

sorting 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). Each set F_i , $i = 2, \dots, Tot_F$, contains the non-dominated solutions when sets F_1, \dots, F_{i-1} are removed from the current population.

A wide review of basic approaches and the specification of original Pareto evolutionary algorithms includes the works of Coello (1996) and Zitzler and Thiele (1999), where the last one describes the original Strength Pareto Evolutionary Algorithm (SPEA). An extension of the SPEA algorithm resulted in SPEA2 (ZITZLER et al., 2001) and Pareto Envelope-Based Selection Algorithm (PESA) (CORNE et al., 2000). This algorithm was updated to Pareto Envelope-Based Selection Algorithm II (CORNE et al., 2001), where the authors described a grid-based fitness assignment strategy in environmental selection. Another improved version, called IPESA-II (LI et al., 2013) introduced three improvements in environmental selection, regarding the performance: convergence, uniformity, and extensivity.

One of the most popular approaches based on Pareto dominance is NSGA-II (DEB et al., 2002). Recently, its reference-point based variant, referred as NSGA-III (DEB; JAIN, 2014) was suggested to deal with many-objective problems, where the maintenance of diversity among population members is aided by supplying and adaptively updating a number of well-spread reference points.

We can point out the fact that Pareto approaches simultaneously consider all the objectives - every point/solution of the Pareto front is part of the set of solutions - and maintain the diversity of solutions. However two main disadvantages are that these approaches are computationally expensive and they are not very intuitive when the number of objectives is large. All the previously mentioned algorithms evolve toward the Pareto set with a good distribution of solutions but none of them guarantees the convergence to Pareto front.

2.3 SCALARIZING FUNCTION-BASED METHODS AND OTHERS APPROACHES

In addition to the Pareto-dominance based methods, there are many others developed for multi-criteria optimization, mostly based on the scalarization of the objective functions (TRIVEDI et al., 2017). This way the MOP can be easily transformed into simpler single-objective problem. Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) (ZHANG; LI, 2007), Multi-objective selection based on dominated hypervolume (SMS-EMOA) (BEUME et al., 2007) and the algorithm for Fast Hypervolume-based Many-objective Optimization (HypE) (BADER, 2009) are examples of these approaches.

SMS-EMOA is a hypervolume-based algorithm. Its high search ability for many-objective problems has been demonstrated in the literature (WAGNER et al., 2007). The basic idea of SMS-EMOA is to search for a solution set with the maximum hypervolume for its corresponding objective vectors. In SMS-EMOA, two parents are randomly selected from a current population of size N to generate a single solution by crossover and mutation. The next population with N solutions is constructed by removing the worst solution from the merged population with $(N + 1)$ solutions. In the same manner as in NSGA-II, a rank is assigned to each solution in the merged population as the primary criterion to select individuals. Each solution with the same rank is evaluated by its hypervolume contribution as the secondary criterion.

In HypE, as in NSGA-II, first the non-dominated sorting is applied and then a criterion is used to select individuals among those with the same rank. The main difference is that the hypervolume maximization is used as the secondary criterion in environmental selection. Solution selection for the hypervolume maximization is performed in an approximate manner using Monte Carlo simulation given a number of sampling points.

MOEA/D is an efficient scalarizing function-based algorithm. A multi-objective problem is decomposed into a number of single-objective problems. Each single-objective problem is defined by the same scalarizing function with a different weight vector. The number of the weight vectors is the same as the number of the single-objective problems, which is also the same as the population size. A single solution is stored for each single-objective problem (ZHANG; LI, 2007).

There are several approaches for converting the problem of approximation of the Pareto front into a number of scalar optimization problems. In the following, we introduce three of them.

2.3.1 WEIGHTED SUM APPROACH

This approach considers a combination of the different objectives function $f_1(\mathbf{x}), \dots, f_R(\mathbf{x})$. Let $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_R)^T$ be a weight vector, i.e. $\lambda_r \geq 0$, for all $r = 1, \dots, R$ where R is the number of objectives and $\sum_{r=1}^R \lambda_r = 1$. Then, the optimal solution to the following scalar optimization problem:

$$\begin{aligned} & \text{maximize} && z^{ws}(\mathbf{x}|\boldsymbol{\lambda}) = \sum_{r=1}^R \lambda_r f_r(\mathbf{x}), \\ & \text{subject to} && \mathbf{x} \in X \end{aligned} \tag{2}$$

$z^{ws}(\mathbf{x}|\boldsymbol{\lambda})$ is a Pareto optimal point to Equation 1¹, $\boldsymbol{\lambda}$ is a weight vector in the z^{ws} objective function, while \mathbf{x} is the vector of variables to be optimized. To generate a set of different Pareto optimal vectors, we can use different weight vectors $\boldsymbol{\lambda}$ in this scalar optimization problem (MIETTINEN, 1999). If the Pareto front is concave (convex in the case of minimization), this approach could work well. However, not every Pareto optimal vector can be obtained by this approach in the case of nonconcave Pareto fronts (ZHANG; LI, 2007). To overcome these shortcomings, some effort has been made to incorporate other techniques such as ε -constraint into this approach, more details can be found in Miettinen (1999).

2.3.2 TCHEBYCHEFF APPROACH

Another well-known method is the Tchebycheff Approach, for which scalar optimization problem uses the z^{te} function in the form:

$$\begin{aligned} & \text{minimize } z^{te}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^*) = \max\{\lambda_r | f_r(\mathbf{x}) - z_r^* |\}, \\ & \text{subject to } \mathbf{x} \in X \end{aligned} \quad (3)$$

where $\mathbf{z}^* = (z_1^*, \dots, z_R^*)^T$ is the ideal point in the decision space, i.e., $z_r^* = \max\{f_r(\mathbf{x}) | \mathbf{x} \in X\}$ ² for each $r = 1, \dots, R$. For each Pareto optimal point \mathbf{x}^* there exists a weight vector $\boldsymbol{\lambda}$ such that \mathbf{x}^* is the optimal solution of Equation 3 and each optimal solution of Equation 3 is a Pareto optimal solution of Equation 1. Therefore, the method is able to obtain different Pareto optimal solutions by altering the weight vector. One weakness with this approach is that this aggregation function is not smooth for a continuous MOP (ZHANG; LI, 2007).

2.3.3 BOUNDARY INTERSECTION APPROACH

Several recent MOP decomposition methods such as Normal-Boundary Intersection Method (DAS; DENNIS, 1998) and Normalized Normal Constraint Method (MESSAC et al., 2003) can be classified as Boundary Intersection (BI) approaches. They were designed for a continuous MOP. Under some regularity conditions, the Pareto front of a continuous MOP is part of the most top right³ boundary of its attainable objective set. Geometrically, these BI approaches aim to find intersection points of the most top boundary and a set of lines. If these lines are evenly distributed in a sense, one can expect that the resultant intersection points provide a good approximation to the whole Pareto front. These approaches are able to deal

¹If Equation 1 is for minimization, "maximize" in 2 should be changed to "minimize".

²In the case of minimization, $z_r^* = \min\{f_r(\mathbf{x}) | \mathbf{x} \in X\}$

³In the case of minimization, it will be part of the most left bottom.

with nonconcave Pareto fronts. Mathematically, we consider the following scalar optimization subproblem with z^{bi} function ⁴:

$$\begin{aligned} & \text{minimize} && z^{bi}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^*) = d, \\ & \text{subject to} && \mathbf{z}^* - \mathbf{f}(\mathbf{x}) = d\boldsymbol{\lambda}, \\ & && \mathbf{x} \in X \end{aligned} \tag{4}$$

where $\boldsymbol{\lambda}$ and \mathbf{z}^* , as in the previous subsection, are a weight vector and the ideal point, respectively. As illustrated in Figure 2, the constraint $\mathbf{z}^* - \mathbf{f}(\mathbf{x}) = d\boldsymbol{\lambda}$ ensures that $\mathbf{f}(\mathbf{x})$ is always in L , the line with direction $\boldsymbol{\lambda}$ and passing through \mathbf{z}^* . The goal is to push $\mathbf{f}(\mathbf{x})$ as high as possible so that it reaches the boundary of the attainable objective set (ZHANG; LI, 2007).

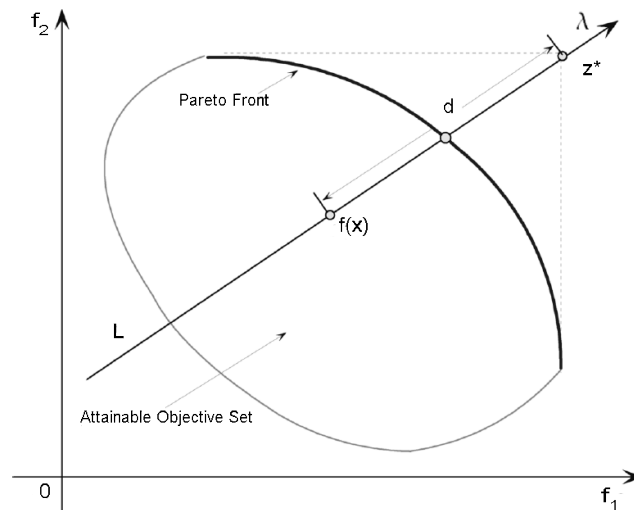


Figure 2: Illustration of the boundary intersection approach (ZHANG; LI, 2007).

Appendix B contains some cutting edge detailed algorithms implemented in this work for further comparison.

2.4 SOLUTION QUALITY INDICATORS

Comparing different optimization techniques always involves the notion of performance. In the case of multi-objective optimization, the definition of solution quality is substantially more complex than for single-objective optimization problems, because the optimization goal itself consists of multiple objectives (ZITZLER et al., 2000):

- The distance of the resulting approximated Pareto-front to the Pareto optimal front should

⁴In the case of minimization, this equality constraint in this subproblem should be changed to $f(\mathbf{x}) - \mathbf{z}^* = d\boldsymbol{\lambda}$

be minimized (i.e., convergence);

- A good (in most cases uniform) distribution of the objective vectors found in the objective space is desirable. The assessment of this criterion might be based on a certain distance metric;
- The spread of the approximated Pareto-front should be maximized along the Pareto optimal front (i.e., diversity).

According to Martí-Orosa (2011), the stochastic nature of EAs prompts the use of statistical tools in order to reach a valid judgement of the solutions quality and how different algorithms compare with each other. The straightforward approach to experimental design is to run the algorithm for a given number of independent executions and then extract descriptive statistics of the performance indicators. These statistic measures can be used to support a given hypothesis.

A statistical hypothesis testing is used to determine significant differences between algorithms. This test generates the probability, p -value, of supporting a null hypothesis according to a threshold probability: the significance level α . Regarding MOEA testing, it is desired to evaluate MOEAs by approximated Pareto sets.

There is a rather broad set of hypothesis test techniques. They can be grouped in parametric and non-parametric tests. According to García et al. (2008), in order to use the parametric tests it is necessary to check the following conditions:

- Independence: in statistics, two events are independent when the fact that one occurs does not modify the probability of the other one occurring;
- Normality: an observation is normal when its behaviour follows a normal or Gaussian distribution with a certain value of average μ and variance σ^2 . A normality test applied over a sample can indicate the presence or absence of this condition in observed data. García et al. (2008) proposes three normality tests:
 - Kolmogorov-Smirnov: it compares the accumulated distribution of observed data with the accumulated distribution expected from a Gaussian distribution, obtaining the p -value based on both discrepancies.
 - Shapiro-Wilk: it analyzes the observed data to compute the level of symmetry and kurtosis (shape of the curve) in order to compute the difference with respect to a Gaussian distribution afterwards, obtaining the p -value from the sum of the squares of these discrepancies.

- D’Agostino-Pearson: it first computes the skewness and kurtosis to quantify how far from Gaussian the distribution is in terms of asymmetry and shape. It then calculates how far each of these values differs from the value expected with a Gaussian distribution, and computes a single p-value from the sum of these discrepancies.
- Heteroscedasticity: this property indicates the existence of a violation of the hypothesis of equality of variances. Levene’s test (LEVENE, 1960) can be used for checking whether or not k samples present this homogeneity of variances (homoscedasticity).

Non-parametric tests can be used for comparing algorithms whose results represent average values for each problem. Given that the non-parametric tests do not require explicit conditions for being conducted, it is recommendable that the sample of results is obtained following the same criteria, that is, computing the same aggregation (average, mode, etc.) over the same number of runs for each algorithm and problem.

Non-parametric tests consist of two meta-level approaches: rank tests and permutation tests. Rank tests pool the values from several samples and convert them into ranks by sorting them, and then they build tables describing the limited number of ways in which ranks can be distributed (between two or more algorithms) to determine the probability that the samples come from the same source. Permutation tests use the original values without converting them to ranks but explicitly estimate the likelihood that samples come from the same source by Monte Carlo simulation. Rank tests are the less powerful but are also less sensitive to outliers and computationally inexpensive. Permutation tests are more powerful because information is not discarded, and they are also better when there are many tied values in the samples, however they can be expensive to compute for large samples (CONOVER, 1999).

There are many statistical tests for MOEA quality indicators that can be used when comparing if two or more algorithms are different (better or worse) from another (KNOWLES et al., 2006; COELLO et al., 2007; ZITZLER et al., 2008). One of the most common non-parametric tests is the general form of the Mann-Whitney test called Kruskal-Wallis (KRUSKAL; WALLIS, 1952) H test, where K independent samples can be compared. This statistical test can be performed for each comparable quality indicator using gathered experimental data (indicator values yielded by each algorithm’s run), and can determine if the samples are from the same population. This test is primarily used when no knowledge of the type of distribution is available. Expressing it more formally, for a set of algorithms A_1, \dots, A_K , each one running nr times (from a total of $Nr = nrK$) on the same problem, let $I_{k,j}$ be the indicator value yielded by algorithm k in run j . In a particular problem (or instance), the Kruskal-Wallis test goes on by sorting $I_{k,j}$, by relying on $\tilde{\text{Rank}}(I_{k,j})$, the ranking

function that returns the position of measurement $I_{k,j}$ in the list. Following that, the rank sum is calculated for each algorithm according to Equation 5:

$$\tilde{\text{Rank}}_k = \sum_{j=1}^{nr} \tilde{\text{Rank}}(I_{k,j}); k = 1, \dots, K \quad (5)$$

The definition of the Kruskal-Wallis H test is reflected in Equation 6.

$$H = \frac{12}{Nr(Nr+1)} \sum_{k=1}^K \frac{\tilde{\text{Rank}}_k^2}{Nr_k} - 3(Nr+1) \quad (6)$$

Given:

K sample of different sizes Nr_1, Nr_2, \dots, Nr_K , where $Nr = \sum_{k=1}^K Nr_k$. When $Nr_1, Nr_2, \dots, Nr_K = nr$ and $Nr = nrK$, all algorithms runs the same nr times.

Upon calculation of H using Equation 6, its value is treated as though it was a value of chi-square (χ^2) sampling distribution with the degrees of freedom (df) = $K - 1$, meaning that as the test statistic H is approximately χ^2 -distributed, the null hypothesis is reject if $H > \chi_{K-1;\alpha}^2$ (if H value is too great to fit in χ^2 distribution).

In case that the null hypothesis is rejected the Dunn-Sidak post-hoc test (CONOVER, 1999) can be applied in a pairwise manner in order to determine if the results of one algorithm are significantly better than those of the other. Dunn (1964) has proposed a test for multiple comparisons of rank sums based on the z statistics of the standard normal distribution and proved accurate by Sidak (HOCHBERG; TAMHANE, 1987). In particular, the difference of indicator values yielded by algorithms A_k and A_h is statistically significant if

$$|\tilde{\text{Rank}}_k - \tilde{\text{Rank}}_h| > z_{1-\alpha/2*} \sqrt{\left[\frac{Nr(Nr+1)}{12} \right] \left[\frac{1}{Nr_k} + \frac{1}{Nr_h} \right]} \quad (7)$$

with $z_{1-\alpha/2*}$ defined as the value of the standard normal distribution for a given adjusted $\alpha/2*$ level.

According to Martí-Orosa (2011), these statistical tests can be performed over some metrics values obtained by each algorithm's run. On this context, we adopted two principals convergence-diversity (JIANG et al., 2014) metrics that measure the quality of the solution set for MOP's: Hypervolume (HV) (ZITZLER; THIELE, 1999; BADER, 2009) and Inverted Generational Distance (IGD) (VELDHUIZEN; LAMONT, 1999; ZITZLER et al., 2003; LI; ZHANG, 2009).

2.4.1 HYPERVOLUME INDICATOR - HV

Hypervolume was introduced by Zitzler et al. (2003) and considers the size of the portion of the objective space that is dominated by the corresponding solutions region as an indicator of convergence (DEB, 2011).

Let Ref be a set of distributed points (objective vectors) in a reference set and AP the approximated Pareto-front in the objective space obtained from the set of non-dominated solutions ND in the variable space. $HV^*(.)$ is the portion of the objective space whose solutions are dominated by the corresponding solutions from the set defined in the function argument. It can be defined as:

$$HV^-(AP) = HV^*(Ref) - HV^*(AP) \quad (8)$$

Note that we consider the hypervolume difference to the Ref . So, smaller values of $HV^-(AP)$ correspond to higher quality solutions in non-dominated sets and they indicate both a better convergence as well a good coverage of the reference set (YAN et al., 2007; ARROYO et al., 2011).

2.4.2 INVERTED GENERATIONAL DISTANCE - IGD

The IGD is the average distance from each point on the reference set to the nearest point in the approximated set (in the objective space) (CZYZZAK; JASZKIEWICZ, 1998).

This metric measures both convergence and diversity (YEN; HE, 2014). Let Ref be a set of distributed points (objective vectors) in the reference set. The $IGD(AP)$ for the ND set is:

$$IGD(AP) = \frac{\sum_{v \in Ref} d(v, AP)}{|Ref|} \quad (9)$$

where $d(v, X)$, denotes the minimum Euclidean distance between point v in Ref and the points in AP . To have a low value of IGD, the set AP should be close to Ref and cannot miss any part of the whole Ref . The less the IGD, the better the algorithm's performance.

2.4.3 CAPACITY METRICS

Capacity metrics quantify the number or ratio of non-dominated solutions in an approximated set (in the objective space) (JIANG et al., 2014). An example is the Overall

Non-dominated Vector Generation (ONVG) (VELDHUIZEN; LAMONT, 2000), given by Equation 10:

$$ONVG = |AP| \quad (10)$$

In general, a large number of non-dominated solutions in AP is preferred (JIANG et al., 2014). However, counting the number of non-dominated solutions in AP does not reflect how far AP is from Ref (VELDHUIZEN; LAMONT, 2000).

The Error Ratio (ER) (VELDHUIZEN; LAMONT, 1999), also used as a capacity metric, considers the solution intersections between AP and Ref , given by Equation 11:

$$ER = 1 - \frac{|AP \cap Ref|}{|Ref|} \quad (11)$$

where $AP \cap Ref$ denotes the solutions existing in both AP and Ref .

In this work, these metrics are obtained considering the non-dominated solutions gathering all executions from each algorithm.

2.5 SUMMARY

In this chapter, we have introduced the multi-objective optimization, describing its basic concepts and a general formulation for a MOP. We have also presented a MOEA classification into Pareto dominance-based, Scalarizing function-based and other approaches, providing some representatives algorithms of each class. In order to compare different techniques, we have discussed some statistical tools for MOEA quality indicators. Moreover, we have presented the hypervolume indicator, the IGD and the capacity metrics, considered in this work. In the next chapter we aim to explore EDAs concepts and model building, concerning their application to MOPs.

3 ESTIMATION OF DISTRIBUTION ALGORITHM - EDA

Claimed as a paradigm shift in the field of EA, EDAs are population based optimization algorithms, which employ explicit probability distributions (LARRAÑAGA; LOZANO, 2002). This chapter provides a brief introduction to EDAs, followed by a description of Bayesian Network, which is considered the most prominent Probabilistic Graphical Model (LARRAÑAGA et al., 2012).

The main idea of EDAs (MÜHLENBEIN; PAAB, 1996; LARRAÑAGA; LOZANO, 2002) is to extract and represent, using a PGM, the regularities shared by a subset of high-valued problem solutions. The PGM then samples new solutions guiding the search toward more promising areas.

EDAs have achieved good performance when applied to several problems (PHAM, 2011) including environmental monitoring network design (KOLLAT et al., 2008), protein side chain placement problem (SANTANA et al., 2008) and table ordering (BENGOETXEA et al., 2011). In the context of Travelling Thief Problem (TTP), other probabilistic models have been recently explored in a hyperheuristic framework using Genetic Programming (GP) (EL YAFRANI et al., 2017) and EDA (MARTINS et al., 2017b). Both approaches use well known low-level-heuristics in order to evolve combinations of these heuristics aiming to find a good model for the TTP instance at hand.

EDAs have also been applied to solve multi-objective knapsack (SHAH; REED, 2011) and MOPs in a noisy environment (SHIM et al., 2013). Normally they integrate both, the model building and sampling techniques, into evolutionary multi-objective optimizers using special selection schemes (KHAN et al., 2002; LAUMANN; OCENASEK, 2002). However, recently the role of the probabilistic model has been extended to model the dependencies between variables and objectives (KARSHENAS et al., 2014). In addition, MOEDAs can be notably enhanced by adding a local optimizer that can refine the solutions found by sampling from the PGM (LI et al., 2004; WANG et al., 2012; ZHOU et al., 2015). In this work, we use this type of enhancement in the context of multi-objective optimization.

3.1 BASIC CONCEPTS

Let Y_m be a random variable. A possible instantiation of Y_m is denoted by y_m , where $p(Y_m = y_m)$, or simply $p(y_m)$, denotes the probability that the variable Y_m takes the value y_m . $\mathbf{Y} = (Y_1, \dots, Y_M)$ represents an M -dimensional vector of random variables, and $\mathbf{y} = (y_1, \dots, y_M)$ is a realization. The conditional probability of y_m given y_j is written as $p(Y_m = y_m | Y_j = y_j)$ (or simply $p(y_m | y_j)$). The joint probability distribution of \mathbf{Y} is denoted by $p(\mathbf{Y} = \mathbf{y})$, or simply $p(\mathbf{y})$. In this section, Pop denotes a data set, or the set of N instantiations of the vector of the random variables $\mathbf{Y} = (Y_1, \dots, Y_M)$.

The general framework of an EDA is illustrated in Figure 3.

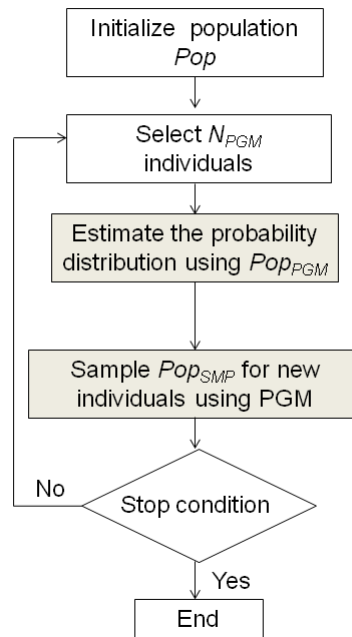


Figure 3: The general framework of an EDA.

Usually, an EDA starts by generating a random population Pop of N solutions in *Initialize population Pop* block. Each solution is an individual $\mathbf{y} = (y_1, \dots, y_M)$ with M elements.

Solutions are then evaluated using one or more objective functions and, in *Select N_{PGM} individuals* block, a subset of them is selected according to a pre-defined criterion. Pop_{PGM} represents the population of the N_{PGM} individuals selected from Pop .

The selected solutions are used to learn a PGM in the block *Estimate the probability distribution using PGM*, according $p(\mathbf{y} | Pop_{PGM})$, that is, the conditional probability of a particular individual \mathbf{y} among the selected ones. This conditional probability must be estimated

at every generation. The most important step is to find out the interdependencies between variables that represent one point in the search space. The basic idea consists in inducing probabilistic models from the best individuals of the population.

Once the probabilistic model has been estimated, the model is sampled to generate a population Pop_{smp} of new individuals (new solutions) in block *Sample Pop_{smp} for new individuals using PGM*. The cycle of evaluation, selection, modeling, and sampling is repeated until a stop condition is fulfilled.

The steps that differentiate EDAs from other EAs are the construction of a probabilistic model and the process of sampling new candidate solutions based on the model. These blocks are highlighted in different color in Figure 3.

The next section introduces EDAs classification, based on the type of variables and the interdependencies that the PGM can account for (LARRAÑAGA; LOZANO, 2002).

3.2 CLASSIFICATION

Depending on the problem solution representation, EDAs can be categorized as discrete, permutation and real-valued based variables. Candidate solutions in EDAs have usually fixed length. However, variables can either be discrete or receive a real value that covers an infinite domain. Candidate solutions can also be represented by a permutation over a given set of elements, e.g. solutions for travelling salesman or quadratic assignment problem. This research concerns only EDAs for discrete variables. EDAs can be divided into three groups: Univariate, Bivariate and Multivariate according to the level of interactions among variables.

Univariate EDAs assume no interaction among variables. The joint probability mass function (pmf) of a solution \mathbf{y} , which will be used afterward in the sampling process, is simply the product of univariate marginal probabilities of all M variables in that solution, that is $p(\mathbf{y}) = \prod_{m=1}^M p(y_m)$. Algorithms in this category have simple model building and sampling procedures and can solve problems where variables are independent. However, for problems with strong variable interactions, they tend to produce poor results. Different variants in this category include: Population-based Incremental Learning (PBIL) (BALUJA, 1994), Univariate Marginal Distribution Algorithm (UMDA) (MÜHLENBEIN; PAAB, 1996) and Compact Genetic Algorithm (cGA) (HARIK et al., 1999).

For bivariate EDAs, pairwise interactions among variables in the solutions are considered. Therefore, the probabilistic models in this category contain factors involving conditional probability of two interacting variables. New solutions are also sampled in a certain

ordering of variables to address the conditional probabilities. These algorithms outperform univariate EDAs in problems with pair-wise variable interactions; however, they tend to fail when multiple interactions among variables exist in the problem (PHAM, 2011). Mutual Information Maximization for Input Clustering (MIMIC) (BONET et al., 1997), Combining Optimizer with Mutual Information Tree (COMIT) (BALUJA; DAVIES, 1997) and Bivariate Marginal Distribution Algorithm (BMDA) (PELIKAN; MUEHLENBEIN, 1999), all of them use bivariate models to estimate probability distribution.

Multivariate EDAs use probabilistic models able to capturing multivariate interactions between variables. Algorithms using multivariate models of probability distribution include: Extended Compact Genetic Algorithm (ECGA) (HARIK, 1999), Estimation of Bayesian Network Algorithm (EBNA) (ETXEBERRIA; LARRAÑAGA, 1999), Factorized Distribution Algorithm (FDA) (MÜHLENBEIN; MAHNIG, 1999), Bayesian Optimization Algorithm (BOA) (PELIKAN et al., 1999), Hierarchical Bayesian Optimization Algorithm (hBOA) (PELIKAN et al., 2003), Markovianity-based Optimization Algorithm (MOA) (SHAKYA; SANTANA, 2008), Affinity Propagation EDA (AffEDA) (SANTANA et al., 2010).

One of the most general probabilistic models for discrete variables used in EDAs and MOEDAs is the Bayesian Network (PEARL, 2000; KOLLER; FRIEDMAN, 2009), and we briefly describe it in the next section.

3.3 BAYESIAN NETWORK

Bayesian networks (BN) are directed acyclic graphs (DAG) 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 (COOPER; HERSKOVITS, 1992; KORB; NICHOLSON, 2010).

3.3.1 BASIC CONCEPTS

As in Section 3.1, let $\mathbf{Y} = (Y_1, \dots, Y_M)$ be a vector of 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 (LARRAÑAGA et al., 2012): a structure and a set of local parameters. The set of local parameters Θ 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 (BENGOETXEA, 2002). 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, a Bayesian network encodes a factorization for the mass probability as follows:

$$p(\mathbf{y}) = p(y_1, y_2, \dots, y_M) = \prod_{m=1}^M p(y_m | \mathbf{pa}_m^B) \quad (12)$$

Equation 12 states that the joint pmf of the variables can be computed as the product of each variable's conditional probability given the values of its parents.

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

$$p(y_m^k | \mathbf{pa}_m^{j,B}) = \theta_{y_m^k | \mathbf{pa}_m^{j,B}} = \theta_{m,jk} \quad (13)$$

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 and t_m is the total number of different possible instantiations of the parent variables of Y_m given by $t_m = \prod_{Y_v \in \mathbf{Pa}_m^B} s_v$, where s_v is the total of possible values (states) that Y_v can assume. The parameter $\theta_{m,jk}$ represents the conditional probability that variable Y_m takes its k -th value (y_m^k), knowing that its parent variables have taken their j -th combination of values ($\mathbf{pa}_m^{j,B}$). This way, the parameter set is given by $\Theta = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m, \dots, \boldsymbol{\theta}_M\}$, where $\boldsymbol{\theta}_m = (\theta_{m11}, \dots, \theta_{m,jk}, \dots, \theta_{m,t_m,s_m})$.

