Analysis of linkage learning in evolutionary optimization

Jean Paulo Martins

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Jean Paulo Martins

Análise da aprendizagem de ligações em otimização evolutiva

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Alexandre C. B. Delbem

**USP – São Carlos
Julho de 2015**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

M386a Martins, Jean Paulo
 Analysis of linkage learning in evolutionary
 optimization / Jean Paulo Martins; orientador
 Alexandre Claudio Botazzo Delbem. -- São Carlos,
 2015.
 136 p.

 Tese (Doutorado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2015.

 1. linkage-learning. 2. multimodality. 3. EDA.
 4. MOEDA. I. Delbem, Alexandre Claudio Botazzo,
 orient. II. Título.

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Jean Paulo Martins

Analysis of linkage learning in evolutionary optimization

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Alexandre C. B. Delbem

**USP – São Carlos
July 2015**

“In this book we find a subterrestrial at work, digging, mining, undermining [...] He might even be called happy, despite his labours in the dark. Does it not seem as if some faith were leading him on, some solace recompensing him for his toil? Or that he himself desires a long period of darkness, an unintelligible, hidden, enigmatic something, knowing as he does that he will in time have his own morning, his own redemption, his own rosy dawn? - Yea, verily he will return: ask him not what he seeketh in the depths; for he himself will tell you, this apparent Trophonius and subterrestrial, whensoever he once again becomes man. One easily unlearns how to hold one’s tongue when one has for so long been a mole, and all alone, like him.”

The Dawn of Day
F. Nietzsche

Agradecimentos

Agradeço a todos que me acompanham neste caminho tortuoso, de escolhas passionais num mundo envolto por profissionais. Sei que a cada mudança de direção assumimos a responsabilidade de perder, e muitas mudanças desejadas vem acompanhadas de perdas indesejadas. Não há como recuperar o tempo que me privei da presença de minha família e de meus amigos e não haveria razões que os fizessem entender meus motivos, mas entenderam. Meu coração se partia em todas idas e vindas, do olhar de Maria, das dores tão minhas, da saudade do José, da falta de Cris, de Lú. A alegria de ver Gustavo na vila verde, de ouvir o Marquês de Michel, o Ari ali na beira, o Montanaro lá em Hollywood, e muitos outros amigos, se tornou menos frequente.

Mas a amizade é amor que brota em todo lugar, e aqui encontrei mais amigos, pessoas que também se importam e acreditam que as alegorias de nossas profissionais vidas não superam os sentimentos do contato cotidiano. Amigos como o Antonio, Arnaldo, Diogo, Erinaldo, Gustavo Sabino, Helson e Marcilyanne com os quais passei grande parte desses últimos anos e que tornaram estresse e dificuldades em momentos de alegria e identificação que serão nossos para sempre.

Agradeço em especial todos que me incentivaram e inspiraram, Humberto Longo, Claudio Meneses, Alexandre Delbem, Carlos Fonseca, dentre outros. Esta tese é importante pra mim, mas tudo que aprendi com a vida e as pessoas ao longo desses quatro anos e oito meses é o que realmente me motiva a lhes agradecer.

Agradeço ainda a FAPESP, pelo apoio financeiro que tornou possível o doutorado (número de processo 2011/07792-4) e ao CNPq e programa *Ciência sem Fronteiras* pelo apoio financeiro que me possibilitou estadia em Portugal (número de processo 246565/2012-8).

Obrigado,
Jean Paulo Martins

Abstract

MARTINS, J. P. (2015). **Analysis of linkage learning in evolutionary optimization**. 136 pp. (*doctoral thesis*). Instituto de Ciências Matemáticas e Computação, ICMC, USP – São Carlos.