BN's are often used for modeling multinomial data with discrete variables (PEARL, 1988) generating new solutions using the particular conditional probability described in Equation 13 (probabilistic logic sampling (HENRION, 1986)).

The next sections present a way to estimate Θ parameters and the B structure.

3.3.2 PARAMETER ESTIMATION

Generally, the parameters in the whole set Θ are unknown, and their estimation process is based on $p(\Theta | Pop, B)$, where Pop is the current data with N observations (instantiations) of \mathbf{Y} . Assuming a fixed structure B , let us consider the following assumption based on Bayes Theorem:

$$p(\Theta|Pop, B) = \frac{L(\Theta; Pop, B) * p(\Theta|B)}{p(Pop, B)} \quad (14)$$

In order to estimate the *posteriori* $p(\Theta|Pop, B)$ lets assume that the parameters $\theta_{mj} = \{\theta_{mj1}, \theta_{mj2}, \dots, \theta_{mjs_m}\}$ are independent (i.e. *priori* and *posteriori* of Θ can be factorized through θ_{mj}) and N_{mjk} is the number of observations in *Pop* for which Y_m assumes the k -th value given the j -th combination of values from its parents. According to this, $\mathbf{N}_{mj} = \{N_{mj1}, \dots, N_{mjs_m}\}$ fits a multinomial distribution, with a pmf defined as follows.

$$p(\mathbf{N}_{mj}|\theta_{mj}, B) = \frac{N_{mj}!}{N_{mj1}!N_{mj2}!\dots N_{mjs_m}!} (\theta_{mj1}^{N_{mj1}})(\theta_{mj2}^{N_{mj2}})\dots(\theta_{mjs_m}^{N_{mjs_m}}) \quad (15)$$

where $\sum_{k=1}^{s_m} N_{mjk} = N_{mj}$ and $\sum_k \theta_{mjk} = 1$.

Considering the likelihood $L(\theta_{mj}; Pop, B)$ given by Equation 15, there are two approaches to estimate each θ_{mjk} parameter: Maximum Likelihood Estimate (MLE) and Bayesian Estimate.

With MLE, we expect to find a vector in Θ that maximizes the likelihood. We can denote this vector as $\hat{\theta}$. In MLE, each $\hat{\theta} \in \Theta$ is a point estimation, not a random variable. Therefore, MLE does not consider any *priori* information, and the estimation is calculated according to Equation 16, based on a frequentist analysis (by setting the derivative of Equation 15 to zero):

$$\hat{\theta}_{mjk} = p(y_m^k | \mathbf{pa}_m^{j,B}, \theta_m) = \frac{p(Y_m = y_m^k, \mathbf{pa}_m^j)}{p(\mathbf{pa}_m^j)} = \frac{f(y_m^k, \mathbf{pa}_m^j)}{f(\mathbf{pa}_m^j)} = N_{mjk}/N_{mj} \quad (16)$$

where $f(\cdot)$ denotes the relative frequency and $\hat{\theta}_{mjk}$ is the MLE estimated parameter for θ_{mjk} .

Regarding to Bayesian estimation, it calculates the *posteriori* distribution $p(\Theta|Pop, B)$ considering a *priori* information $p(\Theta|B)$. In practice, it is useful to require that the prior for each factor is a conjugate prior. For example, Dirichlet priors are conjugate priors for multinomial factors.

Considering we can assume that a *priori* θ fits a Dirichlet distribution with hyperparameters $\alpha_{mj} = (\alpha_{mj1}, \dots, \alpha_{mjs_m})$ where $\alpha_{mjk} \geq 1$, $\alpha_{mj} = \sum_{k=1}^{s_m} \alpha_{mjk}$ and its expected value is given by Equation 17:

$$E(\theta_{mjk}|B) = \alpha_{mjk}/\alpha_{mj}; \quad (17)$$

Dirichlet prior can be written as the following density probability function:

$$p(\boldsymbol{\theta}_{mj}|B, \boldsymbol{\alpha}_{mj}) = \frac{\Gamma(\boldsymbol{\alpha}_{mj})}{\Gamma(\alpha_{mj1}) \dots \Gamma(\alpha_{mjs_m})} \theta_{mj1}^{\alpha_{mj1}-1} \theta_{mj2}^{\alpha_{mj2}-1} \dots \theta_{mjs_m}^{\alpha_{mjs_m}-1} \quad (18)$$

where $\Gamma(\cdot)$ is the Gamma function that satisfies $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1) = 1$; according definition $\Gamma(x) = (x-1)!$, resulting in $\Gamma(1) = (1-1)! = 1$ and $\Gamma(x+1) = x! = x(x-1)(x-2)\dots 1 = x\Gamma(x)$ (DEGROOT, 2005).

The term $\frac{\Gamma(\boldsymbol{\alpha}_{mj})}{\Gamma(\alpha_{mj1}) \dots \Gamma(\alpha_{mjs_m})}$ can be considered a normalizer factor and Equation 18 can be rewritten as follows:

$$p(\boldsymbol{\theta}_{mj}|B, \boldsymbol{\alpha}_{mj}) \propto \theta_{mj1}^{\alpha_{mj1}-1} \theta_{mj2}^{\alpha_{mj2}-1} \dots \theta_{mjs_m}^{\alpha_{mjs_m}-1} \quad (19)$$

Considering the *priori* as $p(\boldsymbol{\theta}_{mj}|B, \boldsymbol{\alpha}_{mj})$ and the likelihood as $p(\mathbf{N}_{mj}|B, \boldsymbol{\theta}_{mj})$, we obtain the *posteriori* as $p(\boldsymbol{\theta}_{mj}|\boldsymbol{\alpha}_{mj}, \mathbf{N}_{mj})$, given by Equation 20, which fits the Dirichlet distribution with parameters $\boldsymbol{\alpha}_{mj} = (\alpha_{mj1} + N_{mj1}, \dots, \alpha_{mjs_m} + N_{mjs_m})$.

$$p(\boldsymbol{\theta}_{mj}|B, \boldsymbol{\alpha}_{mj}, \mathbf{N}_{mj}) \propto \theta_{mj1}^{\alpha_{mj1}+N_{mj1}-1} \theta_{mj2}^{\alpha_{mj2}+N_{mj2}-1} \dots \theta_{mjs_m}^{\alpha_{mjs_m}+N_{mjs_m}-1} \quad (20)$$

Assuming the expected value for the *posteriori* as follows:

$$E(\theta_{mjk}|\mathbf{N}_{mj}, B) = (\alpha_{mjk} + N_{mjk}) / (\alpha_{mj} + N_{mj}). \quad (21)$$

where $N_{mj} = \sum_{k=1}^{s_m} N_{mjk}$, $\alpha_{mj} = \sum_{k=1}^{s_m} \alpha_{mjk}$, and, considering the α_{mjk} values as 1, we have:

$$E(\theta_{mjk}|\mathbf{N}_{mj}, B) = (1 + N_{mjk}) / (s_m + N_{mj}). \quad (22)$$

The expected value $E(\theta_{mjk}|\mathbf{N}_{mj}, B)$ of θ_{mjk} is an estimate of θ_{mjk} , shown in Equation 23.

$$\hat{\theta}_{mjk} = (1 + N_{mjk}) / (s_m + N_{mj}) \quad (23)$$

In this work we estimate the parameters by Bayesian Estimate, using the Dirichlet prior.

3.3.3 STRUCTURE

There are three main approaches to learn BN structures: score-based learning, constraint-based learning, and hybrid methods (YUAN; MALONE, 2013).

Score-based learning methods evaluate the quality of BN structures using a scoring function, like Bayesian Dirichlet (BD)-metric (HECKERMAN et al., 1995), and selects the best one. Constraint-based learning methods typically use statistical tests to identify conditional independence relations from the data and build a BN structure that best fits those relations (PEARL, 1988). Hybrid methods aim to integrate the two approaches, like those adopted by Tsamardinos et al. (2006a) and Perrier et al. (2008). These 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.

Most of the developed structure learning algorithms fall into the score-based approaches. The BD metric, defined by Equation 24, combines a priori knowledge about the problem and the statistical data from a given data set.

$$p(B|Pop) = p(B) \prod_{m=1}^M \prod_{j=1}^{t_m} \frac{\Gamma(\alpha_{mj})}{\Gamma(\alpha_{mj} + N_{mj})} \prod_{k=1}^{s_m} \frac{\Gamma(\alpha_{mjk} + N_{mjk})}{\Gamma(\alpha_{mjk})} \quad (24)$$

where $p(B)$ is the prior factor of quality information of the network B . If there is no prior information for B , $p(B)$ is considered a uniform probability distribution (LUNA, 2004) and generally its value is set to 1 (CROCOMO; DELBEM, 2011). The product on $j \in \{1, \dots, t_m\}$ runs over all combinations of the parents of Y_m and the product on $k \in \{1, \dots, s_m\}$ runs over all possible values of Y_m .

Through α_{mjk} and $p(B)$, a priori information about the problem is incorporated into the metric. The parameter α_{mjk} stands for prior information about the number of instances that have Y_m set to its k -th value and the set of parents of Y_m is instantiated to its j -th combination. The prior network can be set to an empty network, when there is no such information.

In the so-called K2 metric (COOPER; HERSKOVITS, 1992) for instance, the parameters α_{mjk} can be set to 1 as there is no a priori information about the problem, and Equation 24 becomes Equation 25:

$$p(B|Pop) = \prod_{m=1}^M \prod_{j=1}^{t_m} \frac{(s_m - 1)!}{(N_{mj} + s_m - 1)!} \prod_{k=1}^{s_m} (N_{mjk})! \quad (25)$$

Since the factorials in Equation 25 can grow to huge numbers, a computer overflow

might occur. Thus the logarithm of the scoring metric $\log(p(B|Pop))$ is usually used, as shown in Equation 26, where

$$\log(p(B|Pop)) = \sum_{m=1}^M \sum_{j=1}^{t_m} \left(\log \left(\frac{(s_m - 1)!}{(N_{mj} + s_m - 1)!} \right) + \sum_{k=1}^{s_m} \log((N_{mjk}!)) \right) \quad (26)$$

Various algorithms can be used to look for the network structure that maximizes the value of a scoring metric, like a simple greedy search algorithm, local hill-climbing, simulated annealing, tabu search and evolutionary computation (LARRAÑAGA et al., 2012).

3.3.4 NAIVE BN

One of the most important application of BN is as a classifier, where an instance described by a number of features must to be classified in one of several distinct classes. However, many application domains involve instances that have to be assigned to a most likely combination of classes (GAAG; WAAL, 2006). A multidimensional BN (MBN) classifier is a Bayesian network with a restricted topology designed to address these applications, including one or more class variables and one or more feature variables.

An MBN classifier has a set of random variables \mathbf{Y} partitioned into a set $\mathbf{Y}_F = \{F_1, \dots, F_m\}$, $m \geq 1$, of feature variables and a set $\mathbf{Y}_C = \{C_1, \dots, C_n\}$, $n \geq 1$, with the class variables. The subgraph B_C of \mathbf{Y}_C is called the classifiers class subgraph; the subgraph B_F of \mathbf{Y}_F is called its feature subgraph. The subgraph B_{CF} , which includes the arcs from the class variables to the feature variables, is called the feature selection subgraph, since it represents the selection of features that are deemed relevant for classification in view of the variables $\{C_1, \dots, C_n\}$ (WAAL; GAAG, 2007). Figure 4 illustrates an example of an MBN with $m = 4$ and $n = 3$.

According to Karshenas et al. (2014), an MBN can address the class variables of given features variables, the most probable feature values for a given combination of class variables, and the most probable values for a subset of features or classes given the value of the others. Our interest in this work is to apply the MBN concepts to a MOP context, aiming to answer questions like: what are the estimated objectives values of a given solution, and/or what is the most probable solution resulting in a specific value-setting for the objectives.

MBN classifiers use different search algorithms to find the optimal way to represent data (GAAG; WAAL, 2006), and most of them are score-based (MORAN et al., 2009).

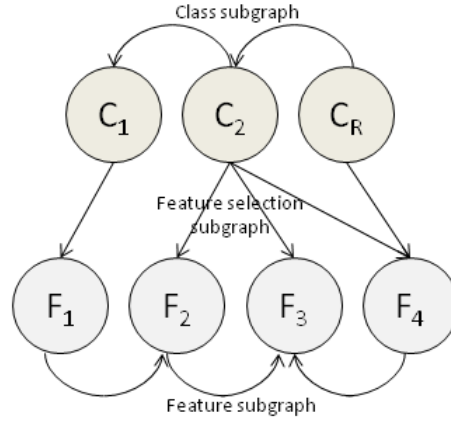


Figure 4: An MBN classifier with 3 class variables and 4 feature variables.

The presence of several classes adds complexity and increases the number of network parameters (BOUHAMED et al., 2012). Therefore, some authors have proposed the use of methods to overcome these shortcomings, such as: K2 algorithms (LERNER; MALKA, 2011; PORWAL et al., 2006; MALKA; LERNER, 2004; HRUSCHKA; EBECKEN, 2007; CARTA et al., 2011; COOPER; HERSKOVITS, 1992; MORAN et al., 2009), the Greedy Search (SPIRITES et al., 2000; MORAN et al., 2009), the Greedy Hill Climber (BN-HC) and the Repeated Hill Climber (BN-RHC) (WITTEN et al., 2016; MORAN et al., 2009).

The K2 algorithm, introduced by Cooper and Herskovits (1992) and presented in Algorithm 1, is a score-based greedy local search technique that applies the K2 metric (Equation 25). It starts by assuming that a node, in an ordered list, does not have any parent (Step 3), 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 scoring metric the most (Step 8). When no edge increases the metric anymore, attention turns to the next node (Step 14). According to Witten et al. (2016), in order to avoid overfitting, the number of parents for each node can be restricted to a predefined upper bound. Due to the fact that only edges from previously processed nodes (pre-defined ordered list) are considered, this procedure guarantees that acyclic graphs are always generated.

Note that Equation 27 is a factor of Equation 25 for a specific m value, as follows:

$$f(m, \mathbf{Pa}_m) = \prod_{j=1}^{t_m} \frac{(s_m - 1)!}{(N_{mj} + s_m - 1)!} \prod_{k=1}^{s_m} (N_{mjk})! \quad (27)$$

An example of the K2 algorithm to learn the topology of a BN is presented in the Appendix A.

Algorithm 1: K2 Algorithm

INPUT: a set of M nodes;
 a pre-defined nodes ordered list **order** with M elements;
 an upper bound u on the number of parents a node may have;
 the current data set Pop with N observations (instantiations);

OUTPUT: the set of the parents \mathbf{Pa}_m of each node m ;

- 1: **for** $e = 1$ to M **do**
- 2: $m = \text{order}_e$;
- 3: $\mathbf{Pa}_m = \emptyset$;
- 4: $\text{Score}_{old} = f(m, \mathbf{Pa}_m)$; {This function is computed using Equation 27}
- 5: $proceed = TRUE$;
- 6: **while** $proceed$ and $|\mathbf{Pa}_m| < u$ **do**
- 7: $Pred(Y_m)$ = the set of the predecessors of Y_m according **order**;
- 8: Let i be the node in $Pred(Y_m) - \mathbf{Pa}_m$ that maximizes $f(m, \mathbf{Pa}_m \cup \{i\})$;
- 9: $\text{Score}_{new} = f(m, \mathbf{Pa}_m \cup \{i\})$;
- 10: **if** $\text{Score}_{new} > \text{Score}_{old}$ **then**
- 11: $\text{Score}_{old} = \text{Score}_{new}$;
- 12: $\mathbf{Pa}_m = \mathbf{Pa}_m \cup \{i\}$;
- 13: **else**
- 14: $proceed = FALSE$;
- 15: **end if**
- 16: **end while**
- 17: **end for**

As presented in Algorithm 2, the BN-HC procedure uses a hill climbing algorithm adding, removing and reversing arcs on the space of the directed graphs until a local maximum (network scoring metric) is reached. It starts with a random network structure (Step 1) and generates a number of new networks applying all possible single-edge addition, removal and reversal operations that will map the current network into a new valid structure (Step 5), and then applies the operation that will result in the highest increase in the network scoring metric (Step 6). This new structure with the highest score is denoted as the new current network (Step 9) and new structures are generated again until reaching a local optimum of scoring metric (MORAN et al., 2009) (Step 8). Any scoring metric can be used.

In the BN-RHC, a BN-HC variant approach, the algorithm searches structures by repeatedly generating a random network and applying the BN-HC to it. The structure search is restarted from a new random structure after reaching a local optimum (MORAN et al., 2009). This is repeated according to a given number of times, usually the number of node score evaluations (KARSHENAS et al., 2014). In both approaches the algorithm returns the highest network scoring metric.

Different types of MBN classifier can be distinguished based on their graphical

Algorithm 2: BN-HC Algorithm

INPUT: Pop , the current data set with N observations (instantiations)

OUTPUT: B =best structure

```

  {Random generate a network structure and calculate its score}
  1:  $B$ =RandomStructure( $Pop$ );
  2:  $Score(B) = p(B|Pop)$ ; {The score is computed using any score metric, i.e., Equation 25}
  3:  $proceed = TRUE$ ;
  4: while  $proceed$  do
    {Apply all possible single-edge addition, removal or reversal operations on  $B$  for
    generating new structures}
  5:  $S$ =NewStructures( $B$ );
  6:  $Score_{max} = ScoreMax(S)$ ;
    {Calculate the score for each structure in  $S$  and find the maximum value}
  7:  $B'$  = the structure  $\in S$  with the maximum score;
  8: if  $Score_{B'} > Score_B$  then
  9:    $B = B'$ ;
  10: else
  11:    $proceed = FALSE$ 
  12: end if
  13: end while

```

structures. An example is the naive multidimensional classifier in which both the class subgraph and the feature subgraph are empty (GAAG; WAAL, 2006) (i.e. there are edges only in feature selection subgraph, as presented in Figure 5). This structure is the base of our proposal, 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.

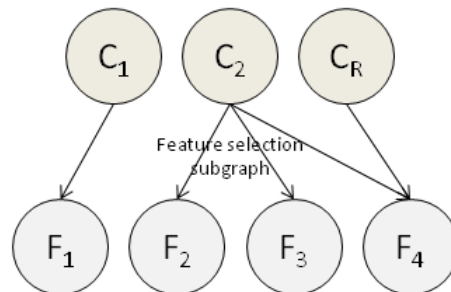


Figure 5: A naive MBN classifier with 3 class variables and 4 feature variables.

In this work we investigate both K2 and BN-HC structure learning algorithm. The BN-RHC method is not considered here due to its computational costs. We adopted K2 metric

as score-based technique, although any other method could be used as well.

3.4 SUMMARY

EDAs are promising algorithms often applied to solve MOPs. In this chapter, we have presented a general framework and a classification, that categorizes EDAs based on the level of interactions among variables. We have related some representative works emphasizing the BN model as a prominent PGM. BNs are often used to represent discrete variables using a multinomial distribution. This chapter has also justified the use of the BN in the context of this work. This way the nodes encode a probability function associated with a conditional relation between parents nodes. The model learning is divided into parameter and structure estimation. With the parameters being estimated through the Bayesian approach, the structure learning algorithms search for the structure that maximizes the value of given scoring metric. We have adopted a naive multidimensional network to further investigate both K2 and BN-HC structure learning algorithm. The next chapter presents the problems addressed in this work, and briefly describes related solving approaches using MOEAs and EDAs.

4 THE MULTI-OBJECTIVE PROBLEMS ADDRESSED IN THIS WORK

This chapter presents the problems addressed in this work. First we investigate the MOKP, according to the multi-objective formulation proposed by Ishibuchi et al. (2015), Tanigaki et al. (2014) and Zitzler and Thiele (1999). After that we explore the MNK-Landscape problem, proposed by Aguirre and Tanaka (2004). We also present some previous work addressing these MOPs in the context of MOEAs and EDAs. We emphasize that although the experiments are conducted using specific MOPs, the proposal is general, and can be extended to a wide range of multi and many-objective combinatorial problems.

4.1 THE MULTI-OBJECTIVE KNAPSACK PROBLEM

The 0/1 knapsack problem is a widely studied problem due to its practical importance. In the last years the generalization of this problem has been well studied and many algorithms for solving some variants have been proposed. Evolutionary approaches for solving MOKP are of great interest (TAN; JIAO, 2013) with many works presented in the last years (JASZKIEWICZ, 2001), (JASZKIEWICZ, 2002b), (VIANNA; VIANNA, 2013), (LUST; TEGHEM, 2012), (GANDIBLEUX; FRÁVILLE, 2000), (ZITZLER; THIELE, 1999), (TANIGAKI et al., 2014) and (ISHIBUCHI et al., 2015).

Next sections present the MOKP general formulation and some related works addressing this problem.

4.1.1 GENERAL FORMULATION

Given a total of R objective functions (knapsacks) and Q items, a_{rq} is the profit of item $q = 1, \dots, Q$, in the knapsack $r = 1, \dots, R$, b_{rq} denotes the weight of item q in the knapsack r , and c_r is the constraint capacity of knapsack r . MOKP can be formulated as follows:

$$\begin{aligned} \max_{\mathbf{x}} \mathbf{z}(\mathbf{x}) &= (z_1(\mathbf{x}), \dots, z_R(\mathbf{x})) \\ \text{subject to } \sum_{q=1}^Q b_{rq}x_q &\leq c_r, \quad r = 1, \dots, R \end{aligned} \quad (28)$$

$$\text{with } z_r(\mathbf{x}) = \sum_{q=1}^Q a_{rq}x_q, \quad \mathbf{x} \in \{0, 1\}^Q$$

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

The mono-objective version of this problem has been extensively studied in the literature (BAZGAN et al., 2009). One variant is Bounded Knapsack Problem (BKP), a generalization of 0/1 knapsack problem in which multiple instances of each item are taken in a single knapsack. BKP is concerned with a knapsack that has positive integer capacity c . There are Q distinct items that may potentially be placed in the knapsack. Item q has a positive integer weight b_q and positive integer profit a_q . In addition, there are n_q available copies of item q , where quantity n_q is a positive integer. Assuming that y_q represents the number of copies of item q that are to be placed into the knapsack, the objective is to choose a subset of items that maximizes the corresponding profit sum without exceeding the capacity of the knapsack. The BKP problem is formulated as follows:

$$\begin{aligned} \max_{\mathbf{x}} \mathbf{z}(\mathbf{x}) &= \sum_{q=1}^Q y_q a_q \\ \text{subject to } \sum_{q=1}^Q b_q y_q &\leq c, \end{aligned} \quad (29)$$

$$\text{with } 0 \leq y_q \leq n_q, \quad q \in \{1, \dots, Q\}$$

In the last decades, many algorithms for solving the MOKP variant have been proposed using different techniques (TAN; JIAO, 2013). Moreover, in the multi-objective case, many real-world applications are reported dealing with capital budgeting (ROSENBLATT; SINUANY-STERN, 1989), selection of transportation investment alternatives (TENG; TZENG, 1996), relocation issues arising in conservation biology (KOSTREVA et al., 1999), planning remediation of contaminated lightstation sites (JENKINS, 2002) and action plan in the social and medico-social sector (CHABANE et al., 2015).

So, the problem has been chosen due to its combinatorial and multi-objective features and because it serves to model several other real-world problems.

4.1.2 MOEA TO SOLVE MOKP

One of the first works is presented by Zitzler and Thiele (1999) and performs a comparative study of five different MOEAs on the MOKP. They compare four well-known MOEAs, i.e., vector-evaluated genetic algorithm (FOURMAN, 1985), Haleja's and Lin's Genetic Algorithm (HAJELA; LIN, 1992), NPGA (HORN et al., 1994), and NSGA (SRINIVAS; DEB, 1994). The authors propose a method called SPEA, which outperforms the other approaches on the experimental results.

Knowles and Corne (2000) have continued the previous experiment using the same set of test instances. They compare SPEA with Memetic-Pareto Archive Evolution Strategy (M-PAES), a multiple-objective memetic algorithm that hybridizes local search with recombination operators. Their results indicate that M-PAES performs better than SPEA.

The goal of the methods proposed by Arroyo (2002) is to generate, in a reasonable time, a set of approximated Pareto-optimal solutions, allowing the decision maker to choose a solution of interest. First, the author develops a constructive heuristic to generate a set of dominant solutions. In order to find sets of dominant solutions that are closer to the Pareto-optimal sets, he develops a local search heuristic and two metaheuristics. The first one is based on Genetic algorithm and uses the concept of Pareto dominance combining elitism, population diversity and local search strategies. Then, he proposes a Tabu Search-based method which explores a set of solutions in parallel to find a variety of solutions distributed along the Pareto front. The performance of the proposed methods is tested on a number of instances for the multi-objective flowshop scheduling problem and the MOKP.

Jaszkiewicz (2002b) compares four algorithms: SPEA, M-PAES, Ishibuchi's and Murata's Multiple-objective Genetic Local Search (IMMOGLS) (ISHIBUCHI; MURATA, 1998), and the MOGLS algorithm proposed by Jaszkiewicz (2002a). The same set of MOKP instances, recombination, mutation, repair operators and similar performance measures as those adopted by Zitzler and Thiele (1999) are used. The only difference is that it sorts the elements removed by a greedy repair procedure on the basis of a weighted scalarizing function value.

Ahn et al. (2010) present a Hybrid Multi-objective Evolutionary Algorithm (HMEA) that deals with MOPs quite efficiently. The aim is to discover new non-dominated solutions in the neighborhood of the most promising individuals in order to effectively push individuals toward the global Pareto front. It was achieved by bringing the strength of an Adaptive Local Search (ALS) to bear upon the evolutionary multi-objective optimization. The ALS is devised by combining a weighted fitness strategy and a knowledge-based local search which does not

incur any significant computational cost. To show the utility of HMEA, the ALS for MOKP is developed by exploiting the problem's knowledge. Experimental results have shown the effectiveness (including convergence and diversity criteria) of the proposed approach.

Vianna and Vianna (2013) developed algorithms based on Greedy Randomized Adaptive Search Procedure (GRASP) and Iterated Local Search (ILS) metaheuristics for the MOKP. Computational experiments on benchmark instances show that the proposed algorithms are very robust outperforming other heuristics in terms of solution quality and run times.

A new version of MOEA/D with uniform design for solving MOKP is proposed by Tan and Jiao (2013). The algorithm adopts the uniform design method to generate the aggregation coefficient vectors so that the decomposed scalar optimization subproblems are uniformly scattered, and therefore the algorithm can uniformly explore the region of interest from the initial iteration. Results show that the proposed algorithm significantly outperforms NSGA-II, SPEA2 and PESA for 2-objective, 3-objective and 4-objective knapsack problems.

Ke et al. (2014) combine ideas from EA, decomposition approaches and Pareto local search, and suggest a simple yet efficient Memetic Algorithm for Combinatorial Multi-objective optimization problems (MoMad). It decomposes a combinatorial MOP into a number of single objective optimization problems using an aggregation method. At each generation, MoMad maintains three populations: population PL for recording the current solution to each subproblem which will undergo perturbation and single objective local search operators; population PP containing all the nondominated solutions in PL for Pareto local search; and population PE for maintaining all the nondominated solutions found so far during the search. A Pareto local search method is first applied to search a neighborhood of each solution in PP to update PL and PE. Then a single objective local search is applied to each perturbed solution in PL for improving PL and PE, and re-initializing PP. Extensive experiments are conducted by Ke et al. (2014) to study MoMad and compare it with some other state-of-the-art algorithms on the multi-objective traveling salesman problem and MOKP. Experimental results show that the proposed algorithm outperforms or performs similarly the best heuristics on these two problems so far.

Ishibuchi et al. (2015) examine the behavior of some classes of MOEAs on MOKPs represented by Pareto dominance-based, scalarizing function-based, and hypervolume-based algorithms. NSGA-II, MOEA/D, SMS-EMOA, and HypE are examined using knapsack problems with 2 to 10 objectives. Experimental results on randomly generated instances of MOKP are consistent with well-known performance deterioration of Pareto dominance-based algorithms. That is, NSGA-II is outperformed by the other algorithms. However, they also

show that NSGA-II outperforms the other algorithms when objectives are highly correlated. MOEA/D shows totally different search behavior depending on the choice of a scalarizing function and its parameter value. Some MOEA/D variants work very well only in two-objective problems while others work well in many-objective problems with 4 to 10 objectives. They also provide other interesting observations such as the performance improvement by similar parent recombination and the necessity of diversity improvement for MOKPs.

Recently, Chabane et al. (2017) present a solution method using the iterated local search based on R2 quality indicator (IBMOLS) to address the issue of elaborating efficient action plans in social and medico-social sector in France, which includes more than 34,000 different structures (rest houses, accommodation and rehabilitation centers, work-based support centers, etc). In this work, the authors investigate a real case of multi-objective action plan optimization problem which aims to improve the whole quality of these structures. The action plan optimization problem is a practical case of the MOKP. They assess the approach on problem instances with 2 to 8 objectives and up to 500 candidate actions and demonstrate its usefulness as a key component of a decision support system for social and medico-social structures, presenting competitive results in comparison with NSGA-II.

Table 1 summarizes the main characteristics of MOEAs approaches to solve MOKP.

Table 1: MOEAs to MOKP main characteristics

MOEA	Objectives	Approach	Local Search	References
SPEA	2,3 and 4	Pareto	No	(ZITZLER; THIELE, 1999)
MPAES	2,3 and 4	Pareto	Yes	(KNOWLES; CORNE, 2000)
MOGLS	2,3 and 4	Pareto	Yes	(JASZKIEWICZ, 2001), (JASZKIEWICZ, 2002b)
MOGLS	2	Pareto	Yes	(ARROYO, 2002)
HMEA	2 and 3	Pareto	Yes	(AHN et al., 2010)
MGRASP	2,3 and 4	Pareto	Yes	(VIANNA; VIANNA, 2013)
MILS	2,3 and 4	Pareto	Yes	(VIANNA; VIANNA, 2013)
MOEA/D	2,3 and 4	Scalarizing Functions	Yes	(TAN; JIAO, 2013)
MOMAD	2,3 and 4	Scalarizing Functions	Yes	(KE et al., 2014)
NSGA-II	2-10	Pareto	No	(ISHIBUCHI et al., 2015)
MOEA/D	2-10	Scalarizing Functions	No	(ISHIBUCHI et al., 2015)
SMS-EMOA	2-10	Hypervolume	No	(ISHIBUCHI et al., 2015)
HypE	2-10	Hypervolume	No	(ISHIBUCHI et al., 2015)
IBMOLS	2-8	R2 based LS	Yes	(CHABANE et al., 2017)

4.1.3 EDA APPROACHES FOR MOKP

Schwarz and Ocenasek (2001b) propose a Pareto-based BOA algorithm to MOKP that replaces the fitness assignment and replacement step of standard BOA by the diversity-preserving niching method based on the promising Pareto technique. They implemented the multi-objective Pareto BOA algorithm as a modification of the original single-objective BOA algorithm using the same strength criterion applied to SPEA (ZITZLER et al., 2001) for the Pareto oriented fitness. This criterion assigns a fitness to each solution

based on the relation between the number of the individuals that this solution dominates in entire population.

The paper presented by Laumanns and Ocenasek (2002) investigates the usefulness of PGM in multi-objective optimization, where the aim is to approximate the set of Pareto-optimal solutions. The PGM is built according to solutions selected by a new operator based on $(\mu + \lambda, \varepsilon)$ ¹, where dominated individuals are also allowed to survive, depending on the number of individuals they are dominated by. The approach integrates the model building and sampling techniques of BOA using binary decision tree as PGM instead of a Bayesian Network. The model is composed of a set of trees, one tree is allocated for each variable. The dependence assertions are expressed by the existence of splitting nodes and the local probabilities are stated in leaf nodes. The behavior of the resulting Bayesian Multi-objective Optimization Algorithm (BMOA) is empirically investigated on the MOKP.

Li et al. (2004) present a Hybrid Estimation of Distribution Algorithm (MOHEDA) based on UMDA and scalarizing function for solving the MOKP. Further, local search using a weighted sum method is applied to the initial and to the sampled population. A random repair method is also used to handle constraints. Results show that MOHEDA outperforms several other state-of-the-art algorithms.

Gupta et al. (2014) develop a new approach to solve BKP. This paper proposes an EDA based on probability vector model using greedy operator approach to solve the mono-objective BKP, defined in Equation 29. Greedy operator ensures the feasibility of the solutions (repair method).

Recently, Zhou et al. (2015) suggest an approach combining an EDA with cheap and expensive local search methods for making use of both global statistical information and individual location information. In this approach, part of a new solution is sampled from a modified UMDA, and the rest is generated by refining a parent solution through a cheap local search method that does not need actual function evaluations. The basic idea of the cheap local search is to replace some components of sampled solutions by those of solutions derived from a surrogate model with a specific probability. To generate high-quality solutions, best solutions are used to construct surrogate models. When the population has converged, an expensive local search method is applied to improve a promising solution found so far. Experimental results have shown that for simple test instances, this algorithm can produce better or similar solutions but with faster convergence speed than the compared methods and for some complex

¹ μ denotes the number of parents that survive to the next generation, λ is the number of offspring solutions and ε is the selection/archiving operator.

test instances it can find better solutions. This work addresses the knapsack problem for a single objective, like (GUPTA et al., 2014), considering it as a global optimization problem.

4.2 THE MULTI-OBJECTIVE NK-LANDSCAPE PROBLEM

This section introduces the MNK-landscape problem, a multi-objective version of the NK fitness landscape problem. The NK model has found application in several fields, since evolutionary biology studies (KAUFFMAN; LEVIN, 1987; KAUFFMAN; WEINBERGER, 1989) to complex systems, such as the representation of complexity in economics and organizational sciences (VALENTE, 2014). Next sections present the model general formulation and related works addressing this problem.

4.2.1 GENERAL FORMULATION

The single *NK fitness landscapes* is a family of problems proposed by Kauffman (1993) in order to explore the way in which the neighborhood structure and the strength of the interactions between neighboring variables (subfunctions) is linked to the ruggedness of search spaces. For the given parameters, the problem consists in finding the global maximum of the function (SANTANA et al., 2015a).

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 (PELIKAN et al., 2009):

- 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 initialized randomly (PELIKAN et al., 2009).

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

$$z_{NK}(\mathbf{x}) = \sum_{q=1}^N f_q(x_q, \Pi(x_q)). \quad (30)$$

The mono-objective model has been extensively analyzed to study EAs and other heuristic algorithms (AGUIRRE; TANAKA, 2003; VEREL et al., 2011; TINÓS; YANG, 2014), such as EDAs (PELIKAN, 2008; PELIKAN et al., 2009; LIAW; TING, 2013). In Pelikan (2008) the author analyzed the performance of the hBOA (PELIKAN et al., 2003), the UMDA (MÜHLENBEIN; PAAB, 1996), and the simple genetic algorithm (GA) (HOLLAND, 1992) with several variants on randomly generated NK fitness landscapes. Pelikan et al. (2009) considered a class of NK landscapes with nearest neighbor interactions and tunable overlap. For both works, hBOA outperformed other compared algorithms especially on the most difficult instances with large neighborhood sizes.

Liaw and Ting (2013) studied the behaviors of EDAs with different model complexities, comparing the solution quality and convergence speed of UMDA, BMDA, and EBNA in the NK landscapes with different parameter settings. The authors provided some insights regarding that BMDA and EBNA can solve more complex problems.

The MNK-landscape (AGUIRRE; TANAKA, 2004, 2007) is a multi-objective combinatorial optimization problem with R objectives, where each objective function is determined by a different instance of a NK-landscape $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), z_2(\mathbf{x}), \dots, z_R(\mathbf{x})) : \mathcal{B}^N \rightarrow \mathcal{R}^R$, over the same binary string \mathbf{x} , where N is the number of variables, 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 size of the neighborhood in the r -th landscape.

MNK-landscape problem can be formulated as follows (AGUIRRE; TANAKA, 2007):

$$\begin{aligned} \max_{\mathbf{x}} \mathbf{z}(\mathbf{x}) &= (z_1(\mathbf{x}), \dots, z_R(\mathbf{x})) \\ \text{subject to } \mathbf{x} &\in \{0, 1\}^N, \end{aligned}$$

with (31)

$$\begin{aligned} z_r(\mathbf{x}) &= \frac{1}{N} \sum_{q=1}^N f_{r,q}(x_q, \Pi_{r,q}(x_q)), \\ r &\in \{1, \dots, R\}, \\ q &\in \{1, \dots, N\}, \end{aligned}$$

where the fitness contribution $f_{r,q}$ of variable x_q is a real number in $[0, 1]$ drawn from a uniform distribution. From this point forward the number of variables N will be called Q .

4.2.2 MOEA APPROACHES FOR MNK-LANDSCAPE

The MNK problem has been explored by Aguirre and Tanaka (2007), Verel et al. (2011), Santana et al. (2015b), Daolio et al. (2015).

Aguirre and Tanaka (2007) study how the parameters of the MNK-landscape influence several features of the fitness landscape, including the size of the Pareto front and the number of fronts. The authors provide some discussions related to the effects that parameters of the MNK-landscape on the problem complexity. They also compare NSGA-II and SPEA2 on the generated landscapes.

The paper presented by Verel et al. (2011, 2013) conducts a fitness landscape analysis on multi-objective NK-landscapes with objective correlation. In the ρ MNK-landscapes, the neighborhood structure is the same for all the objectives and the local fitness functions (subfunctions) are not independent. Instead, the parameters of these subfunctions follow a multi-variate uniform law defined by a correlation matrix C which is used to tune the correlations between the objectives. These authors investigated the effect of the correlations on characteristics of the Pareto front. In Verel et al. (2011), the authors enumerate small ρ MNK-landscape instances based on the Pareto dominance and studied, for example, the correlation degree between objective functions and the number of Pareto local optima found by a Pareto Local Search (PLS) algorithm.

More recently, Santana et al. (2015b) also propose an extension of the NK-landscape to model multi-objective problems (MOPs), providing a variety of configurations of the MNK-landscapes for investigating the effect of variables interactions on: (i) the shapes of the Pareto fronts, (ii) the correlations between the objectives, and (iii) the emergence of dependencies between the variables.

Liefooghe et al. (2015) study which (and how) problem features impact the search performance of a global EMO strategy (GSEMO) in the ρ MNK-landscapes. Daolio et al. (2015) extend the work comparing two dominance-based algorithms: the GSEMO and a PLS. This comparison analyzes the correlations between runtime and problem features, contrasting their association with search performance within and across instance classes, for both GSEMO and PLS algorithms.

Despite of the mono-objective version of this problem has been investigated using EDAs, there is no MOEDA addressing the multi-objective model. Our work considers the same MNK model used by Aguirre and Tanaka (2004), Aguirre and Tanaka (2007) and Santana et al. (2015b), described in Section 4.2.1.

Table 2 summarizes the main characteristics of MOEAs approaches to solve MNK-landscape problem.

Table 2: MOEAs to MNK-landscape problem main characteristics

MOEA	Objectives	Approach	Local Search	References
SPEA	2,3 and 5	Pareto	No	(AGUIRRE; TANAKA, 2007)
NSGA-II	2,3 and 5	Pareto	No	(AGUIRRE; TANAKA, 2007)
PLS	2,3 and 5	Pareto	Yes	(VEREL et al., 2011)
GSEMO	2,3 and 5	Pareto	No	(LIEFOOGHE et al., 2015)
PLS	2,3 and 5	Pareto	Yes	(LIEFOOGHE et al., 2015)

4.3 SUMMARY

This chapter has presented the MOPs addressed in this work. We have also presented related works for each MOP. For MOKP most used techniques solve the problem with an embedded LS procedure. However, some authors have recently proposed competitive MOEAs using scalarizing functions and indicator approaches. We have also shown that several probabilistic models can be used to represent the solution set for both knapsack and NK-landscape problems. In the next chapter we describe our proposed approach HMOBEDA, which jointly encodes variables, objectives and LS parameters in the same PGM model and will be tested on different MOKP and MNK-fitness landscape instances.

5 HYBRID MULTI-OBJECTIVE BAYESIAN ESTIMATION OF DISTRIBUTION ALGORITHM

In this chapter we present a hybrid EDA for solving combinatorial optimization problems. The term hybrid concerns a local search (LS) mechanism included into the EDA framework to improve its performance. The LS is combined with selection techniques usually adopted in MOEAs. In the next sections we present an overview of the proposed approach named Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA) which is based on the estimation of a probabilistic model (based on BN) of objectives, variables and local search parameters, with the last part (LS parameters) as the innovation of the proposed approach. First, we introduce a general schema and the solution encoding, next, we present the algorithmic structure of HMOBEDA including its initialization procedure, local search strategy, and probabilistic model (building and sampling). We also provide the main differences of this novel approach in comparison with related works.

5.1 GENERAL SCHEME

The general scheme of the adopted version of HMOBEDA (MARTINS et al., 2016) is presented in Figure 6.

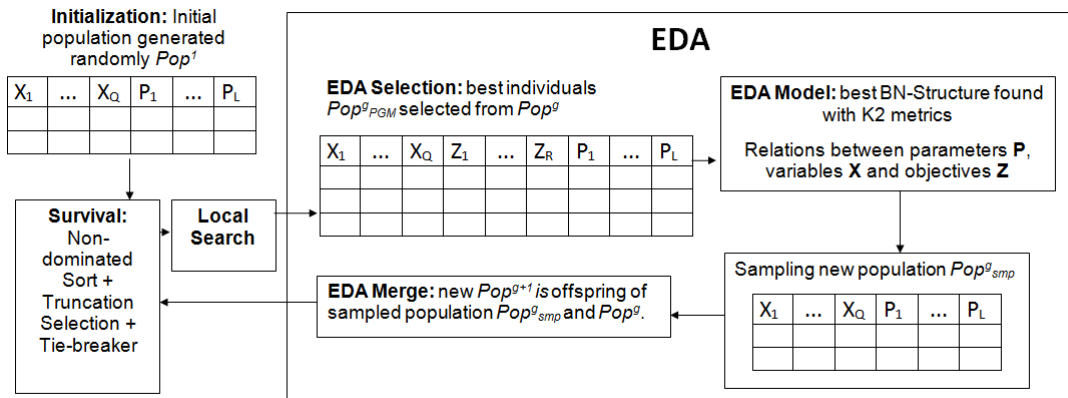


Figure 6: The HMOBEDA Framework.

The *Initialization process* randomly generates N solutions to compose the population at first generation (Pop^1).

The *Survival* block sorts individuals using the non-dominated sorting procedure (SRINIVAS; DEB, 1994) and a truncation selection is applied to select the N best solutions. Different tie-breaker criteria can be used in the HMOBEDA framework (i.e. Crowding Distance, hypervolume indicator, etc.)

After the *Survival* block, the *Local Search* block adopts an LS procedure. For every solution in Pop^g , LS generates a neighbor, calculates its fitness and updates the original solution in case the neighbor has a better fitness.

The *EDA Selection* block starts the PGM construction phase. A binary tournament selects N_{PGM} individuals from Pop^g . The procedure randomly selects two solutions and the one positioned in the best front is chosen. If they lie in the same front, it chooses that solution based on the tie-breaker criterion. Then, Pop_{PGM}^g is obtained encompassing N_{PGM} good individuals.

Aiming to learn the probabilistic model, the BN structure and parameters are estimated in the *EDA Model* block, encoding the joint probabilistic model of decision variables, objectives and LS parameters. Different algorithms can be considered for the *EDA Model* block: in this work we tested K2 and BN-HC as described in Section 3.3.4, but any other algorithm can be used in this block.

In the *Sampling* block, the obtained PGM is used to sample the set of new individuals (Pop_{smp}). In this case, not only decision variables, but also LS parameters can be sampled. This naive Bayesian model (see Figure 7) is adopted to facilitate the sampling process: fixing objective values as target evidences enables the estimation of their associated decision variables and LS parameters.

The union of the sampled population (Pop_{smp}) and the current population (Pop^g) in the *EDA Merge* 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.

5.2 SOLUTION ENCODING

Every individual is 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) . Subvectors \mathbf{x} , \mathbf{z} and \mathbf{p} can be specified as:

- \mathbf{x} is a binary subvector of decision variables, with element $x_q \in \{0, 1\}$, $q = 1 \dots Q$;

- \mathbf{z} is a subvector of objectives, with element z_r , $r = 1 \dots R$, representing the discrete value of the r^{th} objective, where $z_r \in \mathcal{R}$;
- \mathbf{p} is a subvector of elements, where each element p_l , $l = 1 \dots L$, $p_l \in \mathcal{R}$, indicates the value associated with an LS parameter. As will be detailed in Section 5.3.3, different parameters can be considered.

5.3 HMOBEDA FRAMEWORK

The main steps performed by HMOBEDA are described in Algorithm 3.

Algorithm 3: 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_{eval} , maximum number of solutions evaluation.

OUTPUT: ND , the final set of non-dominated solutions;

{Initialization}

- 1: $Pop^1(\mathbf{x}) = \text{RandomVector}(N, Q)$;
- 2: $Pop^1(\mathbf{z}) = \text{Fitness}(Pop^1(\mathbf{x}))$;
- 3: $Pop^1(\mathbf{p}) = \text{RandomVector}(N, L)$;
- 4: $e = 0$; { e is the current number of solutions evaluation}
- 5: $g = 1$; { g is the current generation}

{EDA: main loop}

- 6: **while** $e \leq Max_{eval}$ **do**
- 7: **if** $g > 1$ {EDA sampling} **then**
- 8: $Pop_{smp} = \text{Sampling}(PGM)$; {repair if necessary}
- 9: $Pop_{smp}(\mathbf{z}) = \text{Fitness}(Pop_{smp}(\mathbf{x}))$; {not surrogate assisted}
- 10: $Pop^g = \{Pop^{g-1} \cup Pop_{smp}\}$; {EDA survival}
- 11: **end if**

{Non-dominated Sorting}

{Defines Tot_F Pareto fronts from the best ($i = 1$) to the worst, and assigns a crowding distance}

- 12: $F_1 \dots F_{Tot_F} = \text{ParetoDominance}(Pop^g)$;
- 13: $Pop^g = \text{Select}(N, F_1 \cup \dots \cup F_{Tot_F})$; {Truncation Selection}

{Local search: performed at every β generations}

- 14: **if** $((g = 1) \text{ or } (g \bmod \beta = 0))$ **then**
- 15: $Pop_l = \text{LS}(Pop^g)$; {repair if necessary}
- 16: $Pop^g = \text{ParetoDominance}(Pop_l)$;
- 17: **else**
- 18: $Pop_l = \emptyset$;
- 19: **end if**

{EDA: learning the probabilistic model}

- 20: $Pop_{PGM}^g = \text{Selection}(N_{PGM}, Pop^g)$; {binary tournament}
- 21: $PGM = \text{ProbabilisticModelEstimation}(Pop_{PGM}^g)$;
- 22: $g = g + 1$;
- 23: **end while**
- 24: $ND = Pop^{g-1}(\mathbf{x})$;

The proposed algorithm makes use of specific strategies described in the next subsections.

5.3.1 INITIALIZATION

The block *Initialization process* randomly generates N subvectors \mathbf{x} and \mathbf{p} to compose the Pop^1 . For each subvector \mathbf{x} , the values of the corresponding objectives are calculated to form the subvector \mathbf{z} .

Therefore, N joint vectors ($\mathbf{y} = \mathbf{x}, \mathbf{z}, \mathbf{p}$) are obtained to compose the population in the first generation.

5.3.2 NON-DOMINATED SORTING

In order to sort individuals of the current population, the proposed approach uses the technique implemented by Srinivas and Deb (1994) and Deb et al. (2002), named Non-dominated Sorting, described in Appendix B.1.

Individuals are sorted taking at first their Dominance Ranking (DR) and secondly (in case of ties) the tie-breaker criterion. Finally, truncation selection takes place selecting the N best solutions (notice that at the first generation the entire population is selected)¹.

5.3.3 LOCAL SEARCH

Although HMOBEDA supports any LS procedures, in this work we adopted the Hill Climbing (HC) LS procedure (RUSSEL; NORVIG, 2003) due to its simplicity, aiming to improve a set of solutions selected by the truncation procedure. Algorithm 4 describes main steps of the local search procedure.

For every solution \mathbf{y} in Pop^g , HCLS generates a neighbor (\mathbf{nbh}), calculates its fitness and the neighbor \mathbf{nbh} updates the original solution in case it has a better fitness. As previously discussed, the subvector \mathbf{p} of \mathbf{y} defines the parameters associated with HCLS. In this work it can define, for example, how many iterations (N_{iter}) will be used, how each neighbor will be generated (T_{nbh}), and how to compute the neighbor fitness (T_{Fnbh}) in a mono-objective way - Linear Combination or Alternate of included objectives. This method of fitness calculation is based on Ahn et al. (2010), Li et al. (2004), Wang et al. (2012). In the Linear Combination method applied here the weights are uniformly distributed. For MOKP, if a neighbor is infeasible, the algorithm applies the same greedy repair method adopted by Zitzler and Thiele (1999) (See section 6.1.3 for details). The MNK-landscape problem addressed here has only domain constraints, therefore no repair procedure is necessary.

¹This implementation was adapted from Seshadri ().

Algorithm 4: Local Search (LS)

INPUT: Pop^g , a set of selected solutions
 N_{iter} , the maximum number of iterations;
 T_{Fnbh} , how to compute the neighbor fitness;
 T_{nbh} , neighborhood type;

OUTPUT: Pop_I , the set of improved solutions;

- 1: $t = 1$;
- 2: $Pop_I = Pop^g$;
- 3: **for** each $\mathbf{x}^k \in Pop_I$ **do**
- 4: **while** $t \leq N_{iter}$ **do**
 {Generate a $T_{nbh}(\mathbf{x}^k)$ type feasible neighbor}
- 5: $\mathbf{nbh} = T_{nbh}(\mathbf{x}^k)$;
 {Calculate fitness function f_I according $T_{Fnbh}(\mathbf{x}^k)$ }
- 6: $f_I(\mathbf{x}^k) = Fnbh(\mathbf{x}^k)$;
- 7: $f_I(\mathbf{nbh}) = Fnbh(\mathbf{nbh})$;
- 8: **if** $f_I(\mathbf{nbh}) > f_I(\mathbf{x}^k)$ **then**
- 9: $\mathbf{x}^k = \mathbf{nbh}$;
- 10: **end if**
- 11: $t = t + 1$;
- 12: **end while**
- 13: **end for**

In this work, HCLS is applied at every β generations (steps 13 to 18 in Algorithm 3). At the end of HCLS, Algorithm 3 provides a set of possibly improved solutions Pop_I .

5.3.3.1 BIT-FLIP AND SET MECHANISMS

In Algorithm 4, each neighbor is obtained based on two kinds of neighborhoods: bit-flip (BFp) and set (Set).

The bit-flip neighborhood considers the bit-blip operator (ARROYO, 2002). For every solution $\mathbf{x} = (x_1, \dots, x_q)$ investigated by the local search, we define $J_1(\mathbf{x}) = \{q | x_q = 1\}$ (the set of elements with bit 1) and $J_0(\mathbf{x}) = \{q | x_q = 0\}$ (the set of elements with bit 0). The positions $\nu \in J_1(\mathbf{x})$ and $\mu \in J_0(\mathbf{x})$ are randomly chosen.

A neighbor \mathbf{nbh} of \mathbf{x} with neighborhood type BFp is obtained according to the steps presented in Algorithm 5.

The Set neighborhood is based on a single bit-flip from 0 to 1 (SAMIR et al., 2015), (ARROYO et al., 2011). Each neighbor \mathbf{nbh} of \mathbf{x} is generated by setting a bit q , $1 \leq q \leq Q$, in a random position of the sequence.

Considering $J_0(\mathbf{x}) = \{q | x_q = 0\}$ (the set of elements with bit 0), a neighbor \mathbf{nbh} of \mathbf{x}

Algorithm 5: Bit-flip Neighborhood

```

1:  $v = \text{Random}(J_1)$ 
2:  $\mu = \text{Random}(J_0)$ 
3: end=false
4: Calculate  $\mathbf{nbh} = (nbh_1, \dots, nbh_Q)$ ,  $nbh_v = 0, nbh_\mu = 1$ ,
    $nbh_q = x_q, q = 1, \dots, Q, q \neq v, q \neq \mu$ 
5: while end=false do
6:   if  $\mathbf{nbh}$  is a feasible solution then
7:     return  $\mathbf{nbh}$ ;
8:     end=true;
9:   else
10:    Repair( $\mathbf{nbh}$ );
11:   end if
12: end while

```

with neighborhood type *Set* is obtained as described in Algorithm 6.

Algorithm 6: Set Neighborhood

```

1:  $\mu = \text{Random}(J_0(\mathbf{x}))$ 
2: end=false
3: Calculate  $\mathbf{nbh} = (nbh_1, \dots, nbh_Q)$ ,  $nbh_\mu = 1, nbh_q = x_q, q = 1, \dots, Q, q \neq \mu$ 
4: while end=false do
5:   if  $\mathbf{nbh}$  is a feasible solution then
6:     return  $\mathbf{nbh}$ ;
7:     end=true;
8:   else
9:     Repair( $\mathbf{nbh}$ );
10:  end if
11: end while

```

5.3.4 PROBABILISTIC MODEL

After the local search has been applied (and possible the repair method), HMOBEDA builds a BN (the PGM construction phase). A binary tournament selects N_{PGM} individuals from Pop^g , each one composed of Q decision variables, R objectives and L local search parameters, (i.e. $\mathbf{Y} = (Y_1, \dots, Y_M) = (Z_1, \dots, Z_R, X_1, \dots, X_Q, P_1, \dots, P_L)$). This way the BN structure encodes a factorization of the joint probability distributions or the probability mass function (pmf) given by:

$$p(\mathbf{y}) = \prod_{r=1}^R p(z_r | \mathbf{pa}_r^B) \cdot \prod_{q=1}^Q p(x_q | \mathbf{pa}_q^B) \cdot \prod_{l=1}^L p(p_l | \mathbf{pa}_l^B) \quad (32)$$

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 $\mathbf{Pa}_r^B = \emptyset$, $\mathbf{Pa}_q^B \subseteq \{Z_1, \dots, Z_R\}$ and $\mathbf{Pa}_l^B \subseteq \{Z_1, \dots, Z_R\}$. Therefore, aiming to learn the probabilistic model, the objective values collected from vector \mathbf{z} in Pop_{PGM}^s (selected in step 20 of Algorithm 3) are divided into sd_r discrete states². Considering that the BN model is estimated using the K2-metric (Equation 26) and the Bayesian estimate (Equation 23) with objectives as BN nodes, \mathbf{z} must also fit a multinomial distribution. Thus, to minimize the computational efforts to model B using all the possible discrete values (s_r), we have applied the discretization process with a limited number of possible values for each $z_r \in \mathbf{z}$ fixed as sd_r . In this case we assume the same sd_r value for each z_r node.

Figure 7 presents an example of a PGM structure estimated by HMOBEDA. 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}^s) it can also provide variables and LS parameters more related with good values of objectives fixed as evidence. This naive BN model is conceived to facilitate the sampling process: fixing objective values enables the estimation of their associated decision variables and LS parameters. A limitation of this approach is that the multinomial distribution encoded in the BN model cannot codify decision variables for permutation MOPs, and real numbers codification for BN nodes. In these cases, another probability distribution must be used.

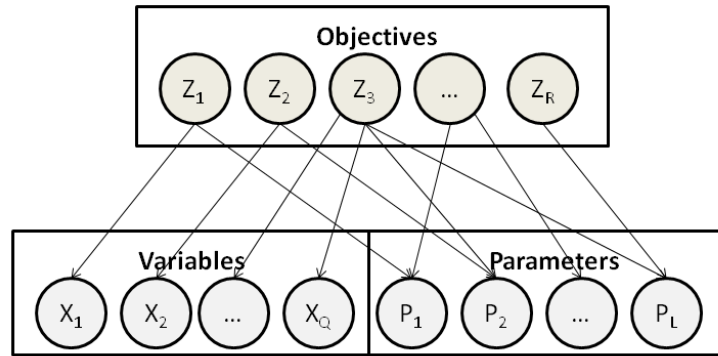


Figure 7: An example of the PGM structure used by HMOBEDA.

PGM 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, PGM

²The discretization process converts each objective value into sd_r discrete states considering the maximum possible value for each objective (Max_r). For each objective z_r , its discrete value is calculated as $zdr = \lceil z_r \cdot sd_r / Max_r \rceil$.

can also be used to sample the objective values) or MLE techniques can provide objective value approximations. New individuals are generated from the joint distribution encoded by the network using probabilistic logic sampling.

The probabilistic logic sampling (HENRION, 1986) method obtains a topological ordering of the nodes according to the Bayesian network structure (each node appears after its parent nodes). Therefore, in our work, the objective nodes appear before variable and LS parameter nodes, due to the restrictions imposed by BN structure. New solutions are generated by sampling the conditional probability distributions estimated for each node in the BN according to the computed ancestral ordering. For example, considering a variable or an LS parameter node denoted by Y_i , since all its parents appear before it in the ordering, all the parents will be already sampled at the time of sampling node it and therefore, the parameters of the conditional probability distribution of this node can be computed (KARSHENAS et al., 2014). Besides, as the BN encodes the dependencies between the variables, LS parameters and objectives, any information about objective values (evidences) can be inserted and propagated in the network in the sampling process, guiding the search, generating variables and LS parameters values that will result in good solutions.