The supposed ubiquity of nearly-decomposable systems was interpreted by Holland (1975) as the rationale for the performance of Genetic Algorithms (GAs), the Building Block (BB) hypothesis. His seminal studies suggest more efficient GAs as viable, but only later on his ideas have become practically tangible in the context of Estimation of Distribution Algorithms (EDAs). EDAs employ probabilistic modeling so as to infer properties of the search space (BBs) that could be useful for the effectiveness of reproduction operators. In both, single- and multi-objective contexts, EDAs have emerged on the assumption there is a correlation between how much information a model can conceive and how effective reproduction operators can be. However, more recent results suggest the difficulties in producing accurate linkage models can prevent such a relation to be true. In other words, for some optimization problems linkage learning might not be able to produce accurate linkage models, hence EDAs would not outperform GAs. This thesis addresses the limits of linkage learning in the context of single- and bi-objective problems, regarding the influence of multimodality on the accuracy of the linkage models and the efficiency of EDAs. A theoretical analysis was performed in terms of additively separable functions and general conclusions are assessed through experimentation with instances of the *NK*-model and the Multidimensional Knapsack Problem (MKP). The results indicated that the accuracy of the linkage models tends to decrease as a result of increasing multimodality, which weakens pairwise dependencies and might lead to *pairwise independence* in extreme cases. Since most EDAs rely on bivariate statistics to estimate multivariate distributions, their applicability is limited to optimization problems within a certain range of multimodality. In multi-objective problems, on the other hand, some EDAs have shown better performance than GAs, which seemed as a contradiction since multi-objective problems are inherently multimodal. Our results suggest that in such cases the correlation among the objective functions becomes statistically evident, as a consequence, linkage learning models such correlation instead of problem's substructures, which is useful to obtain a better exploration of extreme regions of the objective space.

Keywords: Linkage learning, multimodality, multi-objective

Resumo

MARTINS, J. P. (2015). **Analysis of Linkage learning in evolutionary optimization**. 136 pp. (tese de doutorado). Instituto de Ciências Matemáticas e Computação, ICMC, USP – São Carlos.

A suposta ubiquidade de sistemas decomponíveis foi interpretada por Holland (1975) como o principal motivo para o desempenho dos algoritmos genéticos (Genetic Algorithms (GAs)). A hipótese de Building Blocks (BBs) sugere que algoritmos genéticos mais eficientes poderiam ser implementados, contudo, apenas anos depois essas ideias puderam ser avaliadas experimentalmente no contexto de algoritmos de estimação de distribuição (Estimation of Distribution Algorithms (EDAs)). EDAs utilizam modelos probabilísticos, estimados a partir da população, para inferir características do espaço de busca que poderiam ser utilizadas para implementar operadores de reprodução mais eficazes. Tanto em problemas mono- quanto multi-objetivo, EDAs emergiram sob a premissa de que a eficácia dos operadores de reprodução seria proporcional à representatividade dos modelos probabilísticos utilizados. No entanto, estudos recentes tem demonstrado que a dificuldade em se construir modelos confiáveis pode tornar essa premissa inviável. Ou seja, para certos problemas de otimização os modelos probabilísticos utilizados seriam, em geral, de baixa qualidade e, portanto, não produziriam operadores eficazes. Esta tese trata das limitações encontradas na construção de modelos probabilísticos (*linkage learning*) sob a perspectiva da multimodalidade dos problemas em questão. A análise teórica considerou problemas aditivamente separáveis, enquanto a generalização das conclusões foi investigada em instâncias do modelo NK -landscapes e do problema da mochila multidimensional (Multi-dimensional Knapsack Problem (MKP)). Os resultados indicaram que a acurácia dos modelos probabilísticos é se relaciona inversamente ao grau de multimodalidade da função objetivo e que, em casos de extrema multimodalidade a construção de modelos probabilísticos confiáveis pode ser tornar ineficaz. Este resultado poderia inviabilizar o uso de EDAs no contexto multiobjetivo, devido a intrínseca multimodalidade de tais problemas. No entanto, observou-se que apesar da ausência de estatísticas confiáveis sobre cada uma das funções objetivo, a correlação entre elas se torna estatisticamente observável e útil aos operadores de reprodução na manutenção da diversidade e controle convergência da população.

Keywords: EDAs, multimodalidade, linkage learning

Contents

Abstract	iii
List of Figures	xi
List of Algorithms	xv
List of Acronyms	xvii
1 Introduction	1
1.1 Scope and Objectives	3
1.2 Thesis Contributions	4
1.3 Thesis Outline	6
2 Evolutionary Optimization	7
2.1 Introduction	7
2.1.1 Optimization Problems	7
2.1.2 Unified Model of Evolutionary Algorithms	9
2.2 Genetic Algorithms	11
2.2.1 Fitness Assignment	12
2.2.2 Selection Operators	14
2.2.3 Variation Operators	15
2.2.4 Working Principles	18
2.3 Estimation of Distribution Algorithms	18
2.3.1 Univariate Factorizations	19
2.3.2 Bivariate Factorizations	21
2.3.3 Multivariate Factorizations	22
2.4 Multiobjective Optimization Extensions	26
2.4.1 Fitness Assignment	26
2.4.2 Reproduction Operators	31
2.5 Conclusions	34

3	Fundamentals of Linkage-learning	37
3.1	Introduction	37
3.1.1	Information Theory Background	40
3.1.2	Necessary Conditions for Learning	41
3.1.3	Origins of Learning Difficulty	42
3.2	Methodology	43
3.2.1	Additively Separable Functions	43
3.2.2	Population Size Assumption	46
3.3	Results and Discussions	47
3.3.1	Linkage-learning Difficulty	47
3.3.2	Upper-bounds on Linkage-learning Difficulty	52
3.4	Conclusions	55
4	Multimodality and the Limits of Linkage-learning	57
4.1	Introduction	57
4.2	Methodology	58
4.2.1	The NK-landscapes Model	59
4.2.2	Settings of the Experiments	60
4.3	Results and Discussion	60
4.3.1	Effectiveness	60
4.3.2	Efficiency	64
4.3.3	Range of Usefulness	66
4.4	Conclusions	68
5	Linkage-learning and the Multidimensional Knapsack Problem	71
5.1	Introduction	71
5.1.1	Efficiencies and the Core Concept	73
5.1.2	Chu and Beasley Genetic Algorithm	74
5.2	Methodology	75
5.2.1	Benchmark Instances	76
5.2.2	Experiment Settings	76
5.3	Results and Discussion	78
5.3.1	Average Gap from the Optimum	78
5.3.2	Number of Function Evaluations	80
5.3.3	Average Running-time	81
5.3.4	On the Usefulness of Linkage-trees	82
5.4	Conclusions	86

6	Reproductive Bias and Linkage-learning in Multiobjective Optimization	89
6.1	Introduction	89
6.2	Methodology	91
6.2.1	Bi-objective Additively Separable Functions	91
6.2.2	Bi-objective Non-separable Functions	93
6.2.3	Goals of Multiobjective Crossover	95
6.2.4	Reproductive Bias of Crossover	97
6.2.5	Handling Bias with Restricted Mating	100
6.2.6	Handling Bias with Linkage-learning	101
6.3	Results and Discussion	101
6.3.1	The ρMNK -landscapes Problem	102
6.3.2	Traditional Crossover bias	102
6.3.3	Crowding-biased Uniform Crossover	105
6.3.4	Linkage-guided Crossover bias	108
6.3.5	Measuring Crossover Bias	114
6.4	Conclusions	116
7	Conclusions	117
	Bibliography	119