A union of the sampled population (Pop_{smp}) and Pop^g is used to create the new population for the next generation $g + 1$, and the main loop HMOBEDA continues until the stop condition is fulfilled (maximum number of solutions evaluation). At any stage of the evolutionary process, if an infeasible solution is generated, HMOBEDA applies the repair method described in Section 6.1.1.

5.4 DIFFERENCES FROM THE LITERATURE

There are many particularities that can be identified within the proposed approach. The main differences of our work and those presented in Sections 4.1.3 and 4.2.2, 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 proposed by Schwarz and Ocenasek (2001b), their work presents a modified evaluation phase of the single BOA algorithm using the concept of strength criterion adopted by the SPEA algorithm, and their experiments are for 2 objectives. The work proposed by Laumanns and Ocenasek (2002) has been designed using a probabilistic model based on binary decision trees and a special selection scheme based on ϵ -archives, and it was applied to 2, 3 and 4 objectives. Differently, our work uses the

non-dominated sorting scheme, tournament selection and crowding distance (CD) (DEB et al., 2002) procedures as in NSGA-II. We adopt a local search, model the best solutions using a BN, but with objectives and local search parameters besides decision variables as nodes of the network, considering experiments with 2-5 and 8 objectives.

In Li et al. (2004), for MOKP, the following techniques were 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 work and ours are the probabilistic model and the local search: in their work the weights are based on the maximization and the minimization functions of each objective in the population. In our work the local search weights are uniformly distributed.

Gupta et al. (2014) proposed an EDA to find the solution of the BKP using the same repair method we have used in our work, but their addressed problem considers a single objective, a vector as the probabilistic model and a mutation probability exchange a bit from 0 to 1 or 1 to 0 applied to the sampled solutions in order to overcome a local optimum.

The work of Zhou et al. (2015) presents an EDA combining two local search strategies addressing the knapsack problem for a single objective. The main difference between this previous research and our work is that we address the embedded local search parameters in an EDA framework for MOPs. Zhou et al. (2015) present a relevant EDA approach with local search that might extended to the MOKP to be consider in our future research.

Regarding to the MNK-landscape problem, the main differences of our work and those presented in Section 4.2 are that none of them consideras EDAs for the multi-objective NK-landscape problem.

Pelikan (2008) and Pelikan et al. (2009) analyze the performance of hBOA, UMDA, and a genetic algorithm (GA) on the mono-objective NK-landscape problem. Liaw and Ting (2013) also study the behavior of EDAs using UMDA, BMDA and EBNA with different parameter settings. Although the authors use probabilistic model based on BN, such as in our work, they applied to the mono-objective NK-landscape problem.

Aguirre and Tanaka (2007) compare NSGA-II and SPEA2 on the generated landscapes using hypervolume indicator, similar to our work, and analyze the influence of the MNK-landscape parameters in several features of the fitness landscape. However, the main difference of that work and ours is that we aim to compare and study a PGM-based MOEDA, analyzing the probabilistic model. We also apply the non-dominated sort used as the selection scheme inspired by NSGA-II.

The works proposed by Liefvooghe et al. (2015) and Daolio et al. (2015) study which (and how) problem features impact the search performance of such approaches in the ρ MNK-landscapes for a global EMO strategy (GSEMO) and a neighborhood-based local search heuristic (PLS). The differences of these works and ours are that we explore an online parameter tuning of LS, where the LS parameters are tuned and self-adapted as the global optimization algorithm evolves.

Table 3 summarizes the main characteristics of EDA approaches to knapsack problem and NK-landscape problem compared with HMOBEDA.

Table 3: Summarizing EDAs approaches to the knapsack problem and NK-landscape problem

EDA	Problem	Objectives	Fitness assignment	Probabilistic Model	Local Search	References
BOA	MOKP	2	Pareto technique	Bayesian Network	No	(SCHWARZ; OCENASEK, 2001b)
BMOA	MOKP	2,3 and 4	$(\mu + \lambda, \epsilon)$ Operator	Binary decision tree	No	(LAUMANN; OCENASEK, 2002)
MOHEDA	MOKP	2,3 and 4	Scalarizing Functions	UMDA	Yes	(LI et al., 2004)
Greedy EDA	KP	1	Objective function	Probability Vector	No	(GUPTA et al., 2014)
Hybrid EDA	KP	1	Surrogate/Objective Function	UMDA	Yes	(ZHOU et al., 2015)
hBOA	NK	1	Objective function	hBOA	No	(PELIKAN, 2008)
UMDA	NK	1	Objective function	UMDA	No	(PELIKAN, 2008)
UMDA	NK	1	Objective function	UMDA	No	(LIAW; TING, 2013)
BMDA =	NK	1	Objective function	BMDA	No	(LIAW; TING, 2013)
EBNA	NK	1	Objective function	EBNA	No	(LIAW; TING, 2013)
HMOBEDA	MOKP	2-5 and 8	Pareto technique	Bayesian Network	Yes	(MARTINS et al., 2016)

Our work has thus intersections with other previous published works (BADER, 2009; KARSHENAS et al., 2014; LI et al., 2004; LIAW; TING, 2013). It is linked to the work presented in Karshenas et al. (2014) in which a joint probabilistic model of objectives and variables is proposed, and related to Li et al. (2004) by considering the weighted sum method in the fitness computation of each neighbor produced by the LS procedure. However the research presented in Li et al. (2004) does not explore the influence of LS parameters in comparison with off-line configuration techniques and the work described in Bader (2009) and Liaw and Ting (2013) does not consider LS parameters as nodes in the PGM.

5.5 SUMMARY

This chapter has introduced our proposal, named Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA), describing its general scheme and solution encoding. We also detailed the main steps and framework procedures. We have explained how a naive BN is modeled according to the population of solutions and we highlighted the main differences between our work and related approaches. We summarized these differences, which are more related with the LS embedded technique, the parameter tuning process, the BN as the PGM and finally, with the joint probabilistic model of objectives, variables and LS parameters. In the next chapter we investigate this proposal comparing different versions of the main framework, we also compare it with off-line configurations methods and with some state-of-the-art algorithms on a set of MOKP and NK-landscape problem instances.

6 EXPERIMENTS AND RESULTS

To analyze the performance of the proposed approach on multi and many objective combinatorial optimization, HMOBEDA is evaluated over a set of MOKP and MNK-landscape problem instances with different sizes and number of objectives.

The optimal Pareto front for each instance of the addressed problems is not known. So, we use a reference set, denoted by Ref , which is constructed by gathering all non-dominated solutions obtained by all algorithms over all executions. Two main convergence-diversity (JIANG et al., 2014) metrics, usually adopted for measuring the quality of the optimal solution set for multi and many objective optimization, are then considered: Hypervolume (HV^-) (ZITZLER; THIELE, 1999; BADER, 2009), Inverted Generational Distance (IGD) (VELDHUIZEN; LAMONT, 1999). In this work we also consider two capacity metrics: Overall Non-dominated Vector Generation (ONVG) (VELDHUIZEN; LAMONT, 2000) and Error Ratio (ER) (VELDHUIZEN; LAMONT, 1999). It is important to point out that different from the HV^- and IGD indicators that are considered at each execution of the algorithm (so their results are compared based on a statistical analysis over a set of executions), the capacity metrics are obtained at the end of all executions and measures the contribution of each algorithm to the reference front.

As discussed in Section 2.4, the hypervolume metric considers the difference (HV^-) between the hypervolume of the solution set of an algorithm and that of the reference set. The IGD metric is the average distance from each solution in the reference set to the nearest solution in the solution set. 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 ONVG metric is the number of non-dominated solutions from an approximated set, so larger values are preferred, and the Error Ratio considers the ratio of the solutions from the approximated set that are not in the reference set, so smaller values indicate better performances.

6.1 SETUP OF EXPERIMENTS

This section describes the assumptions assumed in the experiments regarding the instances of each problem (Section 6.1.1), the parameters of the different HMOBEDA versions (Section 6.1.2) and finally the parameters of the approaches being compared (Section 6.1.3).

6.1.1 PROBLEM INSTANCES

The experiments conducted in this section adopt the union of each set of MOKP instances considered in Ishibuchi et al. (2015), Tanigaki et al. (2014) and Zitzler and Thiele (1999). We use thus instances with 100 and 250 items, and 2 to 5 and 8 objectives. We characterize each instance as R - Q , where R is the number of objectives and Q is the number of items. The values of a'_q and b'_q are specified as integers in the interval $[10, 100]$. According to Zitzler and Thiele (1999), the capacity c_r is specified as 50% of the sum of all weights related to each knapsack r .

HMOBEDA implements a repair method when an infeasible solution is found for MOKP. The infeasible solution is transformed into a feasible one by removing items in an ascending order of the following values until the constraints conditions $\sum_{q=1}^Q b'_q x_q \leq c_r, r = 1, \dots, R$ and $x_q \in \{0, 1\}, q = 1, \dots, Q$ are satisfied:

$$q_j = \max \left(\frac{a_{rq}}{b_{rq}} \right), r = \{1, \dots, R\}, q = \{1, \dots, Q\} \quad (33)$$

That is, a feasible solution is generated from an infeasible one by removing items using the order of items specified by Equation 33. The method repairs the solution by removing items in an ascending order of the relation profit/weight until all the constraints conditions are satisfied.

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

6.1.2 HMOBEDA ALTERNATIVE VERSIONS

This section provides details regarding different HMOBEDA versions considered in the experiments. Figure 8 presents the HMOBEDA variations tested before we achieve the final standard HMOBEDA version (HMOBEDA_{EXT} highlighted in blue). All variations consider the

parameters as presented in Table 5.

Briefly, HMOBEDA considers the K2 algorithm as BN learning structure process, crowding distance as a tie-breaker for the selection scheme, online configuration of LS parameters, and the ideal points of the approximated Pareto front as candidates for the evidences in the root nodes. When the comparison is between HMOBEDA_{BN-HC} , HMOBEDA is called HMOBEDA_{K2} . In order to differentiate from HMOBEDA_{hype} , HMOBEDA receives the HMOBEDA_{CD} denomination. The same occurs when HMOBEDA is compared between off-line versions (HMOBEDA_{online}) and using different sampling methods (HMOBEDA_{IDEAL} versus HMOBEDA_{EXT} and HMOBEDA_{CPT}). In other words, HMOBEDA_{K2} , HMOBEDA_{CD} , HMOBEDA_{online} and HMOBEDA_{IDEAL} implement all the same algorithm.

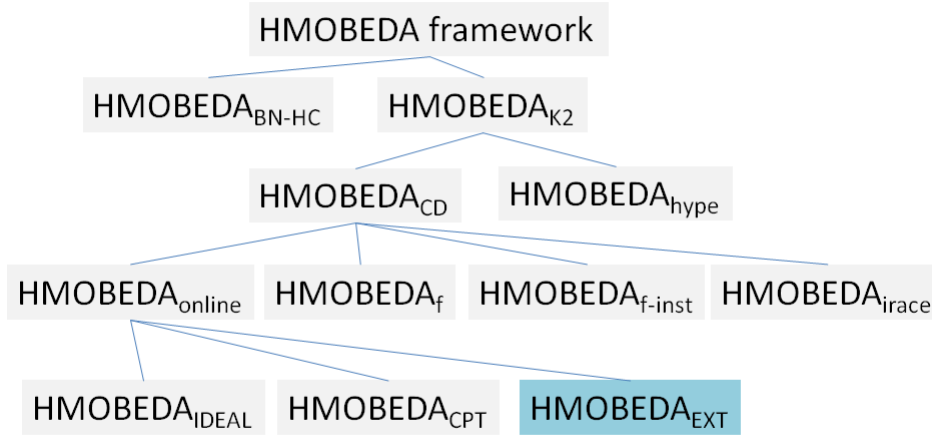


Figure 8: Different HMOBEDA versions considered in the experiments.

For the results obtained by HMOBEDA versions considered in Section 6.2.1.1 to 6.2.1.3, the sub-vector \mathbf{z} defined in Section 5.2 is fixed as evidence, set with target values guided by the ideal point Z^* ¹. However, it is possible to obtain a set of solutions and LS parameters associated with other specific objective values during the probabilistic logic sampling and Section 6.2.1.4 compares different ways to set the evidences.

First, in order to analyze the performance of HMOBEDA framework with different BN learning process we provide two versions: HMOBEDA_{BN-HC} and HMOBEDA_{K2} . HMOBEDA_{BN-HC} considers the BN-HC (Algorithm 2) as structure learning process and HMOBEDA_{K2} considers the K2 (Algorithm 1), both using K2 scoring metric. The BN

¹Usually high values for maximization problems: the ideal point Z^* is calculated as the maximum value of each objective obtained among all algorithms and executions.

learning process used by the best performance approach is then applied in the other two versions (HMOBEDA_{CD} and HMOBEDA_{hype}) aiming to evaluate the tie-breaker criterion for the selection scheme.

When two solutions tie on the dominance criterion, HMOBEDA_{CD} uses the crowding distance (CD) (DEB et al., 2002) and HMOBEDA_{hype} the hypervolume to decide which one is better. The fitness calculation of HMOBEDA_{hype} is based on a hypervolume approximation using Monte Carlo simulation, as presented in HypE algorithm proposed by Bader and Zitzler (2011)² (BADER; ZITZLER, 2011) due to the very expensive and time consuming computation of the exact hypervolume fitness (BADER; ZITZLER, 2011). This method generates random samples in the objective space and it counts the number of samples that are dominated by Pop^g . The hypervolume is approximated by the ratio between the dominated and total samples. In this work the number of samples used for Monte-Carlo approximation is 10,000.

This work aims to answer the question "What is the influence (on the HMOBEDA performance) of including LS parameters as BN nodes"? As previously discussed, this is a relevant question since the automatic and informed determination of the LS parameters can notably improve the efficiency of the search.

As an attempt to answer this question, we named HMOBEDA_{online} and modified this version providing three variants: HMOBEDA_f, HMOBEDA_{f-inst} and HMOBEDA_{irace}. All these variants have no LS parameter encoded as nodes of the PGM structure nevertheless they also adopt the same parameters values (described in Table 5) and all the remaining characteristics considered in HMOBEDA.

The LS online configuration adopted by HMOBEDA_{online} 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 bit-flip and (2) set.

In the LS off-line configuration adopted by the modified algorithms, the first version, HMOBEDA_f, considers $N_{iter} = 19$, $T_{Fnbh} = 0$ and $T_{nbh} = 1$. These values represent the most frequent value of each LS-parameter provided by HMOBEDA in the set of non-dominated solutions, considering all the instances and executions. HMOBEDA_{f-inst} considers the same rule (i.e., the most frequent value found in all HMOBEDA executions) but now separated for each instance, as presented in Table 4. HMOBEDA_{irace} considers $N_{iter} = 14$,

²This implementation is used by Bader and Zitzler (2011) for more than three objectives and can be downloaded from <http://www.tik.ee.ethz.ch/sop/download/supplementary/hype/>.

$T_{Fnbh} = 0$ and $T_{nbh} = 1$. The LS off-line configuration method used for HMOBEDA_{irace} is I/F-Race (BIRATTARI et al., 2010), which is a state-of-the-art automatic configuration method. We use the implementation of I/F-Race provided by the *irace* package (LÓPEZ-IBÁÑEZ et al., 2011). As presented by López-Ibáñez and Stützle (2012), it performs the configuration process using the hypervolume (HV^-) as the evaluation criterion. Additionally, we consider a budget of 10,000 experiments for *irace*.

Table 4: HMOBEDA_{f-inst} parameters.

	Value	Instances
T_{Fnbh}	0	all instances
T_{nbh}	1	all instances
N_{iter}	18	3-100, 2-250 and 4-250
N_{iter}	10	3-250
N_{iter}	19	4-100
N_{iter}	16	2-100 and 5-100
N_{iter}	20	8-100, 5-250 and 8-250

The same instances are used for training and testing off-line versions. The results obtained by HMOBEDA_f , HMOBEDA_{f-inst} and HMOBEDA_{irace} are, thus, quite better than would be expected if the test instances were different. In this case, we assure that all algorithms are well adjusted for the instances considered in the training/testing phase (no one is privileged).

In Section 6.2.1.4 we compare three variants named HMOBEDA_{IDEAL} , HMOBEDA_{EXT} and HMOBEDA_{CPT} which represent different alternatives to guide the search (see Figure 9). The main difference among the approaches concerns the *Sampling* procedure. Roughly speaking, the main characteristic of the HMOBEDA_{IDEAL} is that it uses an estimation of the ideal point Z^* (as depicted in Figure 9a) to guide the search. This is achieved by fixing discretized values of Z^* achieved so far, as evidences in the root nodes of the BN structure (i.e., in nodes Z_1, \dots, Z_R of Figure 7) during the probabilistic logic sampling.

As shown in Figure 9c, HMOBEDA_{CPT} uses the priori probabilities of Z_1, \dots, Z_R described in the Conditional Probability Table (CPT) to draw and further fix their evidences; and 9b shows that HMOBEDA_{EXT} , besides considering Z^* like the HMOBEDA_{IDEAL} version, it includes the extremes points of the approximated Pareto front as candidates for the evidences in the root nodes, i.e., the ideal point Z^* plus the estimated extreme points of the current approximation of the Pareto front. These values are fixed in each generation considering the same probabilities for all the candidates: Z^* and the total number of objectives (each in every extreme of the current approximation Pareto front).

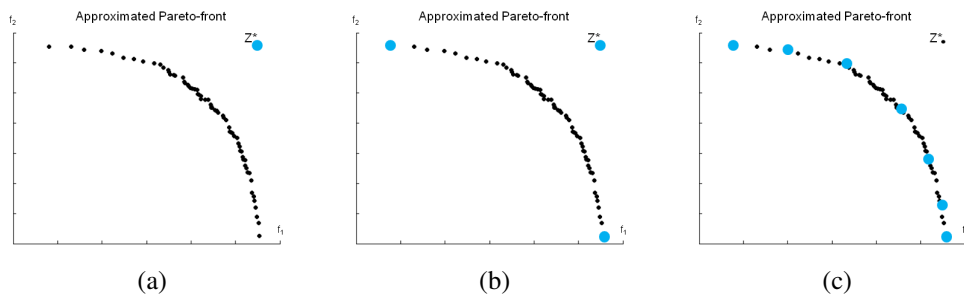


Figure 9: Evidences (blue circles) from the approximated Pareto front with 2 objectives for (a) HMOBEDA_{IDEAL}, (b) HMOBEDA_{EXT} and (c) HMOBEDA_{CPT}.

6.1.3 CUTTING EDGE EVOLUTIONARY APPROACHES FROM THE LITERATURE

After comparing HMOBEDA (in its different versions), and then achieving its best version (HMOBEDA_{EXT}), we compare it with, MBN-EDA (KARSHENAS et al., 2014), NSGA-II (DEB et al., 2002), S-MOGLS (ISHIBUCHI et al., 2008) (NSGA-II with local search), MOEA/D (ZHANG; LI, 2007) and NSGA-III (DEB; JAIN, 2014).

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. MOEA/D and NSGA-III are implemented in C++, and the remaining algorithms in Matlab. Details of these algorithms are provided in Appendix B. All algorithms are run on an AMD Opteron Processor 6378 server, CPU 2.40GHz machine with 125 GB of RAM, running Linux.

The parameters for each algorithm are shown in Table 5.

Table 5: Parameters of the MOEAs used for solving the MOKP and MNK-landscape problem.

	Description	Value	Algorithm
N	Population size	100	HMOBEDA, MBN-EDA, NSGA-II, S-MOGLS
β	Local Search	1	HMOBEDA, S-MOGLS
N_{PGM}	Pop_{PGM}^s size	$N/2$	HMOBEDA, MBN-EDA, NSGA-II, S-MOGLS
N_{smp}	Pop_{smp}^s size	$10 * N$	HMOBEDA, MBN-EDA
sd_r	Total of disc. states	10	HMOBEDA
$p_c r$	Uniform Cross Prob.	0.8	NSGA-II, S-MOGLS, NSGA-III, MOEA/D
p_{mut}	Bit Flip Mutat. Prob.	1/500	NSGA-II, S-MOGLS, NSGA-III, MOEA/D

As discussed by Ishibuchi et al. (2008), 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 Deb and Jain (2014), i.e., the number of reference points (H) defines the population size N . For R -objective functions, if p divisions are considered along each objective, Deb and Jain (2014) define $H = C_{R+p-1}^p$. 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 Zhang and Li (2007) (this is the same procedure used to generate the reference points in NSGA-III). As discussed by Deb and Jain (2014), the size of the neighborhood for each weight vector is

$T = 10$. MOEA/D considers the weighted sum approach ³.

The K2 algorithm is used to learn the BN structure in MBN-EDA and HMOBEDA. The K2 inputs are the ordering of variables, which presents a relationship between a node and its parents (ascendent nodes), and the maximum number of parents for each node (all the objectives nodes in our case). For HMOBEDA we consider crowding distance as a tie-breaker for the selection scheme, online configuration of LS parameters, and both the ideal and extremes points of the approximated Pareto front as candidates for the evidences in the root nodes. We also fix the objectives as parents in the network.

Due to heterogeneous 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.

The next section presents the results obtained by comparing the HMOBEDA versions to each other and the best HMOBEDA version with cutting edge evolutionary approaches from the literature.

6.2 RESULTS

Section 6.2.1 compares HMOBEDA versions in order to define a final standard version to be compared with the literature. As previously discussed, Section 6.2.1.1 investigates HMOBEDA_{BN-HC} and HMOBEDA_{K2} in order to fix the best algorithm to estimate the BN structure. In Section 6.2.1.2 we compare HMOBEDA_{CD} with HMOBEDA_{hype}, regarding the tie-breaker criterion. An analysis of the LS-parameters influence in the PGM structure is presented in Section 6.2.1.3, considering HMOBEDA_{online}, HMOBEDA_f, HMOBEDA_{f-inst} and HMOBEDA_{irace}. Section 6.2.1.4 compares different ways to set evidences during the sampling by considering HMOBEDA_{IDEAL}, HMOBEDA_{EXT} and HMOBEDA_{CPT}. Once the standard HMOBEDA version is defined, in 6.2.1, we extend the comparisons to cutting edge evolutionary algorithms in Section 6.2.2. Finally we discuss, in Section 6.2.3, the PGM structures learned during the evolutionary process.

In this section we used the Shapiro-Wilk normality test (CONOVER, 1999) to verify the performance metrics results are normally distributed. Then, in the case of non-normal

³This approach is usual for MOKP (ZHANG; LI, 2007), and can be downloaded from <http://http://dces.essex.ac.uk/staff/zhang/webofmoead.htm> It is also suggested by Ishibuchi et al. (2015).

distribution, the Mann-Whitney-Wilcoxon test is applied for statistical analysis (CASELLA; BERGER, 2002) of the results when we have two approaches being compared, and the Kruskal-Wallis test (CASELLA; BERGER, 2002) for three or more approaches.

For all the tables in this section, the numbers in parentheses show the results of the pairwise comparisons using Dunn-Sidak's post-hoc test. 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 the entries present no statistically significant difference for the compared approaches the background is emphasized in light blue. The entry related to the algorithm with the lowest (best) average metric is highlighted in bold. All tests have been executed with a confidence level of 95% ($\alpha = 5\%$).

6.2.1 COMPARING ALTERNATIVE VERSIONS OF HMOBEDA

First we investigate the structure learning process used in HMOBEDA_{K2} and HMOBEDA_{BN-HC} . After that we analyze the use of CD versus hypervolume as tie-breaker criterion comparing HMOBEDA with HMOBEDA_{hype} . Then we proceed with the analysis by evaluating the influence of LS online configuration based on BN nodes, comparing the original HMOBEDA with its off-line configured versions. After that we explore how the evidence setting can influence the search by comparing three different versions: HMOBEDA_{IDEAL} , HMOBEDA_{EXT} and HMOBEDA_{CPT} .

6.2.1.1 BN STRUCTURE ESTIMATION: HMOBEDA_{K2} X HMOBEDA_{BN-HC}

To fix the best algorithm to estimate the BN structure, we compare HMOBEDA_{K2} with HMOBEDA_{BN-HC} based on Mann-Whitney-Wilcoxon statistical test. Table 6 shows the statistical analysis of pairwise comparisons between HMOBEDA_{K2} and HMOBEDA_{BN-HC} for each instance with respect to HV^- and IGD values, respectively.

Table 6: Results for pairwise comparison between HMOBEDA_{K2} and HMOBEDA_{BN-HC} using Mann-Whitney-Wilcoxon test 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^-)									
HMOBEDA_{K2}	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,1)	(0,0)	(0,1)	(0,0)	(0,0)
HMOBEDA_{BN-HC}	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(0,0)	(1,0)	(0,0)	(0,0)
	IGD metric									
HMOBEDA_{K2}	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(0,0)	(0,0)	(0,0)
HMOBEDA_{BN-HC}	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,1)	(0,0)	(0,0)	(0,0)

The results reveal statistically significant differences, then the null hypothesis assuming that all HV^- values come from the same distribution can be rejected for all instances.

The same hypothesis can be rejected in the case of IGD. Then the post hoc analysis is performed.

Based on Table 6, we conclude that HMOBEDA_{K2} presents better results than HMOBEDA_{BN-HC} for both 4-250 and 5-250 instances regarding HV^- indicator. However, HMOBEDA_{BN-HC} has the best IGD metric for 5-100 instance. There is no statistically significant differences between HMOBEDA_{K2} and HMOBEDA_{BN-HC} for the remaining instances regarding both HV^- and IGD metric.

We extend the analysis comparing the average run time over 30 executions for each approach, as presented in Table 7.

Table 7: Average Run time (minutes) 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_{K2}	2.45	5.19	5.31	14.58	9.15	24.24	23.54	52.48	34.24	65.54
HMOBEDA_{BN-HC}	5.51	10.26	11.05	28.16	16.48	42.15	40.17	91.78	57.67	103.57

Table 8 presents the runtime statistical analysis using Mann-Whitney-Wilcoxon test with $\alpha = 5\%$ for each problem instance.

Table 8: Results for pairwise run time comparisons between HMOBEDA_{K2} and HMOBEDA_{BN-HC} using Mann-Whitney-Wilcoxon test 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_{K2}	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)
HMOBEDA_{BN-HC}	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)

We can see that HMOBEDA with K2 is faster than using BN-HC for all instances. For this reason, we define K2 structure learning algorithm and $\text{HMOBEDA} = \text{HMOBEDA}_{K2}$ as our current standard version to be compared in the next section.

6.2.1.2 TIE-BREAKER CRITERION: HMOBEDA_{CD} X HMOBEDA_{HYPE}

In this section we compare HMOBEDA_{CD} with HMOBEDA_{hype} to test crowding distance and the hypervolume as the tie-breaker criterion. In Table 9, we can see that HMOBEDA_{CD} presents statistically significant differences in comparison with HMOBEDA_{hype} for 2-250, 5-100, 8-100 and 8-250, and regarding only IGD metric, for 2-100, 3-100, 3-250, 4-100 and 4-250 instances. There is no statistically significant differences between HMOBEDA_{CD} and HMOBEDA_{hype} for instances 2-100, 3-100, 3-250, 4-100, 4-250 and 5-250 for HV^- metric. Regarding IGD metric, HMOBEDA_{CD} 's results present statistically significant differences for almost all instances, except in 5-250, where there is no statistically significant differences, however, HMOBEDA_{CD} presents the lowest metric value.

Table 9: Results for pairwise comparisons between HMOBEDA_{CD} and HMOBEDA_{hype} using Mann-Whitney-Wilcoxon test 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 ⁻)									
HMOBEDA _{CD}	(0,0)	(0,1)	(0,0)	(0,0)	(0,0)	(0,0)	(0,1)	(0,1)	(0,1)	(0,1)
HMOBEDA _{hype}	(0,0)	(1,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(1,0)	(1,0)	(1,0)
	IGD metric									
HMOBEDA _{CD}	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,0)	(0,1)	(0,1)
HMOBEDA _{hype}	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(0,0)	(1,0)	(1,0)

We also analyze the average run time over 30 executions for HMOBEDA_{CD} and HMOBEDA_{hype}, presented in Table 10, and its respective statistical analysis, in Table 11, using Mann-Whitney-Wilcoxon test with $\alpha = 5\%$ for each problem instance.

Table 10: Average run time (minutes) 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 _{CD}	2.45	5.19	5.31	14.58	9.15	24.24	23.54	52.48	34.24	65.54
HMOBEDA _{hype}	3.56	6.98	6.78	17.01	12.38	32.45	34.82	65.32	42.43	75.18

Table 11: Results for pairwise run time comparisons between HMOBEDA_{CD} and HMOBEDA_{hype} using Mann-Whitney-Wilcoxon test 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 _{CD}	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)	(0,1)
HMOBEDA _{hype}	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)

We can conclude that despite using less computational resources (HMOBEDA_{CD} has the lowest average runtime for all instances), HMOBEDA_{CD} is competitive with HMOBEDA_{hype} and we will assume HMOBEDA=HMOBEDA_{CD} for the next comparisons.

6.2.1.3 LS PARAMETER TUNING: ONLINE X OFF-LINE VERSIONS

We can now analyze one of the main contributions of the proposed approach, i.e., the influence of including LS-parameters in the PGM structure considering four different versions: HMOBEDA_{online}, HMOBEDA_f, HMOBEDA_{f-inst}, HMOBEDA_{irace}, where the first is considered the online parameter tuning version and the last ones are the off-line versions. The results are shown in Table 12. A post-hoc analysis is performed to evaluate which algorithms present statistically significant differences.

HMOBEDA_{online} shows statistically significant differences in comparison with its off-line modified versions for almost all instances, particularly those with high number of objectives and variables (4-250, 5-100, 8-100 and 8-250), where HMOBEDA_{online} is better than the other three algorithms. There is no statistically significant differences between HMOBEDA_{online} and HMOBEDA_{irace} for instance 4-100 for both HV⁻ and IGD values. For

all instances with 4, 5 and 8 objectives, HMOBEDA_{online} is better than HMOBEDA_{f-inst} and HMOBEDA_f .

Table 12: Results for pairwise comparisons between HMOBEDA_{online} and its off-line 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
	Hypervolume differences (HV ⁻)									
HMOBEDA_{online}	(0,3)	(1,2)	(0,3)	(0,3)	(0,2)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)
HMOBEDA_{f-inst}	(1,0)	(3,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(3,0)	(1,1)
HMOBEDA_{irace}	(1,0)	(0,3)	(1,0)	(1,0)	(0,0)	(1,0)	(1,0)	(1,0)	(1,2)	(1,1)
HMOBEDA_f	(1,0)	(2,1)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(2,1)	(3,0)
	IGD metric									
HMOBEDA_{online}	(0,0)	(1,2)	(0,3)	(0,1)	(0,2)	(0,3)	(0,3)	(0,2)	(0,3)	(0,3)
HMOBEDA_{f-inst}	(0,0)	(3,0)	(1,0)	(0,0)	(1,0)	(1,0)	(1,0)	(2,0)	(3,0)	(1,1)
HMOBEDA_{irace}	(0,0)	(0,3)	(1,0)	(1,0)	(0,0)	(1,0)	(1,0)	(0,2)	(2,1)	(1,1)
HMOBEDA_f	(0,0)	(2,1)	(1,0)	(0,0)	(1,0)	(1,0)	(1,0)	(2,0)	(1,2)	(3,0)

The average run time for each algorithm is presented in Table 13.

Table 13: Average run time (minutes) 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_{online}	2.45	5.19	5.31	14.58	9.15	24.24	23.54	52.48	34.24	65.54
HMOBEDA_{f-inst}	2.12	5.23	5.09	14.15	9.86	24.87	21.25	53.12	32.12	66.43
HMOBEDA_{irace}	2.67	5.01	5.17	13.87	9.43	24.64	23.12	52.76	33.21	65.01
HMOBEDA_f	2.59	4.91	5.88	13.98	9.51	24.40	23.87	51.42	34.51	65.24

We can observe in Table 13 that the computational efforts for each algorithm are in the same time scale, and increasing the number of objectives and variables impacts the average computational time for all approaches. We also include statistical tests in order to compare the run time for all instances considering the total number of executions. Table 14 shows the results for pairwise comparisons among HMOBEDA run time and all other algorithms. The entry with the lowest average run time is emphasized in bold.

Table 14: Results for pairwise run time comparisons among HMOBEDA_{online} and the off-line 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_{online}	(0,1)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,1)
HMOBEDA_{f-inst}	(0,1)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(3,0)
HMOBEDA_{irace}	(3,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,1)
HMOBEDA_f	(0,1)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,1)

Table 14 shows that HMOBEDA_{online} and off-line variations do not present statistically significant differences between its execution times for almost all instances. HMOBEDA_{irace} presents the worst results for 2-100 instance, as well as HMOBEDA_{f-inst} for 8-250 instance.

These results justify, in our opinion, the relevance of adding LS parameters into the probability model, and we assume that HMOBEDA_{online} is better or at least equal to its modified

versions with LS off-line configuration. So, we define $\text{HMOBEDA} = \text{HMOBEDA}_{\text{online}}$ as our current standard version, i.e. the version using K2 for BN structure learning in the *EDA Model* block, CD as tie-breaker criterion with a joint BN of decision variables, objectives and online LS parameters setting.

6.2.1.4 SETTING EVIDENCE: $\text{HMOBEDA}_{\text{IDEAL}}$ X $\text{HMOBEDA}_{\text{EXT}}$ X $\text{HMOBEDA}_{\text{CPT}}$

We aim to investigate in this section how the evidences may influence the convergence and distribution of non-dominated solutions along the approximated Pareto front. In addition to HV^- and IGD indicators, this section aims to evaluate the quality of the final approximation of the Pareto front 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. Three versions are considered: $\text{HMOBEDA}_{\text{IDEAL}}$, $\text{HMOBEDA}_{\text{EXT}}$ and $\text{HMOBEDA}_{\text{CPT}}$.

Table 15 shows the statistical analysis of pairwise comparisons between $\text{HMOBEDA}_{\text{IDEAL}}$, $\text{HMOBEDA}_{\text{EXT}}$ and $\text{HMOBEDA}_{\text{CPT}}$ for each instance with respect to HV^- and IGD values, respectively.

Table 15: Results for pairwise comparisons between $\text{HMOBEDA}_{\text{IDEAL}}$, $\text{HMOBEDA}_{\text{EXT}}$ and $\text{HMOBEDA}_{\text{CPT}}$ 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^-)										
$\text{HMOBEDA}_{\text{IDEAL}}$	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
$\text{HMOBEDA}_{\text{EXT}}$	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
$\text{HMOBEDA}_{\text{CPT}}$	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
IGD metric										
$\text{HMOBEDA}_{\text{IDEAL}}$	(0,0)	(0,1)	(0,1)	(0,1)	(0,0)	(0,0)	(1,1)	(1,1)	(1,0)	(1,0)
$\text{HMOBEDA}_{\text{EXT}}$	(0,0)	(0,1)	(0,1)	(0,1)	(0,0)	(0,0)	(0,2)	(0,2)	(0,2)	(0,2)
$\text{HMOBEDA}_{\text{CPT}}$	(0,0)	(2,0)	(2,0)	(2,0)	(0,0)	(0,0)	(2,0)	(2,0)	(1,0)	(1,0)

Although Table 15 shows that there is no statistically significant differences between $\text{HMOBEDA}_{\text{IDEAL}}$, $\text{HMOBEDA}_{\text{CPT}}$ and $\text{HMOBEDA}_{\text{EXT}}$ regarding HV^- indicator for all instances, there are statistically significant differences regarding the IGD metric for 5-100, 5-250, 8-100 and 8-250 instances. $\text{HMOBEDA}_{\text{IDEAL}}$ and $\text{HMOBEDA}_{\text{EXT}}$ present better results than $\text{HMOBEDA}_{\text{CPT}}$ for 2-250, 3-100, 3-250, 5-100 and 5-250 with some advantage for $\text{HMOBEDA}_{\text{EXT}}$. Besides, $\text{HMOBEDA}_{\text{EXT}}$ is the best approach for 8-100 and 8-250 instances.

The average run time between the algorithms are presented in Table 16, with the respective post-hoc test showed in Table 17. We can observe that the computational efforts for each algorithm are similar, as expected, keeping the runtime 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 minutes (instance 2-100) to 66 minutes (instance 8-250) for the three HMOBEDA versions, with no statistically significant differences between them for the same instance.

Table 16: Average run time (minutes) 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 _{IDEAL}	2.45	5.19	5.31	14.58	9.15	24.24	23.54	52.48	34.24	65.54
HMOBEDA _{EXT}	2.42	5.34	5.39	14.27	9.84	24.51	23.49	52.54	34.27	65.13
HMOBEDA _{CPT}	2.39	5.26	5.15	14.48	9.57	24.58	23.98	52.84	34.21	65.27

Table 17: Results for pairwise run time comparisons between HMOBEDA_{IDEAL}, HMOBEDA_{CPT} and HMOBEDA_{EXT} 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 _{IDEAL}	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
HMOBEDA _{EXT}	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
HMOBEDA _{CPT}	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)

In order to analyze the final approximated Pareto front achieved by each HMOBEDA version taking into account the information available in the final PGM, we calculate, for each possible solution in the approximated Pareto front, the pmf $P(\mathbf{y})$, defined by Equation (32), 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 set is 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 $P(Z_1 = z_1, \dots, Z_R = z_R)$.

Figure 10 presents the analysis of the approximated Pareto front for the instance 2-100 considering the probabilistic information of the PGM. The approximated Pareto front obtained by each algorithm is calculated gathering all non-dominated solutions over 30 executions. We can observe that HMOBEDA_{IDEAL} and HMOBEDA_{CPT} provide solutions concentrated on a particular region of the Pareto front (around the ideal point), region named Pareto front knee. However, for HMOBEDA_{EXT} solutions are better distributed, with the extreme point associated with objective Z_1 presenting higher probabilities than those for HMOBEDA_{IDEAL} and HMOBEDA_{CPT}.

On the other hand, in Figure 11 for instance 2-250, HMOBEDA_{IDEAL} provides points well distributed along the entire front. HMOBEDA_{EXT} presents solutions concentrated around $\max(z_2)$ and ideal point. While, HMOBEDA_{CPT} provides the worst Pareto front, with less points as some of them located near $\max(z_1)$.

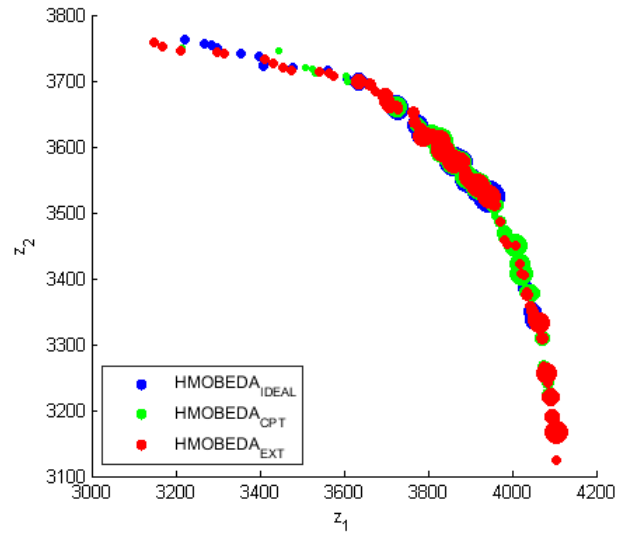


Figure 10: A probabilistic view of the approximated Pareto front for 2 objectives and 100 items for HMOBEDA_{IDEAL} , HMOBEDA_{EXT} and HMOBEDA_{CPT} .

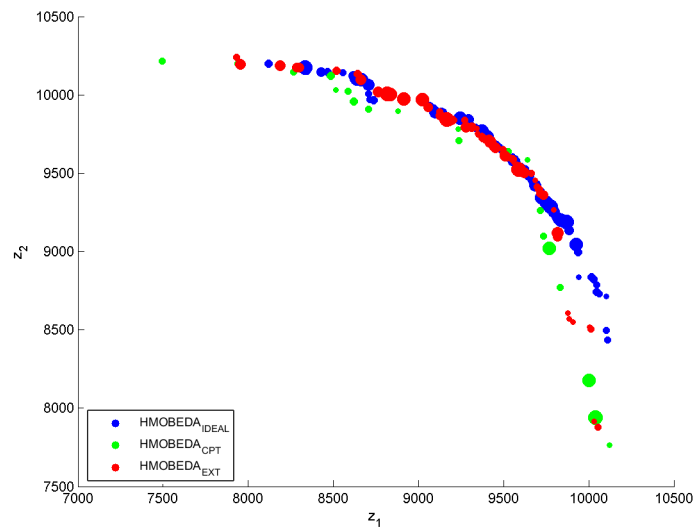


Figure 11: A probabilistic view of the approximated Pareto front for 2 objectives and 250 items for HMOBEDA_{IDEAL} , HMOBEDA_{EXT} and HMOBEDA_{CPT} .

Since it is not possible to visualize the Pareto front for more than two objectives, Figures 12 to 16 illustrate the ordered Euclidean distance between each point from the approximated Pareto front and the ideal point. We note that HMOBEDA_{IDEAL} presents the smoothest distance fluctuation, except in Figure 12a for instance 2-100, where all the approaches present similar Euclidean distances. This observation, associated with the IGD values in Table 18, indicates that HMOBEDA_{IDEAL} provides an approximation of the Pareto front with a slightly high concentration around the ideal point.

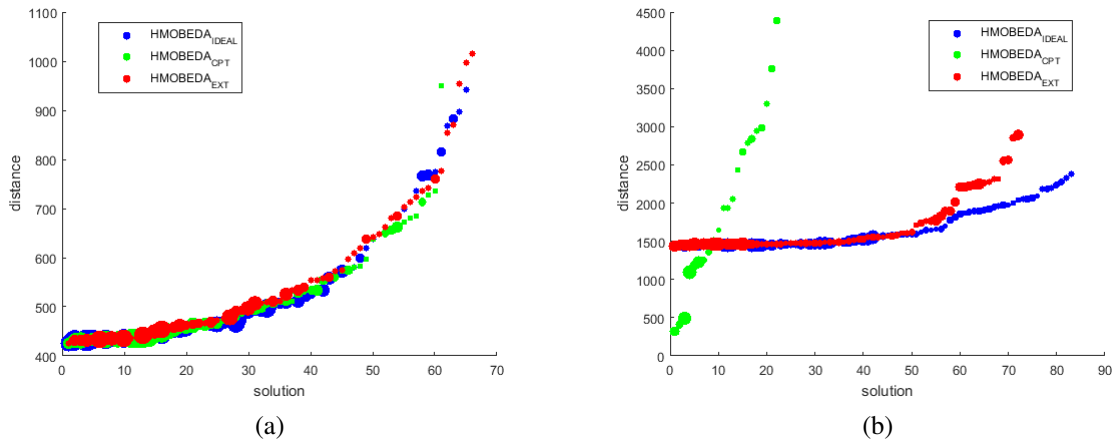


Figure 12: The Euclidean distance between each solution and the ideal point for 2 objectives with (a) 100 items and (b) 250 items.

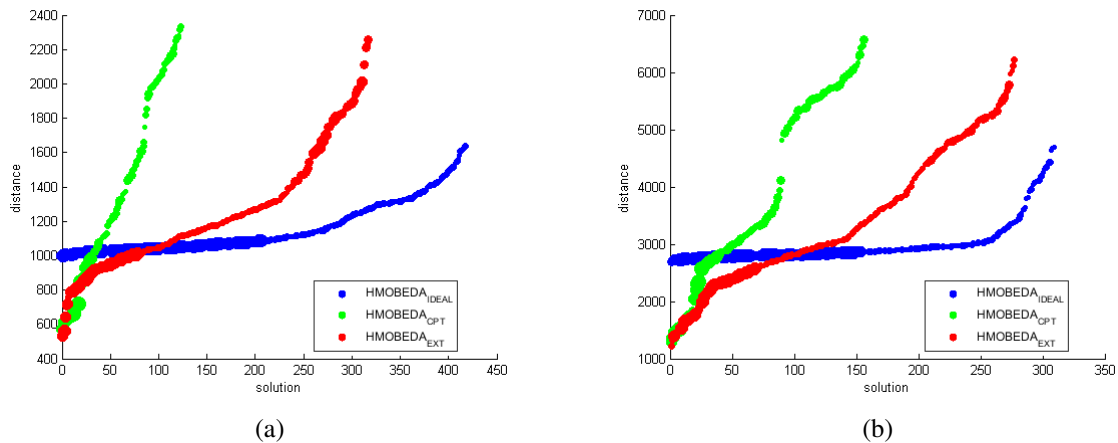


Figure 13: The Euclidean distance between each solution and the ideal point for 3 objectives with (a) 100 items and (b) 250 items.

Notice that each solution/circle in the graph is proportional to its marginal probability $P(Z_1 = z_1, \dots, Z_R = z_R)$. Therefore, in Figures 12 to 14, we can conclude that for 2, 3 and 4 objectives, the solutions present different probabilities along the Pareto front, concentrated around the ideal point, as also noticeable in Figures 10 and 11.

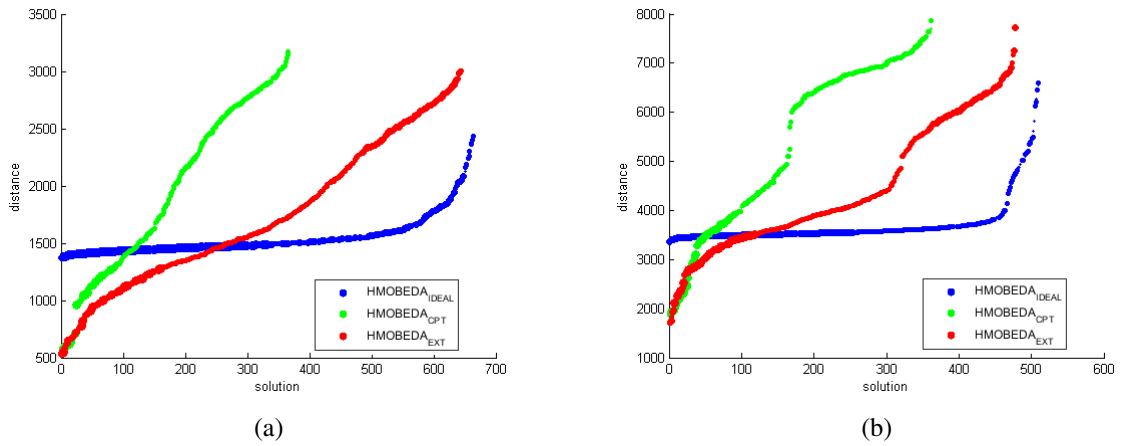


Figure 14: The Euclidean distance between each solution and the ideal point for 4 objectives with (a) 100 items and (b) 250 items.

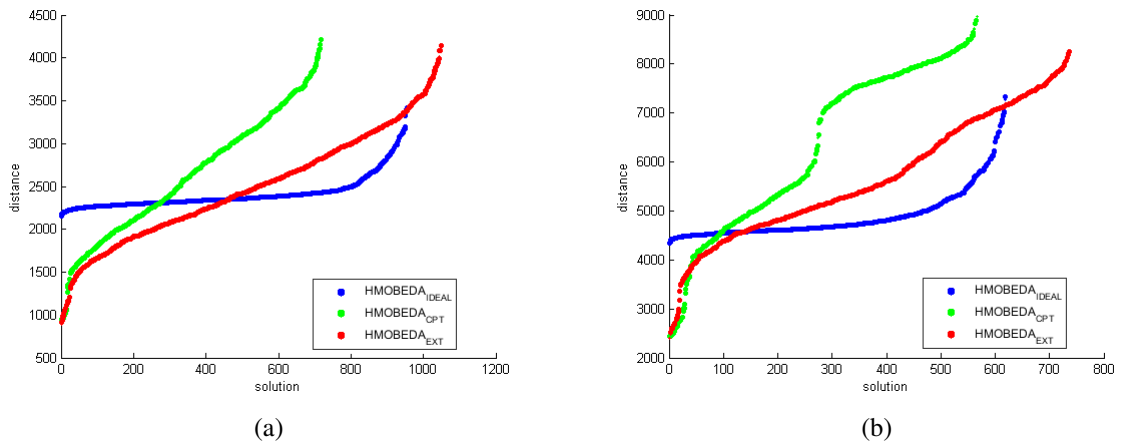


Figure 15: The Euclidean distance between each solution and the ideal point for 5 objectives with (a) 100 items and (b) 250 items.

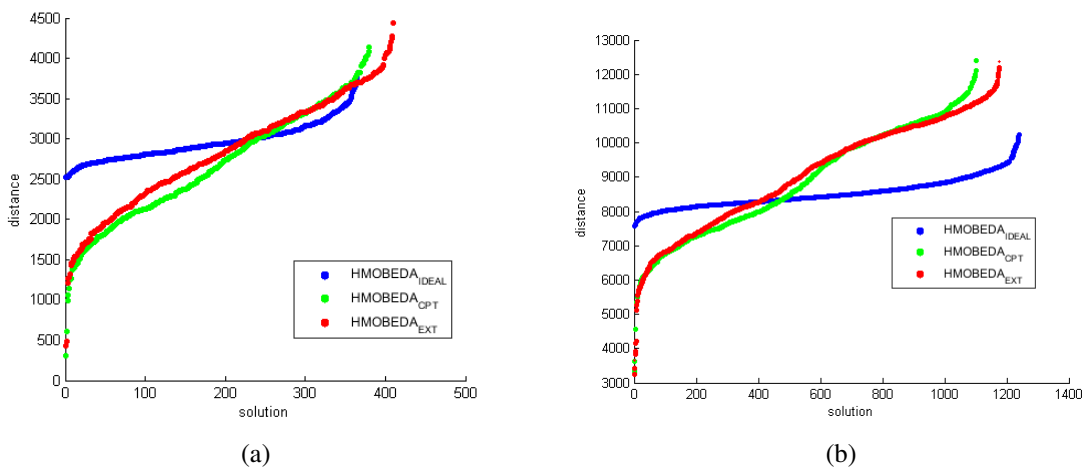


Figure 16: The Euclidean distance between each solution and the ideal point for 8 objectives with (a) 100 items and (b) 250 items.

We note, in Figures 15 and 16, that for 5 and 8 objectives, the solutions present similar probabilities since there are few large points plotted in the Pareto front. However, similarly to the 2, 3 and 4 objectives cases, HMOBEDA_{EXT} and HMOBEDA_{IDEAL} produce solutions with high diversity and small mean distances from ideal point: some solutions might be close to the ideal point but others can be far away. This behavior is also observed by HMOBEDA_{CPT} for 8-250 instance (Figure 16b).

With these experiments, we can conclude that, examining BN structures according to 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 proposed approach.

According to the results we could note that fixing evidences along the evolutionary process can guide the search through specific regions of the Pareto front, providing different convergence and diversity. HMOBEDA_{IDEAL} 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 equally likely chances of occurrence for each of them), like in HMOBEDA_{EXT}, the solutions are better distributed along the front, as depicted in Figure 10 and the remaining figures by the red points based on the fluctuation in the measured distance of each point and Z^* .

These observations based on the probability of the solutions along the Pareto front and Euclidean distance to the ideal point presented in Figures 12 to 16, corroborate the results presented in Table 15, which show that HMOBEDA_{EXT} provides a high diversity (lowest IGD values) when the number of objectives increases. Therefore we define HMOBEDA_{EXT} as our final HMOBEDA standard version (see Figure 8).

6.2.2 COMPARING HMOBEDA_{EXT} WITH CUTTING EDGE APPROACHES

In this section we compare the standard HMOBEDA version (HMOBEDA_{EXT} referred to as HMOBEDA here)⁴ with MBN-EDA (KARSHENAS et al., 2014), NSGA-II (DEB et al., 2002), S-MOGLS (ISHIBUCHI et al., 2008) (NSGA-II with local search), MOEA/D (ZHANG; LI, 2007) and NSGA-III (DEB; JAIN, 2014). In the tables, the lowest values are highlighted in bold and results with no statistically significant differences with the best values are emphasized in light blue for each instance, using Kruskal-Wallis and Dunn-Sidak's post-hoc tests with a significance level of $\alpha = 5\%$.

⁴HMOBEDA with K2 algorithm as learning structure process, crowding distance as a tie-breaker for the selection scheme, online configuration of LS parameters, and both the ideal and extremes points of the approximated Pareto front as candidates for the evidences in the root nodes during the sample process.

6.2.2.1 EXPERIMENTS WITH MOKP

Table 18 shows the hypervolume difference (HV^-) and IGD metric, both averaged over 30 executions for HMOBEDA and cutting edge evolutionary approaches.

Table 18: Average HV^- 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^-)										
	$\times 10^7$	$\times 10^8$	$\times 10^{10}$	$\times 10^{11}$	$\times 10^{14}$	$\times 10^{15}$	$\times 10^{17}$	$\times 10^{19}$	$\times 10^{29}$	$\times 10^{33}$
HMOBEDA	1.5091	1.0550	2.4284	4.1649	1.7502	4.5288	5.2559	5.4919	8.0724	1.3427
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	0.7219	2.4487	4.2432	1.8215	5.2503	6.4815	5.8908	8.1914	1.4467
Average IGD metric										
	$\times 10^3$	$\times 10^3$	$\times 1$	$\times 1$	$\times 1$	$\times 1$	$\times 1$	$\times 1$	$\times 10^3$	$\times 10^4$
HMOBEDA	2.4960	6.5290	21.7847	44.2044	18.2645	26.3493	10.2432	16.3464	8.0231	1.3235
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	2.4931	6.7602	22.5150	45.8885	18.1235	29.5950	14.1986	20.9275	8.2402	1.5197
MOEA/D	2.4992	5.0465	23.0015	45.9216	20.3025	30.2128	14.6657	21.9553	8.1993	1.5268

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

Table 19 presents the capacity metrics ONVG and ER, whose approximated front of each algorithm is calculated over 30 executions and the reference is defined as all non-dominated solutions obtained by all algorithms over all executions.

Table 19: 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
<i>Ref</i>	72	72	408	220	182	619	452	957	1133	1120
Overall Non-dominated Vector Generation (ONVG)										
HMOBEDA	65	83	417	309	663	512	965	736	1260	1239
MBN-EDA	58	53	301	246	512	495	845	652	1032	987
NSGA-II	55	41	161	71	156	143	410	405	595	399
S-MOGLS	50	65	245	102	248	246	658	598	689	659
NSGA-III	82	40	296	224	497	458	769	622	1006	1022
MOEA/D	64	96	369	298	498	498	954	640	998	985
Error Ratio (ER)										
HMOBEDA	0.7222	0.6944	0.6627	0.5927	0.5330	0.4911	0.2898	0.4619	0.1882	0.1295
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	0.6697	0.9000	0.7304	0.9000	0.6484	0.7932	0.8761	0.7868	0.9153	0.9330
MOEA/D	0.8400	0.6250	0.7451	0.6818	0.9451	0.7706	0.8783	0.8015	0.8985	0.9393

We can observe that HMOBEDA is better than the other approaches for almost all instances regarding both ONVG (largest values) and ER (lowest values), except for 2-100 and 2-250 instance. It is import to point out that almost all solutions from the reference set of instances 8-100 and 8-250 are from HMOBEDA.

Therefore, we can affirm that HMOBEDA is a competitive approach for the MOKP particularly for large size instances and can be considered a state-of-the-art algorithm for the instances considered here.

The average run time for each algorithm is presented in Table 20 only as guidance of the corresponding computational effort.

Table 20: Average Run time (minutes) 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	2.42	5.34	5.39	14.27	9.84	24.51	23.49	52.54	34.27	65.13
MBN-EDA	1.54	5.21	3.87	12.90	6.41	25.76	26.15	55.89	31.81	62.45
NSGA-II	0.87	2.21	1.57	5.87	5.89	20.13	26.51	50.32	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

We can observe from Table 20 that the computational effort required by each algorithm for the same instance are similar to each other, keeping the runtime at practical levels. However, by increasing the number of objectives and variables severely impacts the average computational time of all approaches.

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

6.2.2.2 EXPERIMENTS WITH MNK-LANDSCAPE PROBLEM

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

Tables 21 and 22 show the hypervolume difference (HV^-), IGD metric respectively, both averaged over 30 executions of each algorithm. There is no statistically significant differences between HMOBEDA, NSGA-III and MOEA/D regarding both HV^- 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^- indicator and IGD metric, where HMOBEDA presents the best values in comparison with all other approaches.

Tables 23 and 24 present the capacity metrics whose approximated front of each algorithm is calculated over 30 executions and the reference is defined as all non-dominated solutions obtained by all algorithms over all executions.

Table 21: Average HV⁻ over 30 independent executions.

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

Table 22: Average IGD over 30 executions landscapes.

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

Table 23: Overall Non-dominated Vector Generation (ONVG) over 30 executions for each algorithm. The best values are in bold.