List of Figures

2.1	Pareto-optimal solution $\mathbf{x}^*$, a dominated solution $\mathbf{x}$, their respective objective vectors $\mathbf{z}^*$, $\mathbf{z}$ and their relative ordering in $\mathcal{F}$	9
2.2	Example of one-point, two-point, n -point and uniform crossover performed between parents $\mathbf{x}$ and $\mathbf{y}$, producing a solution $\mathbf{z}$	17
2.3	A linkage tree and the possible subsets that can be obtained.	25
2.4	Ranking assignment as implemented in MOGA. Objective vectors get ranking proportional to the number of objective vectors dominating them.	27
2.5	Fitness assignment by non-dominance sorting, as implemented by the NSGA. Solutions in the first fronts receive higher fitness values.	28
2.6	Examples of the crowding distance for objective vectors $\mathbf{z}_2$ and $\mathbf{z}_3$. A small crowding distance (a small box in the figure) means the objective vector is in a more populated region in objective space.	29
2.7	The Dominated hypervolume in relation to $\mathbf{z}_{\text{ref}}$ (shaded region) measures convergence and diversity simultaneously. The hatched region represents the contribution of $\mathbf{z}_1$ to the dominated hypervolume.	31
3.1	Population before (crosses) and after selection (circles). According to the number of local attractors statistical dependencies might be harder to detect.	42
3.2	All possible outcomes of a trap function with $k = 5$ (a), a bipolar function with $k = 6$ (b), a parity function with $k = 6$ and a parity/trap function with $k = 6/k = 5$	44
3.3	Figure (a) shows that accurate linkage-learning from mutual information is harder for larger values of k (small δ_I values). Figure (b), on the other hand, shows that δ_H for the k -trap is stable as k grows, whereas the k -bipolar is not (due to the increase of multimodality according to k).	52
4.1	Average trajectory of the best fitness of each algorithm along $\theta = N \times 10^3$ fitness evaluations. All experiments with $K = 2, 5, 10, 15$ and $N = 100, 500, 1000$ were performed with a fixed $\mu = 100$	61

4.2	Average trajectory of the best fitness of each algorithm along $\theta = N \times 10^3$ fitness evaluations. All experiments with $K = 2, 5, 10, 15$ and $N = 100, 500, 1000$ were performed with a fixed $\mu = 500$	62
4.3	Average trajectory of the best fitness of each algorithm along $\theta = N \times 10^3$ fitness evaluations. All experiments with $K = 2, 5, 10, 15$ and $N = 100, 500, 1000$ were performed with a fixed $\mu = 1000$	63
4.4	Distributions of the first-hitting time with $K = 2, 5, 10, 15$, $N = 100, 500, 1000$ and $\mu = 100$	65
4.5	Distributions of the first-hitting time with $K = 2, 5, 10, 15$, $N = 100, 500, 1000$ and $\mu = 500$	66
4.6	For each pair (K, N) the values indicate the probability of the LTGA outperforming the rLTGA (according to the experiments). Linkage-learning was more useful in light-colored regions.	67
4.7	For each pair (K, N) , the values indicate how intensely the LTGA has outperformed the rLTGA (according to experiments). Light-colored regions indicate values of K in which significant benefits were obtained.	68
5.1	Average percentage gap from the best known solution for $d = 10, 30$ and $N = 100, 250, 500$. The averages were computed from the results for all instances. Figures (a),(b),(c),(g),(h),(i) show the gaps reached Without Diversity Preservation (WoDP). Figures (d),(e),(f),(j),(k),(l) show the gaps reached With Diversity Preservation (WDP).	79
5.2	Average number of function evaluations when the algorithms first hit their best solutions, for $d = 5, 10, 30$ and $N = 100, 250, 500$. The averages were computed from the results for all instances and with diversity preservation. Figures (a),(b),(c) show the number of evaluations for $d = 5$, Figures (d),(e),(f) for $d = 10$ and Figures (g),(h),(i) for $d = 30$	80
5.3	Average total time to perform 500,000 fitness evaluations for $d = 5, 10, 30$ and $N = 100, 250, 500$. The averages were computed from the results for all instances and with diversity preservation. Figures (a),(b),(c) show the number of evaluations for $d = 5$, Figures (d),(e),(f) for $d = 10$ and Figures (g),(h),(i) for $d = 30$	81
5.4	An example of linkage-tree built for an MKP instance with $N = 100$ and $d = 5$ at the first generation of the LTGA, after repair of the solutions. Two giant clusters are evident, on the left, converged variables, on the right, non-converged variables.	84