Instance	$ Ref $	HMOBEDA	MBN-EDA	NSGA-II	S-MOGLS	NSGA-III	MOEA/D
M2N20K2	19	36	32	22	30	31	29
M2N20K4	11	28	34	13	40	34	25
M2N20K6	20	22	26	18	15	26	21
M2N20K8	10	25	39	13	6	40	16
M2N20K10	12	22	32	12	22	33	28
M2N50K2	75	80	64	16	55	64	61
M2N50K4	38	59	43	17	40	42	42
M2N50K6	42	43	55	11	30	53	31
M2N50K8	16	52	52	16	30	53	31
M2N50K10	18	39	34	15	37	34	34
M2N100K2	86	105	85	12	90	85	88
M2N100K4	55	89	71	11	84	73	80
M2N100K6	46	98	71	10	62	71	52
M2N100K8	10	63	74	12	65	75	58
M2N100K10	19	60	64	15	55	63	62
M3N20K2	161	209	168	122	122	169	135
M3N20K4	114	252	164	174	163	164	158
M3N20K6	116	260	164	111	146	164	156
M3N20K8	104	286	206	82	110	206	110
M3N20K10	126	316	200	96	99	200	99
M3N50K2	337	285	202	177	207	202	195
M3N50K4	323	273	227	99	220	226	220
M3N50K6	194	281	209	84	200	208	203
M3N50K8	156	211	245	112	198	245	197
M3N50K10	141	272	223	112	171	225	170
M3N100K2	289	341	237	92	257	238	257
M3N100K4	248	350	234	75	190	233	193
M3N100K6	248	275	210	82	244	211	251
M3N100K8	109	297	226	118	239	225	237
M3N100K10	125	295	211	107	201	211	208
M5N20K2	1114	823	491	619	428	490	434
M5N20K4	1115	954	553	581	542	555	535
M5N20K6	911	939	473	626	412	473	414
M5N20K8	888	927	501	549	400	502	401
M5N20K10	703	951	453	417	430	451	425
M5N50K2	815	1008	602	595	615	601	617
M5N50K4	849	1060	519	489	816	518	810
M5N50K6	559	1125	482	582	633	483	635
M5N50K8	943	1187	518	556	586	518	581
M5N50K10	876	1005	548	584	697	548	694
M5N100K2	842	1243	610	608	713	609	714
M5N100K4	592	1138	658	511	792	658	797
M5N100K6	539	1079	549	562	712	548	718
M5N100K8	925	1152	405	520	683	407	684
M5N100K10	1100	1059	478	524	732	478	734
M8N20K2	3096	1518	894	1376	658	895	660
M8N20K4	3611	1547	957	1438	582	958	577
M8N20K6	3525	1550	951	1434	507	951	509
M8N20K8	3445	1513	934	1412	443	934	437
M8N20K10	3233	1455	865	1230	431	865	434
M8N50K2	4077	1600	885	1402	1064	886	1059
M8N50K4	3980	1736	885	1538	842	884	845
M8N50K6	3888	1701	683	1458	773	683	772
M8N50K8	3797	1730	730	1421	721	730	716
M8N50K10	3970	1677	725	1470	595	725	597
M8N100K2	3249	1757	735	1329	1077	735	1079
M8N100K4	3728	1681	832	1459	1038	833	1027
M8N100K6	4133	1705	752	1432	975	752	970
M8N100K8	4011	1766	572	1370	859	573	861
M8N100K10	3906	1767	603	1307	859	604	855

Table 24: Error Ratio over 30 executions for each algorithm. The best values are in bold.

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

We can observe that HMOBEDA is better than the other approaches for all instances regarding both ONVG (largest values) and ER (lowest values). Another important observation is that all solutions from the reference set of M5N20K6, M5N20K10 and M5N50K2 instances have been generated using HMOBEDA.

Table 25 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, increasing the number of objectives and variables severely impacts the average computational time of all approaches, as noticed for the MOKP as well.

An analysis of the resulting BN structures has also been carried out on the HMOBEDA standard version to evaluate how the interactions among variables, objectives and local search parameters are captured by the BNs. This discussion is presented in the next section.

6.2.3 ANALYZING THE PROBABILISTIC GRAPHIC MODEL

In order to discuss the PGM structures learned during the evolutionary process we provide an interpretation of the probabilistic model resulted for some MOKP and MNK-landscape instances (examples of easy bi-objective and difficult many objective optimization instances) achieved by the HMOBEDA standard version, i.e. the one using K2 algorithm as learning structure process, crowding distance as a tie-breaker for the selection scheme, online configuration of LS parameters, and the ideal and extremes points of the approximated Pareto front as candidates for the evidences in the root nodes.

Figure 17 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 points are located nearby the +1 slope line. In other words, based on Figure 17, we can conclude that variables (specially if the number of interactions for a given variable is either very low or very high) are equally affected by both objectives.

Figure 18 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

Table 25: Average run time (minutes) for each algorithm and instance.

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

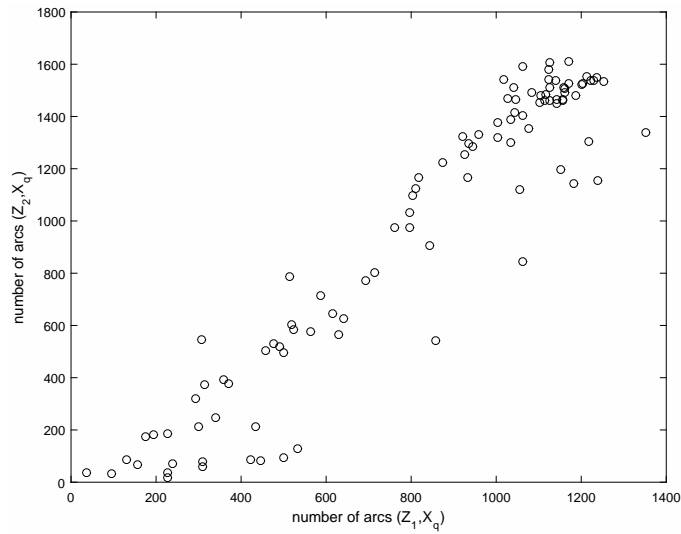


Figure 17: For instance 2-100, number of times arc (Z_1, X_q) has been captured in the BN versus a similar measure for arc (Z_2, X_q) . Each circle corresponds to one decision variable X_q , $q \in \{1, \dots, 100\}$.

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 Figure 18 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.

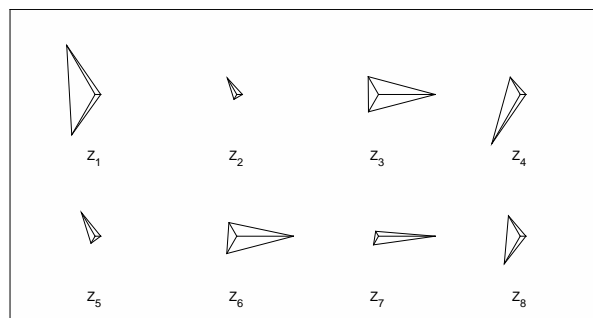


Figure 18: Glyph representation of the three LS parameters (spokes) for each objective Z_1 to Z_8 of instance 8-100.

We extend this analysis for the MNK-landscape problem. Figure 19 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 Figure 19 reveals that the variables are unequally affected by each objective.

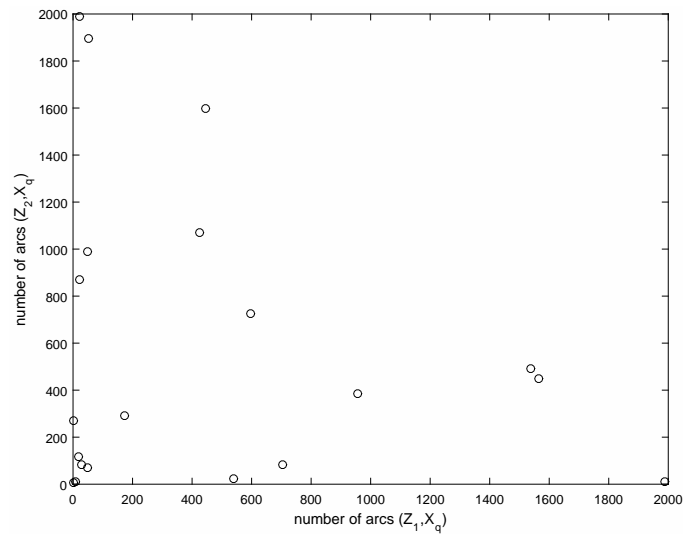


Figure 19: For M2N20K2 instance, number of times arc (Z_1, X_q) has been captured in the BN versus a similar measure for arc (Z_2, X_q) . Each circle corresponds to one decision variable X_q , $q \in \{1, \dots, 20\}$.

Figures 20 and 21 present the relations between objectives and LS parameters for M3N50K6 and M8N100K10 instances, respectively.

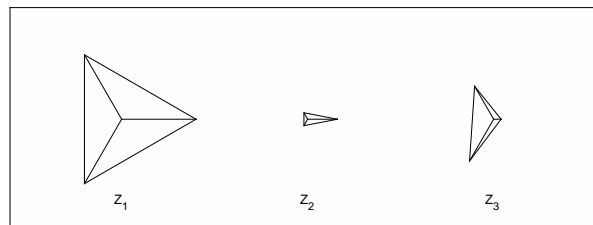


Figure 20: Glyph representation of the three LS parameters (spokes) for each objective Z_1 to Z_3 of M3N50K6 instance.

We can observe, in Figure 20, that objective Z_1 has great influence on the parameters and it is better balanced among them. From Figure 21 we can conclude that Z_4 has the least influence on the parameters.

In this analysis we have aimed to illustrate one of the main advantages of using EDA: the possibility of scrutinizing its probabilistic model which usually encompasses useful information about the relationship among variables. We have shown with these examples that HMOBEDA model 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 parameter

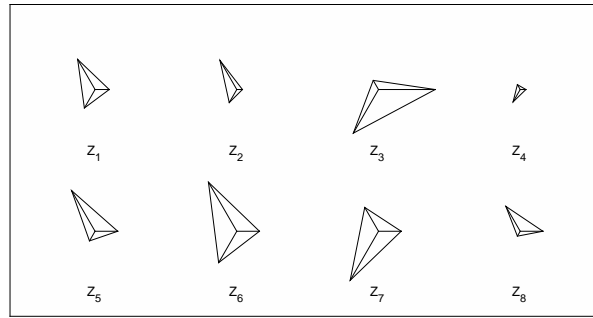


Figure 21: Glyph representation of the three LS parameters (spokes) for each objective Z_1 to Z_8 of M8N100K10 instance.

is from the analysis of objective-parameters interactions in the PGM along the evolution.

6.3 SUMMARY

This chapter begins by describing the setup of experiments. Then we have compared different versions of the proposed approach. In this comparison phase, first we investigated two BN structure learning algorithm, K2 and BN-HC, where K2 presented lower computational runtime in comparison with BN-HC, being defined as our standard BN structure learning algorithm. We also analyzed the use of CD versus hypervolume as a tie-breaker criterion, and no statistically significant difference was observed (CD presented advantage in terms of running). Then we compared HMOBEDA (which is an online LS parameter tuner) using K2 and CD with its off-line LS configuration versions. We provided these versions in order to discuss the relevance of the LS parameters encoded as BN nodes in an online configuration approach. Based on these results we have concluded that the online version of HMOBEDA is better or at least equal to its modified versions with LS off-line configuration. These findings evidence that the flexibility imposed by modeling LS parameters as nodes of the PGM model results in benefits to the hybrid model encompassing variables and objectives.

We have also compared three different versions of HMOBEDA in order to analyze how using evidences during the evolution can influence both convergence and diversity along the approximated Pareto front. We aimed to explore one of the main advantages of using EDA: the possibility of analyzing its probabilistic model which encompasses relationship among objectives, decision variables and configuration parameters of the HCLS procedure. In addition to the analysis of the final achieved PGM, this chapter evaluated the performance of HMOBEDA versions in comparison with traditional approaches considering HV^- and IGD, concluding that uniformly distributing the evidences among ideal and extreme points of the

Pareto front (HMOBEDA_{EXT}) in the sampling process is beneficial for HMOBEDA .

In all the experiments performed in this first phase we considered MOKP instances. At the end of the section we compared HMOBEDA with other techniques on the two addressed problems: MOKP and MNK-landscape, where we concluded that HMOBEDA is a competitive approach, according to hypervolume indicator, IGD and capacity metrics, for the instances considered in this work, and the average run time for each algorithm has been kept in practical levels. An analysis of the resulting BN structures was presented in order to evaluate how the interactions among variables, objectives and local search parameters are captured by the BNs. As a way 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 LS parameters are influenced by the objectives. Next chapter provides final considerations as well as a guide for future work.

7 CONCLUSION

In this work we have analyzed a new Multi-objective Estimation of Distribution Algorithm (MOEDA) based on Bayesian Network in the context of multi and many objective combinatorial optimization. The approach named Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm (HMOBEDA) considers a hill climbing procedure embedded in the probabilistic model as a local optimizer. We have incorporated this local search (LS) procedure to exploit the search space and a joint probabilistic model of objectives, variables and LS parameters to generate new individuals through the sampling process. One of the main issues investigated in this work concerns whether the auto-adaptation of LS parameters improves the search by allowing the Bayesian network to represent and set these parameters.

For the comparison of HMOBEDA versions, we initially tested two different algorithms: a greedy Hill Climber algorithm (BN-HC) as structure learning process using the K2 scoring metric, providing a version called HMOBEDA_{BN-HC}, and another one that uses the K2 algorithm named HMOBEDA_{K2}. We have also implemented two online configurations of HMOBEDA for the tie-breaker of the selection procedure: HMOBEDA_{CD} for crowding distance and HMOBEDA_{hype} for hypervolume indicator. We also built three off-line versions with no LS parameter encoded as node in the probabilistic model. The first version (HMOBEDA_f) is based on the best LS parameters found by HMOBEDA_{online}. HMOBEDA_{online} is the online version that uses K2 algorithm and CD as tie-breaker. The second version (HMOBEDA_{f-ins}) is specialized for each instance by setting the LS parameters to the best ones found by HMOBEDA_{online} for that instance. The third version (HMOBEDA_{irace}) defines the LS parameters by using irace package. Based on the experiments with MOKP instances, we conclude that K2 is better than BN-HC, as structure learning algorithm. HMOBEDA_{hype} neither improves the solution found nor presents good computational cost. The inclusion of online configuration of LS parameters is beneficial for HMOBEDA, since HMOBEDA_{online} outperformed the off-line HMOBEDA versions. In other words, the online parameter setting based on the PGM is more effective than fixing good parameters before the optimization. We have shown that the better performance of the proposed approach is more

related with the probabilistic model than with LS: as modified versions of HMOBEDA using different tuned LS parameters did not provide the same trade-off.

After that we tested different versions of HMOBEDA with ten instances of the multi-objective knapsack problem (MOKP), considering two traditional metrics: Hypervolume (HV^-) and Inverted Generational Distance (IGD). Then we compared the best version with some traditional MOEAs in the literature considering MOKP and MNK landscape problems.

In a way to explore how fixing evidences during the evolution can influence both convergence and diversity along the approximated Pareto front, we provided an interpretation of the probabilistic model resulted at the end of evolution, proposing three versions of HMOBEDA, named $HMOBEDA_{IDEAL}$, $HMOBEDA_{EXT}$ and $HMOBEDA_{CPT}$ which represent different possibilities to guide the search using the sampling process.

Finally we tested and concluded that the proposed hybrid EDA is competitive when compared with the investigated cutting edge evolutionary algorithms for both MOKP and MNK-landscape problems, as it outperformed these approaches, specially for many objective optimization. We also concluded that its PGM structure can represent the interdependencies between variables, objectives and LS parameters. The explanation is based on the fact that since LS parameters are in the same probabilistic model, HMOBEDA provides an automatic and informed decision during the evolutionary process. Thus a variety of non-dominated solutions can be found during different stages of the evolutionary process. This finding is relevant for the development of other adaptive hybrid MOEAs. Probabilistic modeling arises as a sensitive and feasible way to learn and explore dependencies not only between variables and objectives but also for controlling the application of local search operators. Another important conclusion is that it is less sensitive to an increasing number of objectives, figuring as a good candidate to many objective optimization.

7.1 FUTURE WORK

In the future, we will investigate other local search approaches, like Iterated Local Search (ILS) (LOURENÇO et al., 2003), Tabu Search (GLOVER, 1989), Simulated Annealing (AARTS; KORST, 1989) and HMOBEDA will be able to selected the best approach according to the improvement obtained along the search, like an algorithm selection framework. Therefore, we intend that HMOBEDA could learn not only the LS parameters, but also automatically select the LS procedure most adapted to the current evolutionary stage, i.e., it can operate like a hyper-heuristic. Another steps for the future include the investigation

of the MOEA techniques other than Pareto-based approaches, such as scalarizing functions, for example. These new approaches should be investigated with more than eight objectives. Beyond that, we intend to relax some of the current restrictions of our model to represent richer types of interactions (e.g., dependencies between variables). Consequently, other classes of methods to learn the BN structures, such as constraint based (ALIFERIS et al., 2010; TSAMARDINOS et al., 2003) and hybrid (TSAMARDINOS et al., 2006b) methods will be investigated. Additionally, another interesting research direction is the application of other types of PGM that can learn and explore dependencies between variables, objectives and application of local search operators. Finally, we intend to compare HMOBEDA with other MOPs and MaOPs - a multi-objective variant of the Travelling Thief Problem, for example, including a baseline method commonly used in hyperheuristic contexts which randomly generates LS parameters.

REFERENCES

- AARTS, E.; KORST, J. **Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing**. New York, NY, USA: John Wiley & Sons, Inc., 1989. ISBN 0-471-92146-7.
- AGUIRRE, H. E.; TANAKA, K. Genetic algorithms on NK-landscapes: Effects of selection, drift, mutation, and recombination. In: **Workshops on Applications of Evolutionary Computation**. Essex, UK: Springer, 2003. p. 131–142.
- AGUIRRE, H. E.; TANAKA, K. Insights on properties of multiobjective MNK-landscapes. In: **Proceedings of the 2004 Congress on Evolutionary Computation**. Portland, OR, USA: IEEE, 2004. v. 1, p. 196–203 Vol.1.
- AGUIRRE, H. E.; TANAKA, K. Working principles, behavior, and performance of MOEAs on MNK-landscapes. **European Journal of Operational Research**, Elsevier, v. 181, n. 3, p. 1670–1690, 2007.
- AHN, C. W. et al. A hybrid multiobjective evolutionary algorithm: Striking a balance with local search. **Mathematical and Computer Modelling**, v. 52, p. 2048–2059, 2010.
- ALIFERIS, C. F. et al. Local causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical evaluation. **Journal of Machine Learning Research**, v. 11, n. Jan, p. 171–234, 2010.
- ARROYO, J.; OTTONI, R. dos S.; OLIVEIRA, A. de P. Multi-objective variable neighborhood search algorithms for a single machine scheduling problem with distinct due windows. **Electronic Notes in Theoretical Computer Science**, v. 281, p. 5–19, 2011.
- ARROYO, J. E. C. **Heurísticas e Metaheurísticas para otimização combinatória multiobjetivo**. Thesis (PhD Thesis) — Universidade Estadual de Campinas, Campinas, 2002.
- BADER, J.; ZITZLER, E. Hype: An Algorithm for Fast Hypervolume-based Many-objective Optimization. **IEEE Transactions on Evolutionary Computation**, MIT Press, Cambridge, MA, USA, v. 19, n. 1, p. 45–76, mar. 2011. ISSN 1063-6560.
- BADER, J. M. **Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods**. Thesis (PhD Thesis) — ETH Zurich, Zurich, 2009.
- BALUJA, S. **Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning**. Technical Report — Carnegie Mellon University, Pittsburgh, PA, USA, 1994.
- BALUJA, S.; DAVIES, S. Using Optimal Dependency-Trees for Combinatorial Optimization: Learning the Structure of the Search Space. In: **Proceedings of the International Conference on Machine Learning**. San Francisco, California, USA: Morgan Kaufmann, 1997. (ICML'97), p. 30–38.

BAZGAN, C.; HUGOT, H.; VANDERPOOTEN, D. Solving efficiently the 0 – 1 multi-objective knapsack problem. **Computers and Operations Research**, v. 36, p. 260–279, 2009.

BENGOETXEA, E. **Inexact Graph Matching Using Estimation of Distribution Algorithms**. Thesis (PhD Thesis) — University of the Basque Country, Basque Country, 2002.

BENGOETXEA, E. et al. Optimal row and column ordering to improve table interpretation using estimation of distribution algorithms. **Journal of Heuristics**, Springer, v. 17, n. 5, p. 567–588, 2011.

BEUME, N.; NAUJOKS, B.; EMMERICH, M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. **European Journal of Operational Research**, v. 181, n. 2, p. 1653–1669, 2007.

BIRATTARI, M. et al. F-Race and Iterated F-race: An Overview. In: **Experimental Methods for the Analysis of Optimization Algorithms**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 311–336.

BONET, J. S. D. et al. Mimic: Finding optima by estimating probability densities. **Advances in neural information processing systems**, Morgan Kaufmann Publishers, p. 424–430, 1997.

BOUHAMED, H. et al. 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**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. (ICIC'12), p. 387–393. ISBN 978-3-642-31837-5.

CARTA, J. A.; VELÁZQUEZ, S.; MATÍAS, J. Use of Bayesian networks classifiers for long-term mean wind turbine energy output estimation at a potential wind energy conversion site. **Energy Conversion and Management**, Elsevier, v. 52, n. 2, p. 1137–1149, 2011.

CASELLA, G.; BERGER, R. L. **Statistical Inference**. Second. Pacific Grove, CA: Duxbury, 2002.

CHABANE, B.; BASSEUR, M.; HAO, J.-K. A practical case of the multiobjective knapsack problem: Design, modelling, tests and analysis. In: **International Conference on Learning and Intelligent Optimization**. Lille, France: Springer, 2015. p. 249–255.

CHABANE, B.; BASSEUR, M.; HAO, J.-K. R2-IBMOLS applied to a practical case of the multiobjective knapsack problem. **Expert Systems with Applications**, Elsevier, v. 71, p. 457–468, April 2017.

COELLO, C. A. **An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design**. Thesis (PhD Thesis) — Tulane University, New Orleans, USA, 1996.

COELLO, C. A. C. An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends. In: **IEEE Congress on Evolutionary Computation**. Washington, D.C., USA: IEEE, 1999. (CEC'99), p. 3–13.

COELLO, C. C.; LAMONT, G.; VELDHUIZEN, D. v. **Evolutionary Algorithms for Solving Multi-Objective Problems**. Second. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

CONOVER, W. **Practical Nonparametric Statistics**. Third. New York: Wiley, 1999.

- COOPER, G.; HERSKOVITS, E. A Bayesian method for the induction of probabilistic networks from data. **Machine Learning**, v. 9, n. 4, p. 309–347, 1992.
- CORNE, D.; KNOWLES, J. D.; OATES, M. J. The Pareto Envelope-Based Selection Algorithm for Multi-objective Optimisation. In: **Proceedings of the 6th International Conference on Parallel Problem Solving from Nature**. London, UK: Springer-Verlag, 2000. (PPSN VI), p. 839–848. ISBN 3-540-41056-2.
- CORNE, D. W. et al. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In: **Proceedings of the Genetic and Evolutionary Computation Conference**. Burlington, Massachusetts: Morgan Kaufmann Publishers, 2001. (GECCO'2001), p. 283–290.
- CROCOMO, M. K.; DELBEM, A. C. B. **Otimização por Decomposição**. Technical Report, Universidade de São Paulo, 2011.
- CZYZZŹAK, P.; JASZKIEWICZ, A. Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. **Journal of Multi-Criteria Decision Analysis**, John Wiley & Sons, Ltd., v. 7, n. 1, p. 34–47, 1998.
- DAOLIO, F. et al. 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**. [S.l.]: ACM, 2015. (GECCO'15), p. 369–376.
- DAS, I.; DENNIS, J. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. **SIAM J. Optimization**, v. 8, n. 3, p. 631–657, 1998.
- DEB, K. **Multi-objective optimization using evolutionary algorithms**. New York: John Wiley and Sons, 2001.
- DEB, K. Multi-objective optimisation using evolutionary algorithms: An introduction. In: WANG, L.; NG, A. H. C.; DEB, K. (Ed.). **Multi-objective Evolutionary Optimisation for Product Design and Manufacturing**. London: Springer London, 2011. p. 3–34.
- DEB, K. et al. A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, v. 6, p. 182–197, 2002.
- DEB, K.; JAIN, H. 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**, v. 18, n. 4, p. 577–601, 2014.
- DEGROOT, M. H. **Optimal statistical decisions**. Hoboken, New Jersey: John Wiley & Sons, 2005.
- EL YAFRANI, M. et al. A hyperheuristic approach based on low-level heuristics for the travelling thief problem. **Genetic Programming and Evolvable Machines**, p. 1–30, Jul 2017.
- ETXEBERRIA, R.; LARRAÑAGA, P. Global optimization using Bayesian networks. In: **Proceedings of the Second Symposium on Artificial Intelligence**. Havana, Cuba: Editorial Academia, 1999. (CIMAF'99), p. 332–339.

- FOURMAN, M. P. Compaction of symbolic layout using genetic algorithms. In: **Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications**. Pittsburgh, PA: [s.n.], 1985. v. 4, p. 141–153.
- GAAG, L. van der; WAAL, P. de. Multi-dimensional Bayesian network classifiers. In: **Proceedings of the Third European Workshop in Probabilistic Graphical Models**. Prague, Czech Republic: Action M Agency, 2006. p. 107–114.
- GANDIBLEUX, X.; FRÁVILLE, A. Tabu search based procedure for solving the 0-1 multiobjective knapsack problem: The two objectives case. **Journal of Heuristics**, v. 6, p. 361–383, 2000.
- GARCÍA, S. et al. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. **Journal of Heuristics**, v. 15, n. 6, p. 617, 2008.
- GLOVER, F. Tabu search-part I. **ORSA Journal on Computing**, INFORMS, v. 1, n. 3, p. 190–206, 1989.
- GUPTA, S.; BATRA, D.; VERMA, P. Greedy Estimation of Distributed Algorithm to Solve Bounded knapsack Problem. **International Journal of Computer Science and Information Technologies**, v. 5, p. 4313–4316, 2014.
- HAJELA, P.; LIN, C. Y. Genetic Search Strategies in Multicriterion Optimal Design. **Structural Optimization**, Springer, New York, v. 4, p. 99–107, 1992.
- HARIK, G. **Linkage learning via probabilistic modeling in the ECGA**. 801 p. Technical Report — University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory - Tech Report no.99010, Urbana, IL, 1999.
- HARIK, G. R.; LOBO, F. G.; GOLDBERG, D. E. The compact genetic algorithm. **IEEE transactions on Evolutionary Computation**, IEEE, v. 3, n. 4, p. 287–297, 1999.
- HECKERMAN, D.; GEIGER, D.; CHICKERING, D. Learning Bayesian networks: the combination of knowledge and statistical data. **Machine Learning**, v. 20, n. 3, p. 197–243, 1995.
- HENRION, M. Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling. In: **2 Annual Conference on Uncertainty in Artificial Intelligence (UAI-86)**. Amsterdam, NL: Elsevier Science, 1986. p. 149–163.
- HOCHBERG, Y.; TAMHANE, A. C. **Multiple Comparison Procedures**. Hoboken, NJ: John Wiley and Sons, 1987.
- HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. USA: MIT press, 1992.
- HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. E. A Niche Pareto Genetic Algorithm for Multiobjective Optimization. In: **IEEE Conference on Evolutionary Computation**. Piscataway, New Jersey: IEEE, 1994. (CEC'1994, v. 1), p. 82–87.
- HRUSCHKA, E. R.; EBECKEN, N. F. Towards efficient variables ordering for Bayesian networks classifier. **Data & Knowledge Engineering**, Elsevier, v. 63, n. 2, p. 258–269, 2007.