5.5	Examples of linkage-trees built for an MKP instance with $N = 100$ and $d = 5$ at the 2^{nd} , 4^{th} , 6^{th} , 8^{th} , 10^{th} and 12^{th} generations of the LTGA. Variables $x_j \in \mathcal{I} \setminus \mathcal{C}$ converged first and were grouped first. Decision variables $x_j \in \mathcal{C}$ have showed weak statistical dependencies, fact represented by high values of NVI	85
6.1	Pareto front of a bi-objective problem composed of $m = 6$ additively separable deceptive functions of size $k = 5$	93
6.2	Objective space for a bi-objective problem, $N = 15$, $k_1 = 5$, $k_2 = 3$. Bullets ($\bullet$) represent solutions generated by crossover between the extreme points in the front ($*$). Other objective vectors are represented by ($\circ$).	94
6.3	Points representing the offspring generated by unbiased (a) and biased (b) crossovers. The unbiased version exploit the regions of all possible aggregations uniformly and enables the exploration of a more well-spread set of objective vectors (shaded region).	96
6.4	Positioning of objective vectors obtained by different values for $0 \leq p \leq 1$. The expected distance between parents and offspring induces a bias in the population along the generations.	97
6.5	Number of bits from each parent in crossover. For any odd number of points, half the bits are expected to come from each parent. For even number of points, the value decreases exponentially and tends to $N/2$. In uniform crossover, $\mathcal{B}_n$ varies linearly according to p	99
6.6	Illustrative example of the impact of restricted mating on the creation of multiple bias in the objective space and improvement of the spreading of objective vectors.	100
6.7	Comparison between uniform crossover with $p = 0, 0.1, 0.2, 0.3, 0.4, 0.5$ and $p = 0.5$ for instances with $\rho = -0.9$. Dark region represents the better performance of every parameterization.	103
6.8	Comparison between uniform crossover with $p = 0, 0.1, 0.2, 0.3, 0.4, 0.5$ and $p = 0.5$ for instances with $\rho = 0$. Dark region represents the better performance of every parameterization.	104
6.9	Comparison between the crowding-biased uniform crossover (NSGA2-self) and traditional uniform crossover with $p = 0, 0.1, 0.2, 0.3, 0.4$ (NSGA2-p=*) in instances with $\rho = -0.9$. Darker regions represent the better performance of each algorithm.	106
6.10	Comparison between the crowding-biased uniform crossover (NSGA2-self) and traditional uniform crossover with $p = 0, 0.1, 0.2, 0.3, 0.4$ (NSGA2-p=*) in instances with $\rho = 0$. Darker regions represent the better performance of each algorithm.	107

6.11	With $\Psi = 2$ and linkage models $\mathcal{T} = \{\tau_1, \tau_2\}$, these figures show the size distribution of τ_1 along the generations in instances with $\rho = -0.9, -0.5, 0, 0.5, 0.9$ and $K = 2, 4, 8, 12$	108
6.12	Comparison between the linkage-guided crossover (NSGA2-ll-n= Ψ) and traditional uniform crossover with $p = 0.5$ (NSGA2-p=0.5) in instances with $\rho = -0.9$. Darker regions represent the better performance of each algorithm. .	109
6.13	Comparison between the linkage-guided crossover (NSGA2-ll-n= Ψ) and traditional uniform crossover with $p = 0.5$ (NSGA2-p=0.5) in instances with $\rho = 0$. Darker regions represent the better performance of each algorithm.	110
6.14	Reproduction vectors generated by linkage-guided and uniform crossovers, for instances with $\rho = -0.9$. The color of a point represents its distance to the closest parent, darker red implies more distant parents.	111
6.15	Reproduction vectors generated by linkage-guided and uniform crossover, for instances with $\rho = 0$. The color of a point represents its distance to the closest parent, darker red implies more distant parents.	112
6.16	Reproduction vectors generated by linkage-guided with $\Psi = 9, 17, 33$, for instances with $\rho = -0.9$. The color of a point represents its distance to the closest parent; darker red implies more distant parents.	113
6.17	Illustrative example of weight vectors and the subregions they define. The white points are objective vectors whose ideal weight vectors w^* coincide with some $w \in \mathbf{W}$	114
6.18	Crossover bias averaged from 30 runs, each comprised of 2, 500 generations and population size $\mu = 100$. The first row shows the results for n -point crossover with $n = 1, 2, 3, 6, 12$, the second row shows results for p -uniform crossover with $p = 0.01, 0.1, 0.3, 0.4, 0.5$ and the third results for linkage-guided crossover with $\Psi = 2, 9, 17, 65, 257$	115