- ISHIBUCHI, H.; AKEDO, N.; NOJIMA, Y. Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems. **IEEE Transactions on Evolutionary Computation**, v. 19, n. 2, p. 264–283, 2015.
- ISHIBUCHI, H.; HITOTSUYANAGI, Y.; NOJIMA, Y. Scalability of Multiobjective Genetic Local Search to Many-Objective Problems:Knapsack Problem Case Studies. In: **IEEE Conference on Evolutionary Computation**. Piscataway, New Jersey: IEEE, 2008. (CEC'2008), p. 3586–3593.
- ISHIBUCHI, H.; MURATA, T. A multi-objective genetic local search algorithm and its application to flowshop scheduling. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 28, n. 3, p. 392–403, 1998.
- ISHIBUCHI, H.; TSUKAMOTO, N.; NOJIMA, Y. Evolutionary many-objective optimization: A short review. In: **IEEE Conference on Evolutionary Computation**. Piscataway, New Jersey: IEEE, 2008. (CEC'2008), p. 2419–2426.
- JASZKIEWICZ, A. Comparison of local search-based metaheuristics on the multiple-objective knapsack problem. **Foundations of Computing and Decision Sciences**, v. 26, p. 99–120, 2001.
- JASZKIEWICZ, A. Genetic local search for multiple objective combinatorial optimization. **European Journal of Operational Research**, v. 137, p. 50–71, 2002.
- JASZKIEWICZ, A. On the performance of multiple objective genetic local search on the 0/1 knapsack problem: a comparative experiment. **IEEE Transactions on Evolutionary Computation**, v. 6, p. 402–412, 2002.
- JENKINS, L. A bicriteria knapsack program for planning remediation of contaminated lightstation sites. **European Journal of Operational Research**, v. 140, n. 2, p. 427–433, 2002.
- JIANG, S. et al. Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization. **IEEE Transactions on Cybernetics**, v. 44, n. 12, p. 2391–2404, 2014.
- JIANG, S. et al. A Simple and Fast Hypervolume Indicator-based Multiobjective Evolutionary Algorithm. **IEEE Transactions on Cybernetics**, n. 10, p. 2202–2213, 2015.
- KARSHENAS, H. et al. Multiobjective Estimation of Distribution Algorithm Based on Joint Modeling of Objectives and Variables. **IEEE Transactions on Evolutionary Computation**, v. 18, p. 519–542, 2014.
- KAUFFMAN, S.; LEVIN, S. Towards a general theory of adaptive walks on rugged landscapes. **Journal of theoretical Biology**, Elsevier, v. 128, n. 1, p. 11–45, 1987.
- KAUFFMAN, S. A. **The origins of order: Self-organization and selection in evolution**. USA: Oxford University Press, 1993.
- KAUFFMAN, S. A.; WEINBERGER, E. D. The NK model of rugged fitness landscapes and its application to maturation of the immune response. **Journal of theoretical biology**, Elsevier, v. 141, n. 2, p. 211–245, 1989.
- KE, L.; ZHANG, Q.; BATTITI, R. A Simple yet Efficient Multiobjective Combinatorial Optimization Method Using Decomposition and Pareto Local Search. **IEEE Transactions on Cybernetics**, v. 44, p. 1808–1820, 2014.

KHAN, N.; GOLDBERG, D. E.; PELIKAN, M.

Multi-objective Bayesian optimization algorithm — University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory - Tech Report no.2002009, 2002.

KNOWLES, J. et al. **A tutorial on the performance assessment of stochastic multiobjective optimizers**. Technical Report — Computer Engineering and Networks Laboratory (TIK), ETH Zurich, TIK Report 214, 2006.

KNOWLES, J. D.; CORNE, D. W. A comparison of diverse approaches to memetic multiobjective combinatorial optimization. In: **Genetic and Evolutionary Computation Conference**. Burlington, Massachusetts: Morgan Kaufmann Publishers, 2000. (GECCO'00), p. 103–108.

KOLLAT, J. B.; REED, P.; KASPRZYK, J. A new epsilon-dominance hierarchical Bayesian optimization algorithm for large multiobjective monitoring network design problems. **Advances in Water Resources**, Elsevier, v. 31, n. 5, p. 828–845, 2008.

KOLLER, D.; FRIEDMAN, N. **Probabilistic Graphical Models: Principles and Techniques**. Cambridge, MA: The MIT Press, 2009.

KORB, K. B.; NICHOLSON, A. E. **Bayesian Artificial Intelligence**. Second edition. Boca Raton, Florida: CRC Press, 2010.

KOSTREVA, M.; OGRYCZAK, W.; TONKYN, D. Relocation problems arising in conservation biology. **Computers and Mathematics with Applications**, v. 37, p. 135–150, 1999.

KRASNOGOR, N.; GUSTAFSON, S. Toward truly "memetic" memetic algorithms: discussion and proof of concepts. In: **Advances in Nature-Inspired Computation: the PPSN VII Workshops**. London, UK, UK: Springer-Verlag, 2002. (PPSN VII).

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion analysis of variance. **Journal of the American Statistical Association**, v. 47, p. 583–621, 1952.

LARA, A. et al. HCS: A new local search strategy for memetic multiobjective evolutionary algorithms. **IEEE Transactions on Evolutionary Computation**, v. 14, n. 1, p. 112–132, Feb 2010.

LARRAÑAGA, P. et al. A review on probabilistic graphical models in evolutionary computation. **Journal of Heuristics**, v. 18, p. 795–819, 2012.

LARRAÑAGA, P.; LOZANO, J. A. **Estimation of distribution algorithms: A new tool for evolutionary computation**. Netherlands: Springer, 2002.

LAUMANNNS, M.; OCENASEK, J. Bayesian Optimization Algorithms for Multi-Objective Optimization. In: **Parallel Problem Solving from Nature**. London, UK, UK: Springer-Verlag, 2002. (PPSN VII - Lecture Notes in Computer Science, v. 2439), p. 298–307.

LERNER, B.; MALKA, R. Investigation of the K2 Algorithm in Learning Bayesian Network Classifiers. **Applied Artificial Intelligence**, p. 74–96, 2011.

LEVENE, H. Contributions to probability and statistics. **Essays in honor of Harold Hotelling**, Stanford University Press, p. 278–292, 1960.

LI, H.; ZHANG, Q. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. **IEEE Transactions on Evolutionary Computation**, v. 13, n. 2, p. 284–302, 2009.

LI, H. et al. Hybrid Estimation of Distribution Algorithm for Multiobjective Knapsack Problem. **Evolutionary Computation in Combinatorial Optimization**, Springer Berlin Heidelberg, p. 145–154, 2004.

LI, M. et al. IPESA-II: Improved Pareto Envelope-Based Selection Algorithm II. In: **Proceedings of 7th International Conference on Evolutionary Multi-Criterion Optimization**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. (EMO'13), p. 143–155.

LIAW, R.-T.; TING, C.-K. Effect of model complexity for estimation of distribution algorithm in NK landscapes. In: **Symposium on Foundations of Computational Intelligence**. [S.l.]: IEEE, 2013. (FOCI'13), p. 76–83.

LIEFOOGHE, A. et al. A feature-based performance analysis in evolutionary multiobjective optimization. In: **International Conference on Evolutionary Multi-Criterion Optimization**. Guimarães, Portugal: Springer, 2015. p. 95–109.

LÓPEZ-IBÁÑEZ, M. et al.

The irace Package: Iterated Racing for Automatic Algorithm Configuration — Université Libre de Bruxelles - Tech Report no. 2011-004, 2011.

LÓPEZ-IBÁÑEZ, M.; STÜTZLE, T. The Automatic Design of Multiobjective Ant Colony Optimization Algorithms. **IEEE Transactions on Evolutionary Computation**, v. 16, n. 6, p. 861–875, Dec 2012. ISSN 1089-778X.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: GENDREAU MICHEL, P. J.-Y. (Ed.). **Handbook of metaheuristics**. USA: Springer, 2003. p. 320–353.

LUNA, J. E. O. **Algoritmos EM para Aprendizagem de Redes Bayesianas a partir de Dados Incompletos**. Thesis (Master Thesis) — Universidade Federal do Mato Grosso do Sul, Campo Grande, 2004.

LUQUE, M. Modified Interactive Chebyshev Algorithm (MICA) for non-convex multiobjective programming. **Optimization Letters**, v. 9, n. 1, p. 173–187, 2015.

LUST, T.; TEGHEM, J. The multiobjective multidimensional knapsack problem: a survey and a new approach. **International Transactions in Operational Research**, v. 19, p. 495–520, 2012.

MALKA, R.; LERNER, B. Classification of fluorescence in situ hybridization images using belief networks. **Pattern Recognition Letters**, Elsevier, v. 25, n. 16, p. 1777–1785, 2004.

MARTÍ-OROSA, L. **Scalable Multi-Objective Optimization**. Thesis (PhD Thesis) — Universidad Carlos III de Madrid, Colmenarejo, Spain, 2011.

- MARTINS, M. S. et al. HMOBEDA: Hybrid Multi-objective Bayesian Estimation of Distribution Algorithm. In: **Proceedings of the Genetic and Evolutionary Computation Conference**. New York, NY, USA: ACM, 2016. (GECCO'16), p. 357–364. ISBN 978-1-4503-4206-3.
- MARTINS, M. S. R. et al. Hybrid multi-objective Bayesian estimation of distribution algorithm: a comparative analysis for the multi-objective knapsack problem. **Journal of Heuristics**, Sep 2017.
- MARTINS, M. S. R. et al. HSEDA: A heuristic selection approach based on estimation of distribution algorithm for the travelling thief problem. In: **Proceedings of the Genetic and Evolutionary Computation Conference**. New York, NY, USA: ACM, 2017. (GECCO'17), p. 361–368.
- MESSAC, A.; ISMAIL-YAHAYA, A.; MATTSON, C. A. The normalized normal constraint method for generating the Pareto frontier. **Structural and Multidisciplinary Optimization**, Springer, v. 25, n. 2, p. 86–98, 2003.
- MIETTINEN, K. **Nonlinear Multiobjective Optimization**. Norwell, MA: Kluwer, 1999.
- MORAN, S.; HE, Y.; LIU, K. Choosing the best Bayesian classifier: An empirical study. **IAENG International Journal of Computer Science**, v. 36, n. 4, p. 322–331, 2009.
- MÜHLENBEIN, H.; MAHNIG, T. Convergence theory and applications of the factorized distribution algorithm. **Journal of Computing and Information Theory**, v. 7, n. 1, p. 19–32, 1999.
- MÜHLENBEIN, H.; PAAB, G. From Recombination of Genes to the Estimation of Distributions I. Binary parameters. In: **Parallel Problem Solving from Nature**. London, UK, UK: Springer-Verlag, 1996. (PPSN IV - Lecture Notes in Computer Science 1411), p. 178–187.
- PEARL, J. **Probabilistic reasoning in intelligent systems: Networks of plausible inference**. San Mateo, CA: Morgan Kaufmann, 1988.
- PEARL, J. **Causality: Models, Reasoning and Inference**. USA: Cambridge University Press, 2000.
- PELIKAN, M. Analysis of estimation of distribution algorithms and genetic algorithms on NK landscapes. In: **Proceedings of the 10th annual conference on Genetic and Evolutionary Computation**. Atlanta, Georgia: ACM, 2008. (GECCO'08), p. 1033–1040.
- PELIKAN, M.; GOLDBERG, D. E.; CANTÚ-PAZ, E. BOA: The Bayesian optimization algorithm. In: **Proceedings of the Genetic and Evolutionary Computation Conference**. San Francisco, CA: Morgan Kaufmann Publishers, 1999. (GECCO'99, I), p. 525–532.
- PELIKAN, M.; GOLDBERG, D. E.; TSUTSUI, S. Hierarchical Bayesian optimization algorithm: toward a new generation of evolutionary algorithms. In: **Society of Instrument and Control Engineers Annual Conference**. Piscataway, New Jersey: IEEE, 2003. (SICE'03, v. 3), p. 2738–2743.
- PELIKAN, M.; MUEHLENBEIN, H. The Bivariate Marginal Distribution Algorithm. In: **Advances in Soft Computing: Engineering Design and Manufacturing**. London: Springer London, 1999. p. 521–535. ISBN 978-1-4471-0819-1.

- PELIKAN, M. et al. 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**. Montréal, Canada: ACM, 2009. (GECCO'09), p. 851–858.
- PERRIER, E. et al. Finding Optimal Bayesian Network Given a Super Structure. **Journal of Machine Learning Research**, v. 9, p. 2251–2286, 2008.
- PHAM, N. **Investigations of constructive approaches for examination timetabling and 3D-strip packing**. Thesis (PhD Thesis) — School of Computer Science and Information Technology, University of Nottingham, UK, 2011.
- PORWAL, A.; CARRANZA, E. J. M.; HALE, M. Bayesian network classifiers for mineral potential mapping. **Computers & Geosciences**, Elsevier, v. 32, n. 1, p. 1–16, 2006.
- ROSENBLATT, M.; SINUANY-STERN, Z. Generating the discrete efficient frontier to the capital budgeting problem. **Operations Research**, v. 37, p. 384–394, 1989.
- RUSSEL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Second. Upper Saddle River, New Jersey, USA: Prentice Hall, 2003. ISBN 0-13-790395-2.
- SAMIR, B.; YACINE, L.; MOHAMED, B. Local Search Heuristic for Multiple Knapsack Problem. **International Journal of Intelligent Information Systems**, v. 4, n. 2, p. 35–39, 2015.
- SANTANA, R.; LARRAÑAGA, P.; LOZANO, J. A. Combining variable neighborhood search and estimation of distribution algorithms in the protein side chain placement problem. **Journal of Heuristics**, v. 14, p. 519–547, 2008.
- SANTANA, R.; LARRAÑAGA, P.; LOZANO, J. A. Learning factorizations in estimation of distribution algorithms using affinity propagation. **Evolutionary Computation**, v. 18, n. 4, dec. 2010.
- SANTANA, R.; MENDIBURU, A.; LOZANO, J. A. Evolving MNK-landscapes with structural constraints. In: **IEEE Congress on Evolutionary Computation**. Sendai, Japan: IEEE, 2015. (CEC'15), p. 1364–1371.
- SANTANA, R.; MENDIBURU, A.; LOZANO, J. A. Multi-objective NM-landscapes. In: **Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation**. Orlando, Florida, USA: ACM, 2015. (GECCO'15), p. 1477–1478.
- SCHWARZ, J.; OCENASEK, J. Multiobjective Bayesian Optimization Algorithm for Combinatorial Problems: Theory and Practice. **Neural Network World**, v. 11, n. 5, p. 423–441, 2001. ISSN 1210-0552.
- SCHWARZ, J.; OCENASEK, J. Pareto Bayesian Optimization Algorithm for the Multiobjective 0/1 Knapsack problem. In: **7th International Mendel Conference on Soft Computing**. Piscataway, New Jersey: IEEE, 2001. (MENDEL'2001), p. 131–136.
- SEADA, H.; DEB, K. U-NSGA-III: A unified evolutionary algorithm for single, multiple, and many-objective optimization. **COIN report**, n. 2014022, 2015.

SESHADRI, A. **Multi-Objective Optimization using Evolutionary Algorithm**. [Http://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm](http://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm). Access date: 2 feb.2016.

SHAH, R.; REED, P. Comparative analysis of multiobjective evolutionary algorithms for random and correlated instances of multiobjective d-dimensional knapsack problems. **European Journal of Operational Research**, Elsevier, v. 211, n. 3, p. 466–479, 2011.

SHAKYA, S.; SANTANA, R. An EDA Based on Local Markov Property and Gibbs Sampling. In: **Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2008. (GECCO'08), p. 475–476. ISBN 978-1-60558-130-9.

SHIM, V. A. et al. Multi-objective optimization with estimation of distribution algorithm in a noisy environment. **Evolutionary computation**, MITP, v. 21, n. 1, p. 149–177, 2013.

SPIRITES, P.; GLYMOUR, C. N.; SCHEINES, R. **Causation, prediction, and search**. USA: MIT press, 2000.

SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary Computation**, v. 2, p. 221–248, 1994.

TAN, Y.-Y.; JIAO, Y.-C. MOEA/D with Uniform Design for Solving Multiobjective Knapsack Problems. **Journal of Computers**, v. 8, p. 302–307, 2013.

TANIGAKI, Y. et al. 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)**. Kita-Kyushu, Japan: IEEE, 2014. p. 637–642.

TENG, J.; TZENG, G. A multiobjective programming approach for selecting non-independent transportation investment alternatives. **Transportation Research-B**, v. 30, p. 201–307, 1996.

TINÓS, R.; YANG, S. Analysis of fitness landscape modifications in evolutionary dynamic optimization. **Information Sciences**, Elsevier, v. 282, p. 214–236, 2014.

TRIVEDI, A. et al. A survey of multiobjective evolutionary algorithms based on decomposition. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 21, n. 3, p. 440–462, 2017.

TSAMARDINOS, I. et al. Algorithms for Large Scale Markov Blanket Discovery. In: **FLAIRS conference**. St. Augustine, Florida, USA: AAAI Press, 2003. v. 2, p. 376–380.

TSAMARDINOS, I.; BROWN, L. E.; ALIFERIS, C. F. The Max-min Hill-climbing Bayesian Network Structure Learning Algorithm. **Machine Learning**, Kluwer Academic Publishers, Hingham, MA, USA, v. 65, n. 1, p. 31–78, 2006.

TSAMARDINOS, I.; BROWN, L. E.; ALIFERIS, C. F. The max-min hill-climbing Bayesian network structure learning algorithm. **Machine learning**, Springer, v. 65, n. 1, p. 31–78, 2006.

VALENTE, M. An NK-like model for complexity. **Journal of Evolutionary Economics**, Springer, v. 24, n. 1, p. 107–134, 2014.

- VELDHUIZEN, D. A. D. A. van; LAMONT, G. B. Multiobjective Evolutionary Algorithm Test Suites. In: **Proceedings of the 1999 ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 1999. (SAC'99), p. 351–357.
- VELDHUIZEN, D. A. V.; LAMONT, G. B. On measuring multiobjective evolutionary algorithm performance. In: **Proceedings of the Congress on Evolutionary Computation**. La Jolla, California, USA: IEEE, 2000. v. 1, p. 204–211.
- VEREL, S. et al. Pareto local optima of multiobjective NK-landscapes with correlated objectives. **Evolutionary computation in combinatorial optimization**, Springer, p. 226–237, 2011.
- VEREL, S. et al. On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives. **European Journal of Operational Research**, Elsevier, v. 227, n. 2, p. 331–342, 2013.
- VEREL, S.; OCHOA, G.; TOMASSINI, M. Local optima networks of NK landscapes with neutrality. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 15, n. 6, p. 783–797, 2011.
- VIANNA, D. S.; VIANNA, M. de F. D. Local search-based heuristics for the multiobjective multidimensional knapsack problem. **Produção**, v. 23, p. 478–487, 2013.
- WAAL, P. R. de; GAAG, L. C. van der. 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**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. (ECSQARU'07), p. 501–511. ISBN 978-3-540-75256-1.
- WAGNER, T.; BEUME, N.; NAUJOKS, B. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. **Evolutionary Multi-Criterion Optimization**, Springer, Berlin, p. 742–763, March 2007.
- WANG, L.; WANG, S.; XU, Y. An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem. **Expert Systems with Applications**, v. 39, p. 5593–5599, 2012.
- WITTEN, I. H. et al. **Data Mining: Practical machine learning tools and techniques**. Fourth edition. [S.l.]: Morgan Kaufmann, 2016.
- YAN, J. et al. Diversity metrics in multi-objective optimization: Review and perspective. In: **IEEE International Conference on Integration Technology**. Shenzhen, China: [s.n.], 2007. (ICIT'07), p. 553–557.
- YEN, G. G.; HE, Z. Performance Metric Ensemble for Multiobjective Evolutionary Algorithms. **IEEE Transactions on Evolutionary Computation**, v. 18, n. 1, p. 131–144, Feb 2014.
- YUAN, C.; MALONE, B. Learning Optimal Bayesian Networks: A Shortest Path Perspective. **Journal of Artificial Intelligence Research**, v. 48, n. 1, p. 23–65, 2013. ISSN 1076-9757.
- ZHANG, Q.; LI, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. **IEEE Transactions on Evolutionary Computation**, v. 11, n. 6, p. 712–731, 2007.

- ZHOU, A. et al. Multiobjective evolutionary algorithms: A survey of the state of the art. **Swarm and Evolutionary Computation**, n. 1, p. 32–49, 2011.
- ZHOU, A.; SUN, J.; ZHANG, Q. An Estimation of Distribution Algorithm with Cheap and Expensive Local Search Methods. **IEEE Transactions on Evolutionary Computation**, v. 19, n. 6, p. 807–822, Dec 2015.
- ZITZLER, E.; KNOWLES, J.; THIELE, L. Quality assessment of Pareto set approximations. **Multiobjective Optimization, ser. Lecture Notes in Computer Science**, Springer, Verlag, Berlin, Heidelberg, v. 5252, p. 373–404, 2008.
- ZITZLER, E.; LAUMANN, M.; BLEULER, S. A Tutorial on Evolutionary Multiobjective Optimization. **Metaheuristics for Multiobjective Optimisation**, Berlin, v. 535, p. 3–37, 2004.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. **SPEA2: Improving the Strength Pareto Evolutionary Algorithm**. Technical Report — Computer Engineering and Networks Laboratory (TIK), ETH Zurich, TIK-Report 103, Zurich, Switzerland, 2001.
- ZITZLER, E.; THIELE, L. Multiple objective evolutionary algorithms: A comparative case study and the strength Pareto approach. **IEEE Transactions on Evolutionary Computation**, v. 3, p. 257–271, 1999.
- ZITZLER, E.; THIELE, L.; DEB, K. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. **Evolutionary Computation**, v. 8, n. 2, p. 173–195, 2000.
- ZITZLER, E. et al. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. **IEEE Transactions on Evolutionary Computation**, v. 7, p. 117–132, 2003.

APPENDIX A – EXAMPLE OF THE K2 ALGORITHM

An example of the K2 algorithm to learn the topology of a BN is presented as follows.

Considering the dataset given by Table 26, where the variables have two possible values $\{0, 1\}$.

Table 26: A given dataset for the K2 algorithm example

	Y_1	Y_2	Y_3
case 1	0	1	0
case 2	1	0	1
case 3	1	1	1
case 4	0	0	1
case 5	1	1	0
case 6	1	1	1
case 7	0	0	0
case 8	0	1	0
case 9	1	0	1
case 10	0	0	0

The inputs for the K2 algorithm are:

- The set of M nodes: $\{Y_1, Y_2, Y_3\}$;
- A pre-defined order: **order**=(Y_1, Y_2, Y_3);
- The upper bound on the number of parents a node may have: $u = 2$;
- The dataset Pop : $N = 10$ cases.

iteration $e = 1$:

$$m = order_1 : m = 1;$$

$$\mathbf{Pa}_1 = \emptyset;$$

$$Score_{old} = f(1, \emptyset);$$

$$s_1 = 2;$$

Since $\mathbf{Pa}_1 = \emptyset$, Step 3, $t_1 = 0$ and Y_1 has no parents, j will be ignored in the $Score_{old}$ formula (COOPER; HERSKOVITS, 1992). Then, according to the dataset Pop presented in Table 26:

$$N_{1j1} = 5, \text{ cases where } Y_1 = 0 \text{ (1,4,7,8 and 10);}$$

$$N_{1j2} = 5, \text{ cases where } Y_1 = 1 \text{ (2,3,5,6 and 9).}$$

$$N_{1j} = 10.$$

Hence:

$$f(1, \mathbf{Pa}_1) = \prod_{j=1}^{t_1} \frac{(s_1-1)!}{(N_{1j}+s_1-1)!} \prod_{k=1}^{s_1} (N_{1jk})!$$

$$f(1, \emptyset) = \frac{(2-1)!}{(10+2-1)!} \prod_{k=1}^2 (N_{1jk})! = \frac{1!}{11!} * 5! * 5! = 1/2772$$

Since $Pred(Y_1) = \emptyset$, the iteration for $e = 1$ ends with $\mathbf{Pa}_1 = \emptyset$.

iteration $e = 2$:

$$m = order_2 : m = 2;$$

$$\mathbf{Pa}_2 = \emptyset;$$

$$Score_{old} = f(2, \emptyset);$$

$$s_2 = 2;$$

Considering the same analysis used for iteration $e = 1$, for $m = 2$ we have:

$$N_{2j1} = 5, \text{ cases where } Y_2 = 0 \text{ (2,4,7,9 and 10);}$$

$$N_{2j2} = 5, \text{ cases where } Y_2 = 1 \text{ (1,3,5,6 and 8);}$$

$$N_{2j} = 10.$$

Hence:

$$f(2, \mathbf{Pa}_2) = \prod_{j=1}^{t_2} \frac{(s_2-1)!}{(N_{2j}+s_2-1)!} \prod_{k=1}^{s_2} (N_{2jk})!$$

$$f(2, \emptyset) = \frac{(2-1)!}{(10+2-1)!} \prod_{k=1}^2 (N_{2jk})! = \frac{1!}{11!} * 5! * 5! = 1/2772$$

$$Score_{old} = 1/2772;$$

However $Pred(Y_2) = \{Y_1\}$, then, in Step 8, we have to calculate $f(2, \mathbf{Pa}_2 \cup \{i\})$ for $i = Y_1$:

$$f(2, \mathbf{Pa}_2 \cup \{Y_1\}) = f(2, \{Y_1\})$$

$$f(2, \{Y_1\}) = \prod_{j=1}^{t_2} \frac{(s_2-1)!}{(N_{2j}+s_2-1)!} \prod_{k=1}^{s_2} (N_{2jk})!$$

Since $t_2 = 2$ ($Y_1 = 0$ or $Y_1 = 1$), then:

$N_{211} = 3$, cases where $Y_1 = 0$ and $Y_2 = 0$ (4,7 and 10);

$N_{212} = 2$, cases where $Y_1 = 0$ and $Y_2 = 1$ (1 and 8);

$N_{221} = 2$, cases where $Y_1 = 1$ and $Y_2 = 0$ (2 and 9);

$N_{222} = 3$, cases where $Y_1 = 1$ and $Y_2 = 1$ (3,5 and 6);

$N_{21} = 5$;

$N_{22} = 5$.

$$f(2, \{Y_1\}) = \frac{(2-1)!}{(5+2-1)!} * \prod_{k=1}^2 (N_{21k})! * \frac{(2-1)!}{(5+2-1)!} \prod_{k=1}^2 (N_{22k})!$$

$$f(2, \{Y_1\}) = \frac{(1)!}{(6)!} * (N_{211})! * (N_{212})! * \frac{(1)!}{(6)!} * (N_{221})! * (N_{222})!$$

$$f(2, \{Y_1\}) = \frac{(1)!}{(6)!} * 3! * 2! * \frac{(1)!}{(6)!} * 2! * 3!$$

$$f(2, \{Y_1\}) = \frac{1}{6*5*4} * 2 * \frac{1}{6*5*4} * 2 = 1/3600$$

$$Score_{new} = 1/3600$$

Since $Score_{new} = 1/3600 < Score_{old} = 1/2772$, the iteration for $e = 2$ ends with $\mathbf{Pa}_2 = \emptyset$.

iteration $e = 3$:

$m = order_3 : m = 3$;

$\mathbf{Pa}_3 = \emptyset$;

$Score_{old} = f(3, \emptyset)$;

$s_3 = 2$;

Considering the same analysis used for iteration $e = 1$, for $m = 3$ we have:

$N_{3j1} = 4$, cases where $Y_3 = 0$ (1,5,7 and 8);

$N_{3j2} = 6$, cases where $Y_3 = 1$ (2,3,4,6 and 9);

$N_{3j} = 10$.

$$f(3, \mathbf{Pa}_3) = \prod_{j=1}^{t_3} \frac{(s_3-1)!}{(N_{3j}+s_3-1)!} \prod_{k=1}^{s_3} (N_{3jk})!$$

$$f(3, \emptyset) = \frac{(2-1)!}{(10+2-1)!} \prod_{k=1}^2 (N_{3jk})! = \frac{1!}{11!} * 4! * 6! = 1/2310$$

$$Score_{old} = 1/2310$$

Note that $Pred(Y_3) = \{Y_1, Y_2\}$, then, in Step 8, we have to calculate $argmax(f(3, \mathbf{Pa}_3 \cup \{Y_1\}), f(3, \mathbf{Pa}_3 \cup \{Y_2\}))$.