List of Algorithms

2.1	Canonical genetic algorithm.	12
2.2	Canonical estimation of distribution algorithm.	19
2.3	Optimal mixing recombination (R_{LTGA}).	25
3.1	Linkage-guided crossover.	38
3.2	Deterministic hill-climbing (dHC).	46
5.1	Chu and Beasley genetic algorithm (CBGA).	75
5.2	Repair feasibility	75
6.1	Crowding-biased uniform crossover	105

List of Acronyms

BB Building Block.

BMDA Bivariate Marginal Distribution Algorithm.

BMOA Bayesian Multi-objective Optimization Algorithm.

BOA Bayesian Optimization Algorithm.

CBGA Chu& Beasley Genetic Algorithm.

CGA Compact Genetic Algorithm.

CPF Concatenated Parity Function.

CPTF Concatenated Parity-Trap Function.

dHC deterministic Hill-Climbing.

EA Evolutionary Algorithm.

EAF Empirical Attainment Function.

eCGA Extended Compact Genetic Algorithm.

EDA Estimation of Distribution Algorithm.

EP Evolutionary Programming.

ES Evolutionary Strategy.

FOS Family of Subsets.

GA Genetic Algorithm.

GP Genetic Programming.

hBOA hierarchical Bayesian Optimization Algorithm.

- LTGA** Linkage Tree Genetic Algorithm.
- MA** Memetic Algorithm.
- mhBOA** Multi-objective Hierarchical Bayesian Optimization Algorithm.
- MIDEA** Multi-objective Iterated Density Estimation Algorithm.
- MIMIC** Mutual Information Maximizing Input Clustering.
- MKP** Multidimensional Knapsack Problem.
- MOEA** Multi-objective Evolutionary Algorithm.
- MOEA/D** Multi-objective Evolutionary Algorithm based on Decomposition.
- MOEDA** Multi-objective Estimation of Distribution Algorithm.
- MOGA** Multi-objective Genetic Algorithm.
- MOGLS** Multi-objective Genetic Local Search.
- MONEDA** Multi-objective Neural Estimation of Distribution Algorithm.
- MOP** Multi-objective Optimization Problem.
- NSGA** Non-dominated Sorting Genetic Algorithm.
- NSGA-II** Non-dominated Sorting Genetic Algorithm II.
- PBIL** Population-Based Incremental Learning.
- PGM** Probabilistic Graphical Model.
- PMBGA** Probabilistic Model Building Genetic Algorithm.
- rHC** randomized Hill-Climbing.
- rLTGA** Random Linkage Tree Genetic Algorithm.
- RTS** Restricted Tournament Selection.
- SPEA** Strength Pareto Evolutionary Algorithm.
- SPEA2** Strength Pareto Evolutionary Algorithm 2.
- UMDA** Univariate Marginal Distribution Algorithm.
- UPGMA** Unweighted Pair Grouping Method with Arithmetic-mean.

Chapter 1

Introduction

Nearly-decomposable systems with hierarchical structures were investigated by Simon (1996) and conceptually interpreted by Holland (1975) as the rationale for the performance of simple Genetic Algorithms (GAs). From these ideas the schema theory and the Building Block (BB) hypothesis were proposed, which were further extended by Goldberg (1989, 2002). In a general sense, Holland and Goldberg’s theories rely on reductionist ideas, in which complex optimization problems would be composed of interacting substructures that could increase the efficiency of the search if effectively exploited by GAs.