Given:

$$t_3 = 2 \text{ (} Y_1 = 0 \text{ or } Y_1 = 1 \text{);}$$

$$N_{311} = 3, \text{ cases where } Y_1 = 0 \text{ and } Y_3 = 0 \text{ (1,7 and 8);}$$

$$N_{312} = 2, \text{ cases where } Y_1 = 0 \text{ and } Y_3 = 1 \text{ (4 and 10);}$$

$$N_{321} = 1, \text{ case where } Y_1 = 1 \text{ and } Y_3 = 0 \text{ (5);}$$

$$N_{322} = 4, \text{ cases where } Y_1 = 1 \text{ and } Y_3 = 1 \text{ (2,3,6 and 9);}$$

$$N_{31} = 5;$$

$$N_{32} = 5;$$

$$f(3, \{Y_1\}) = \prod_{j=1}^{t_3} \frac{(s_3-1)!}{(N_{3j}+s_3-1)!} \prod_{k=1}^{s_3} (N_{3jk})!$$

$$f(3, \{Y_1\}) = \frac{(2-1)!}{5+2-1)!} \prod_{k=1}^2 (N_{31k})! * \frac{(2-1)!}{5+2-1)!} \prod_{k=1}^2 (N_{32k})!$$

$$f(3, \{Y_1\}) = \frac{(1)!}{(6)!} * (N_{311})! * (N_{312})! * \frac{(1)!}{(6)!} * (N_{321})! * (N_{322})!$$

$$f(3, \{Y_1\}) = \frac{1!}{6!} * 3! * 2! * \frac{1!}{6!} * 1! * 4! = 1/1800$$

Given:

$$t_3 = 2 \text{ (} Y_2 = 0 \text{ or } Y_2 = 1 \text{);}$$

$$N_{311} = 2, \text{ case where } Y_2 = 0 \text{ and } Y_3 = 0 \text{ (7 and 10);}$$

$$N_{312} = 3, \text{ cases where } Y_2 = 0 \text{ and } Y_3 = 1 \text{ (2,4 and 9);}$$

$$N_{321} = 3, \text{ cases where } Y_2 = 1 \text{ and } Y_3 = 0 \text{ (1,5 and 8);}$$

$$N_{322} = 2, \text{ cases where } Y_2 = 1 \text{ and } Y_3 = 1 \text{ (3 and 6);}$$

$$N_{31} = 5;$$

$$N_{32} = 5;$$

$$f(3, \{Y_2\}) = \prod_{j=1}^{t_3} \frac{(s_3-1)!}{(N_{3j}+s_3-1)!} \prod_{k=1}^{s_3} (N_{3jk})!$$

$$f(3, \{Y_2\}) = \frac{(2-1)!}{5+2-1)!} \prod_{k=1}^2 (N_{31k})! * \frac{(2-1)!}{5+2-1)!} \prod_{k=1}^2 (N_{32k})!$$

$$f(3, \{Y_2\}) = \frac{(1)!}{(6)!} * (N_{311})! * (N_{312})! * \frac{(1)!}{(6)!} * (N_{321})! * (N_{322})!$$

$$f(3, \{Y_2\}) = \frac{1!}{6!} * 2! * 3! * \frac{1!}{6!} * 3! * 2! = 1/3600$$

Since $f(3, \{Y_1\}) = 1/1800 > f(3, \{Y_2\}) = 1/3600$, then $Score_{new} = f(3, \{Y_1\}) = 1/1800$, $i = Y_1$, $\mathbf{Pa}_3 = \mathbf{Pa}_3 \cup \{Y_1\}$ and $Score_{old} := Score_{new}$ at Step 11.

Now, the next iteration of the algorithm for $e = 3$ considers adding the remaining predecessor of Y_3 , namely Y_2 , to the parents of Y_3 .

Given:

$$t_3 = 4 (Y_1 = 0, Y_2 = 0 \text{ or } Y_1 = 0, Y_2 = 1 \text{ or } Y_1 = 1, Y_2 = 0 \text{ or } Y_1 = 0, Y_2 = 1);$$

$$N_{311} = 2, \text{ cases where } Y_1 = 0, Y_2 = 0 \text{ and } Y_3 = 0 (7 \text{ and } 10);$$

$$N_{312} = 1, \text{ case where } Y_1 = 0, Y_2 = 0 \text{ and } Y_3 = 1 (4);$$

$$N_{321} = 2, \text{ cases where } Y_1 = 0, Y_2 = 1 \text{ and } Y_3 = 0 (1 \text{ and } 8);$$

$$N_{322} = 0, \text{ cases where } Y_1 = 0, Y_2 = 1 \text{ and } Y_3 = 1 ();$$

$$N_{331} = 0, \text{ cases where } Y_1 = 1, Y_2 = 0 \text{ and } Y_3 = 0 ();$$

$$N_{332} = 2, \text{ cases where } Y_1 = 1, Y_2 = 0 \text{ and } Y_3 = 1 (2 \text{ and } 9);$$

$$N_{341} = 1, \text{ case where } Y_1 = 1, Y_2 = 1 \text{ and } Y_3 = 0 (5);$$

$$N_{342} = 2, \text{ cases where } Y_1 = 1, Y_2 = 1 \text{ and } Y_3 = 1 (3 \text{ and } 6);$$

$$N_{31} = 3;$$

$$N_{32} = 2;$$

$$N_{33} = 2;$$

$$N_{34} = 3;$$

$$f(3, \mathbf{Pa}_3 \cup \{Y_2\}) = f(3, \{Y_1, Y_2\})$$

$$f(3, \{Y_1, Y_2\}) = \prod_{j=1}^{t_3} \frac{(s_3-1)!}{(N_{3j}+s_3-1)!} \prod_{k=1}^{s_3} (N_{3jk})!$$

$$f(3, \{Y_1, Y_2\}) = \prod_{j=1}^{t_3} \frac{(s_3-1)!}{(N_{3j}+s_3-1)!} \prod_{k=1}^{s_3} (N_{3jk})!$$

$$f(3, \{Y_1, Y_2\}) = \frac{(2-1)!}{3+2-1)!} \prod_{k=1}^2 (N_{31k})! * \frac{(2-1)!}{2+2-1)!} \prod_{k=1}^2 (N_{32k})! * \frac{(2-1)!}{2+2-1)!} \prod_{k=1}^2 (N_{33k})! * \frac{(2-1)!}{3+2-1)!} \prod_{k=1}^2 (N_{41k})!$$

$$f(3, \{Y_1, Y_2\}) = \frac{(1)!}{(4)!} * (N_{311})! * (N_{312})! * \frac{(1)!}{(3)!} * (N_{321})! * (N_{322})! * \frac{(1)!}{(3)!} * (N_{331})! * (N_{332})! * \frac{(1)!}{(4)!} * (N_{341})! * (N_{342})!$$

$$f(3, \{Y_1, Y_2\}) = \frac{1!}{4!} * 2! * 1! * \frac{1!}{3!} * 2! * 0! * \frac{1!}{3!} * 0! * 2! * \frac{1!}{4!} * 1! * 2! = 1/1296$$

$$Score_{new} = 1/1296$$

Since $Score_{new} = 1/1296 > Score_{old} = 1/1800$, $\mathbf{Pa}_3 = \mathbf{Pa}_3 \cup \{Y_2\}$, $Score_{old} := Score_{new}$ at Step 11 and the iteration for $e = 3$ ends with $\mathbf{Pa}_3 = \{Y_1, Y_2\}$.

The parents output for each node are:

$$\bullet Y_1: \mathbf{Pa}_1 = \emptyset;$$

$$\bullet Y_2: \mathbf{Pa}_2 = \emptyset;$$

$$\bullet Y_3: \mathbf{Pa}_3 = \{Y_1, Y_2\}$$

Then the learned structure is:

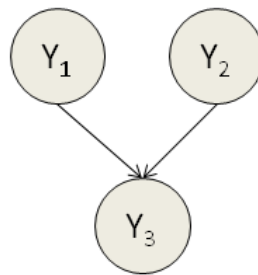


Figure 22: An example of structure learned by K2 Algorithm.

APPENDIX B – CUTTING EDGE EVOLUTIONARY ALGORITHMS

In the following subsections we detail some Pareto and scalarizing function based algorithms. These algorithms have been chosen aiming to observe: 1- a very popular MOEA described in the literature (NSGA-II); 2- a MOEA with local search (S-MOGLS); 3- a quite recent approach applied to many objectives optimization problems (NSGA-III); 4- an EDA that incorporates a Pareto based selection technique (MBN-EDA); and 5- a scalarizing function-based algorithm, MOEA/D, that decomposes a MOP into a number of mono-objective problems using the same scalarizing function with a different weight vector. Note that these algorithms can be used on continuous and discrete MOPS, but this work they will be explored in the context of discrete and combinatorial optimization problems.

B.1 NSGA-II

NSGA-II is the one of the most applied Pareto dominance-based algorithms in the literature (DEB et al., 2002) and it is described in Algorithm 7. Its fitness evaluation is based on a rank assignment mechanism. An offspring population of size N is generated by binary tournament selection, crossover, and mutation from a current population of size N , Steps 5 – 8. These two populations are combined into a merged population of size $(N + N)$, Step 9. The next population is constructed by choosing (based on the Non-dominated Sorting) the best N solutions from the merged population, Step 11. ND is the final set of non-dominated solutions.

The initialization process randomly generates every vector \mathbf{x} of the initial population Pop^1 , a total of N vector solutions is generated. For each solution in Pop^1 the values of all objectives are calculated based on the objective functions and set in \mathbf{z} . Therefore, N joint vectors (\mathbf{x}, \mathbf{z}) are obtained and complete the population at first generation. Finally, the initialization procedure sets $g = 1$.

In order to sort individuals of the current population, NSGA-II uses Non-dominated Sorting technique (SRINIVAS; DEB, 1994) in Step 6.

After classifying the population at g^{th} generation (Pop^g) into Tot_F sub-populations

Algorithm 7: NSGA-II

INPUT: N , population size;

N_{best} , maximum of parents selected to provide the offspring population;

Max_{eval} , maximum number of solutions evaluation.

OUTPUT: ND , the final set of non-dominated solutions;

{Initialization}

1: $Pop^1(\mathbf{x}) = \text{Random}(N)$;

2: $Pop^1(\mathbf{z}) = \text{Fitness}(Pop^1(\mathbf{x}))$;

3: $e = 0$; { e is the current number of solutions evaluation}

4: $g = 1$;

{NSGA-II: main loop}

5: **while** $e \leq Max_{eval}$ **do**

 {Non-dominated Sorting}

 {Defining Tot_F Pareto fronts: from the best ($i = 1$) to the worst}

6: $F_1 \dots F_{Tot_F} = \text{ParetoDominance}(Pop^g)$;

 {NSGA-II: Selecting the parents}

7: $Pop_{best}^g = \text{Selection}(N_{best}, F_1 \dots F_{Tot_F})$; {crowding distance to binary tournament}

 {NSGA-II: Perform crossover and mutation operator to provide offspring population}

8: $Pop_{offspring}^g = \text{GeneticOperator}(Pop_{best}^g)$;

9: $Pop_{merged}^g = Pop^g \cup Pop_{offspring}^g$;

10: $F_1 \dots F_{Tot_F} = \text{ParetoDominance}(Pop_{merged}^g)$;

11: $Pop^{g+1} = \text{Selection}(N, F_1 \dots F_{Tot_F})$; {Truncation Selection}

12: $g = g + 1$;

13: **end while**

14: $ND = Pop^g(\mathbf{x})$;

$(F_1, F_2, \dots, F_{TotF})$ using Non-dominated Sorting technique, the method assigns a CD (DEB et al., 2002) to every solution in each F_i . CD is based on a technique proposed by Deb et al. (2002) which tries to preserve the dispersion of the population Pop^g . Initially the solutions in F_i are sorted in ascending order of the corresponding first objective. For each point \mathbf{z}^k in F_i the distance $d_1(\mathbf{z}^k) = z_1^{k+1} - z_1^{k-1}$ is calculated. Sorting the points similarly to the other objectives we obtain $d_2(\mathbf{z}^k), \dots, d_r(\mathbf{z}^k)$. So, for the estimation of the total of points which are located around \mathbf{z}^k we used $CD = dist(\mathbf{z}^k) = \sum_{r=1}^R d_r(\mathbf{z}^k)$.

In Step 7 of Algorithm 7, N_{best} individuals are selected through binary tournament to compose the Pop_{best}^g , taking at first DR and secondly (in case of ties) the CD criterion (solutions with the greatest CD is chosen). The *GeneticOperator* procedure randomly selects individuals from Pop_{best}^g to generate using genetic operators (crossover and mutation) a $Pop_{offspring}^g$ population. Pop_{merged}^g is the union of Pop^g and $Pop_{offspring}^g$. Finally, truncation selection in Step 11 selects the best N solutions from Pop_{merged}^g to compose the next generation population Pop^{g+1} .

B.2 S-MOGLS

Simple Multi-Objective Genetic Local Search Algorithm (S-MOGLS) is a memetic algorithm proposed by Ishibuchi et al. (2008) that implements a hybrid algorithm of NSGA-II and local search.

The local search is probabilistically applied to solutions in the offspring population $Pop_{offspring}^g$. Everytime a solution in $Pop_{offspring}^g$ is improved by local search, it is included in the improved population Pop_I^g . The next population is chosen from the current population Pop^g , the offspring population $Pop_{offspring}^g$, and the improved population Pop_I^g in the same way manner as performing in NSGA-II. When probability P_{ls} (probability of LS occurrence) is zero S-MOGLS is exactly the same as NSGA-II.

B.2.1 S-MOGLS LOCAL SEARCH PROCEDURE

An initial solution for LS is selected from the offspring population $Pop_{offspring}^g$ by the tournament selection with replacement using following the weighted sum fitness function.

$$\mathbf{f}(\mathbf{x}) = \lambda_1 f_1(\mathbf{x}) + \lambda_2 f_2(\mathbf{x}) + \dots + \lambda_R f_R(\mathbf{x}) \quad (34)$$

where $\{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_R(\mathbf{x})\}$ are the objective values $\{z_1, z_2, \dots, z_R\}$, $\boldsymbol{\lambda} =$

$(\lambda_1, \lambda_2, \dots, \lambda_R)$ is a non-negative weight vector for R -objectives controlled by the parameter $d = \sum_{r=1}^R \lambda_r$, where $\lambda_r \in \{0, 1, \dots, d\}$ for $r = \{1, 2, \dots, R\}$, i.e, we have six weight vectors $(0, 0, 2)$, $(0, 2, 0)$, $(2, 0, 0)$, $(0, 1, 1)$, $(1, 0, 1)$, $(1, 1, 0)$ when $d = 2$ for $R = 3$ objectives.

LS is applied to the chosen solution with probability P_{ls} . A neighbor is randomly generated and the current solution is replaced if this neighbor is better. This means that each LS adopted by S-MOGLS accepts the first improved neighbor instead of the best improved in the neighborhood. After a neighbor is accepted, LS continues to search for better solutions around the updated current solution (accepted neighbor). The total number of examined neighbors in a series of LS from the initial solution, N_{ls} , is the termination condition. The selection of an initial solution and the probabilistic application of LS are iterated N times in each generation, where N is the population size. It should be noted that a weight vector is randomly drawn whenever an initial solution for LS is selected. This means that the initial solution can be evaluated through a unique fitness value using different weights for each objective. According to Ishibuchi et al. (2008), this provides a wide range of objectives values in the objective space, which can increase quite well the quality of the Pareto front approximation.

B.3 NSGA-III

NSGA-III, proposed by Deb and Jain (2014), is a new approach of reference-point-based many-objective EA that follows the NSGA-II main steps and emphasizes non-dominated population members yet close to a set of reference points. The basic NSGA-III framework is similar to the original NSGA-II algorithm (DEB et al., 2002) with main changes focused on the selection operator, where CD is replaced. The maintenance of diversity in NSGA-III is achieved by supplying and adaptively updating a number of well-spread reference points.

The main NSGA-III steps are presented in Algorithm 8.

NSGA-III starts with a random population of size N and a set of pre-established R -dimensional reference points on a unit hyper-plane having a normal vector of ones covering the entire \mathbf{R}_+^R region. The hyper-plane is placed in a manner so that it intersects each objective axis at one.

Deb and Jain (2014) have used the approach described by Das and Dennis (1998) for generating the structured reference points Rp^N .

If H divisions are considered along each objective, the total number of reference points, which are often considered the same of the population size (N) (DEB; JAIN, 2014) in an

Algorithm 8: NSGA-III

INPUT: N structured reference points (Rp^N);
 Max_{eval} , maximum number of solutions evaluation;

OUTPUT: ND , the final set of non-dominated solutions;

{Initialization}

- 1: $Pop^1(\mathbf{x}) = \text{RandomVector}(N)$;
- 2: $Pop^1(\mathbf{z}) = \text{Fitness}(Pop^1(\mathbf{x}))$;
- 3: $e = 0$; {e is the current number of solutions evaluation}
- 4: $g = 1$;
- 5: $S^1 = \emptyset$;

{NSGA-III: main loop}

- 6: **while** $e \leq Max_{eval}$ **do**
 {Perform crossover and mutation operator to provide offspring population}
- 7: $Pop_{offspring}^g = \text{GeneticOperator}(Pop^g)$;
- 8: $Pop_{merged}^g = Pop^g \cup Pop_{offspring}^g$;
 {Non-dominated Sorting}
- {Defining Tot_F Pareto fronts: from the best ($i = 1$) to the worst}
- 9: $F_1 \dots F_{Tot_F} = \text{ParetoDominance}(Pop_{merged}^g)$;
- 10: **repeat**
 11: $S^g = S^g \cup F_i$;
- 12: $i = i + 1$;
- 13: **until** $|S^g| \geq N$
- 14: $F_k = F_i$; {Last front to be included}
- 15: **if** $|S^g| = N$ **then**
 16: $Pop^{g+1} = S^g$;
- 17: **break**;
- 18: **else**
 19: $Pop^{g+1} = \cup_{j=1}^{k-1} F_j$;
- 20: $K = N - |Pop^{g+1}|$; {Points to be chosen from F_k };
- 21: $(Pop^{g+1}(\mathbf{z}), Rp^{ref}) = \text{Normalize}(S, Rp^N)$; {Normalize objectives and create reference set Rp^{ref} };
- 22: $[\pi(\mathbf{z}), d(\mathbf{z})] = \text{Associate}(S^g, Rp^{ref})$; {Associate each individual \mathbf{z} (objective vector) of S^g with a reference point. $\pi(\mathbf{z})$: closest reference point, $d(\mathbf{z})$: distance between \mathbf{z} and $\pi(\mathbf{z})$ };
- 23: $\rho_j = \sum_{\mathbf{z} \in S^g / F_k} ((\pi(\mathbf{z}) = j) ? 1 : 0)$; {Compute niche count of reference point $j \in Rp^{ref}$ };
- 24: $Pop^{g+1} = \text{Niching}(K, \rho_j, \pi, d, Rp^{ref}, F_k, Pop^{g+1})$; {Choose K members one at a time from F_k to construct Pop^{g+1} };
- 25: **end if**
- 26: $g = g + 1$;
- 27: **end while**
- 28: $ND = Pop^g(\mathbf{x})$;

R -objective problem is given by Equation 35.

$$N = C_{R+H-1}^H \quad (35)$$

For example, in a three-objective problem ($R = 3$) the reference points are created on a triangle with the apex at $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. If 13 divisions ($H = 13$) are chosen for each objective axis, $N = 105$ reference points will be created.

Since the reference points are widely distributed on the entire normalized hyper-plane, the obtained objectives vectors are also distributed likely on or near the Pareto front.

The offspring population is created using the GeneticOperator procedure, Step 7, as in NSGA-II, and merged with parent population, Pop_{merged}^g .

At a generation g , first the combined population Pop_{merged}^g is sorted according to different non-dominated levels ($F_1, F_2 \dots F_{TotF}$), as in the Non-dominated Sorting procedure. Then, each non-dominated level is selected, one at a time, to construct a new population S^g , starting from F_1 and proceeding with each F_k .

In most situations, the last accepted level can only be partially accepted. In such a case, only those solutions that maximize the diversity of the last accepted front are chosen from S^g using a niche-preserving operator, which we describe next (Steps 18 – 25).

First, the corresponding objective vector of each solution from S^g and the last front is normalized by using the current population spread (in the objective space) so that all objective vectors and reference points have commensurate values, as shown in Step 21.

After adaptively normalizing each objective based on the extent of individuals of S^g in the objective space, we need to associate each population individual (in the objective space) with a reference point in Step 22. For this purpose we calculate the perpendicular distance of the corresponding objective vector of each solution from S^g with a reference line created by joining the origin with a supplied reference point (SEADA; DEB, 2015). The reference point whose reference line is closest to a population individual in the normalized objective space is considered to be associated with the population individual.

A reference point can have no one or several population individuals associated with it. We count the number of population individuals from $Pop^{g+1} = S^g / F_k$ that are associated with each reference point as a niche count, ρ_j , for the j -th reference point. First, we identify the reference point set $J_{min} = \{j : \operatorname{argmin}_{j \in R^{ref}} \rho_j\}$ having minimum ρ_j . In the case of multiple such reference points, one ($j \in J_{min}$) is chosen at random.

When there is no associated Pop^{g+1} individual to the reference point j , $\rho_j = 0$ and we have two possibilities: there exists one or more individuals of S^g in front F_k that can be associated with the reference point j and the one having the shortest perpendicular distance from the reference line is added to Pop^{g+1} ; or, the front F_k does not have any individual associated with the reference point j and the reference point is not considered. When there exists one individual associated with the reference point in S^g/F_k , $\rho_j \geq 1$, then a randomly chosen individual from front F_k associated with the reference point j is added to Pop^{g+1} . The procedure is repeated for a total of K times until population Pop^{g+1} is fulfilled.

In order to provide non-dominated individuals, the whole process is then expected to find one population individual corresponding to each supplied reference point close to the Pareto front.

B.4 MBN-EDA

MBN-EDA was proposed by Karshenas et al. (2014) as a new MOEDA based on joint probabilistic modeling of objectives and variables. This EDA uses a multidimensional BN (MBN) as its probabilistic model, where the objectives are continuous-valued class nodes and the variables are modeled as features nodes. The model considers relations between the problem variables encoded at the feature subgraph, between objectives, at the class subgraph and between variables and objectives, representing by direct iterations at the bridge subgraph.

The probabilistic MBN model can capture characteristics of selected solutions according to their objective values, and samples new candidate solutions aiming to find Pareto optimal solutions.

Figure 23 illustrates the algorithm. A selection mechanism, e.g., non-dominated sorting and truncation selection, are used to select a subset of solutions. These solutions are extended in order to keep their objectives values encoding as a joint vector. These extended solutions, comprising values for both variables and objectives, are used to serve as a dataset for the MBN learning process.

The model samples new candidate solutions from the learned MBN, according to the values of both objectives and variables. Finally, these new solutions are added to the population based on a replacement strategy.

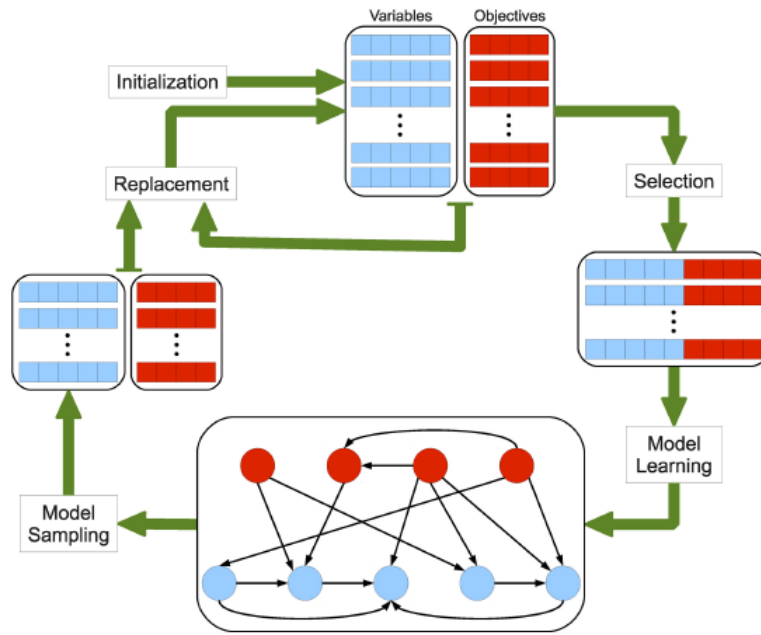


Figure 23: An overview of the proposed MBN-EDA by Karshenas et al. (2014).

B.5 MOEA/D

MOEA/D is based on aggregation approaches, as those presented in Section 2.3, once it decomposes a MOP into a number of single objective optimization subproblems. The objective of each subproblem is a linear (or nonlinear) weighted aggregation of all individual objectives in the MOP. Neighborhood relations among these subproblems depend on distances among their aggregation weight vectors.

A set of weight vectors, where each vector $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_R)$ is specified by the following relations for a R -objective problem:

$$\lambda_1 + \lambda_2 + \dots + \lambda_R = 1 \quad (36)$$

$$\lambda_r \in \left\{ 0, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H} \right\}, r = 1, 2, \dots, R \quad (37)$$

where H is a user defined positive integer. MOEA/D uses all weight vectors satisfying Equation 36 and relation 37.

Thus, each subproblem is simultaneously optimized using mainly information from its neighbor subproblems (ISHIBUCHI et al., 2015).

The MOEA/D is presented in Algorithm 9.

Algorithm 9: MOEA/D

INPUT: N , number of sub-problems;
 $\lambda_1 \dots \lambda_N$, a uniform spread of weight vectors;
 T , the number of the weight vectors in the neighborhood of each weight vector;
 Max_{eval} , maximum number of solutions evaluation.

OUTPUT: ND , all non-dominated solutions found during the search;

{Initialization}

- 1: $B(i) = \{i_1, \dots, i_T\} = \text{EuclideanDistance}(\lambda^1 \dots \lambda^N), T$; {For $i = 1 \dots N$, set the indexes of the T closest weight vectors to λ^i }
- 2: $Pop^1(\mathbf{x}) = \text{RandomVector}(N)$;
- 3: $Pop^1(z) = \text{Fitness}(Pop^1(\mathbf{x}, \lambda))$;
- 4: $ND = \emptyset$;
- 5: $e = 0$; { e is the current number of solutions evaluation}
- 6: $g = 1$;
- {MOEA/D: main loop}
- 7: **while** $e \leq Max_{eval}$ **do**
- 8: **for** $i = 1$ to N **do**
 - {Perform crossover and mutation operator to provide a new solution from \mathbf{x}^k and \mathbf{x}^l , $k, l \in B(i)$ }
 - 9: $\mathbf{x}^{offspring} = \text{GeneticOperator}(\mathbf{x}^k, \mathbf{x}^l)$; {repair method if infeasible solution}
 - {Update of neighbor solutions}
 - 10: **for each** $j \in B(i)$ **do**
 - 11: **if** $(\text{Fitness}(\mathbf{x}^{offspring}, \lambda^j) \geq \text{Fitness}(\mathbf{x}^j, \lambda^j))$ **then**
 - 12: $\mathbf{x}^j = \mathbf{x}^{offspring}$;
 - 13: $z(j) = \text{Fitness}(\mathbf{x}^{offspring}, \lambda^j)$;
 - 14: **end if**
 - 15: **end for**
 - {Update ND }
 - {Remove from ND all the vector dominated by $\mathbf{x}^{offspring}$ };
 - 16: **if** (no vectors in ND dominate $\mathbf{x}^{offspring}$) **then**
 - 17: $ND = ND \cup \mathbf{x}^{offspring}$;
 - 18: **end if**
 - 19: **end for**
 - 20: $g = g + 1$;
 - 21: **end while**

In *Initialization*, set $B(i)$ contains the indexes of the T closest vectors of λ^i . We use the Euclidean distance (*EuclideanDistance*) procedure to measure the closeness between any two weight vectors. Therefore, λ^i 's closest vector is itself, and then $i \in B(i)$. If $j \in B(i)$, the j -th subproblem can be regarded as a neighbor of the i -th subproblem.

The initialization process randomly generates every vector \mathbf{x} of the initial population Pop^1 , so a total of N vector solutions is generated. For each solution in Pop^1 the fitness z is calculated based on one of the scalarizing function approach presented in Section 2.3. ND is an

external population containing all non-dominated solutions found during the search.

The *GeneticOperator* procedure randomly selects two indexes, k and l from $B(i)$, and then generates a new solution $\mathbf{x}^{offspring}$ from \mathbf{x}^k and \mathbf{x}^l by using genetic operators (crossover and mutation).

Steps 11 to 16 consider all the neighbors of the i -th subproblem, replacing \mathbf{x}^j with $\mathbf{x}^{offspring}$ if $\mathbf{x}^{offspring}$ performs better than \mathbf{x}^j with regard to the j -th subproblem. z is needed in computing the value of $z = \text{Fitness}(\mathbf{x}^{offspring} | \boldsymbol{\lambda}^j)$ in Step 14. In Step 17 we remove from ND all the vectors dominated by $\mathbf{x}^{offspring}$ and add $\mathbf{x}^{offspring}$ if no vectors in NP dominates $\mathbf{x}^{offspring}$.

The number of subproblems (N) and weight vectors $\lambda^1, \dots, \lambda^N$ in MOEA/D is controlled by the parameter H Zhang and Li (2007). More precisely, $\lambda^1, \dots, \lambda^N$ are all the weight vectors in which each individual weight takes a value from $\{\frac{0}{H}, \frac{1}{H}, \dots, \frac{H}{H}\}$. Therefore, the number of such vectors is calculated by Equation 38.

$$N = C_{R+H-1}^{R-1} \quad (38)$$

The same method is applied in NSGA-III for the N supplied reference points.