The schema theory was an attempt to formalize the concept of “substructures” and model GAs’ working principles. According to Holland (1975), the common substructures which compose high-quality solutions are implicitly privileged in the selection step of GAs. However, the randomized reproduction operators employed, e.g. crossover and mutation, may disrupt those substructures and slow-down the evolutionary progress. Therefore, to increase the efficiency of the search, reproduction operators should be implemented in synergy with the selection step, in such a way to avoid frequent disruption of substructures previously selected. In summary, Holland (1975)’s argument is that substructures should be seen as BBs of high-quality solutions and their proper use could enable more effective GAs, such ideas grounded the BB hypothesis.

These seminal studies suggested more effective GAs as hypothetically possible. However, in practice, BBs are unknown entities and simple GAs are not able to exploit them effectively. Only with the use of machine learning techniques for BB identification, GAs inspired by BB hypothesis have become practically tangible. The new class of GAs that emerged has been referred to as Probabilistic Model Building Genetic Algorithms (PMBGAs) (Pelikan et al., 2006a) by some authors and Estimation of Distribution Algorithms (EDAs) (Larraanaga and Lozano, 2001) by others. Their underlying idea regards the use of probabilistic modeling to obtain useful information about the population and avoid frequent BB disruption during reproduction.

Although the first EDAs employed simple univariate models and were not, necessarily, inspired by the BB-hypothesis (Mühlenbein and Schlierkamp-Voosen, 1993), the compatibility between EDAs and BBs was soon noticed. By that time, many EDAs envisaged an effective implementation of ideas originated from the BB hypothesis. As a consequence, models of increas-

ing complexity (univariate, bivariate and multivariate) were successively employed (Larrañaga et al., 2012). One of the first EDAs to use multivariate models, the Extended Compact Genetic Algorithm (eCGA), proposed by Harik (1997), was so strongly influenced by the BB-hypothesis that its model could only represent statistical dependencies among disjoint subsets of decision variables. The eCGA introduced the term *linkage-learning* as denoting the identification of BBs from statistical dependencies. However, due its limitations, EDAs employing more representative models were soon investigated. The Bayesian Optimization Algorithm (BOA), proposed by Pelikan et al. (1999), for example, uses Bayesian networks in which overlapping high-order statistical dependencies can be represented to a certain extent (Pelikan, 2005, 2010).

The perspective of using probabilistic modeling as a tool for BB identification also influenced the development of Multi-objective Estimation of Distribution Algorithms (MOEDAs). Extensions of single-objective EDAs to Multi-objective Optimization Problems (MOPs) were proposed based on previous Multi-objective Evolutionary Algorithms (MOEAs) (Pelikan et al., 2006b). Such straight adaptations of MOEDAs from EDAs, as formalized by (Zydallis, 2003), introduced some controversial ideas to the field. Despite the interesting results reported by Laumanns and Ocenasek (2002) and Shah and Reed (2011), the theoretical analysis performed by Sastry et al. (2005) on simple additively separable bi-objective functions predicted a much worse scenario for the applicability of MOEDAs. In summary, Sastry et al. (2005) conclusions suggested the need for a mixture of models to effectively solve MOPs whereas Laumanns and Ocenasek (2002) and Shah and Reed (2011) reported relevant results from the use of a single model. Martí et al. (2011b) also investigated MOEDAs issues and evidenced the lack of a proper treatment for outliers (isolated objective vectors) and the unnecessary computational cost associated with model-building (Martí et al., 2008b; Martí et al., 2009, 2011a).

In both single- and multi-objective contexts, EDAs have emerged on the assumption that there is a correlation between the model-complexity (representativity) and the efficiency of the resulting EDA. However, theoretical analysis and empirical results have shown that the difficulties in producing accurate linkage models could be an impediment for such a relation to be true (such difficulties were first analyzed in the context of additively separable functions). Coffin and Smith (2007), for example, concluded that BOA takes an exponential number of fitness evaluations to find the optimum of separable Concatenated Parity Functions (CPFs). The authors also reported that linkage-learning based on bivariate statistics cannot find high-order linkages for the CPFs due to an intrinsic *allelic-pairwise independence* between decision variables. On the other hand, Chen and Yu (2009) showed that the Compact Genetic Algorithm (CGA) (Harik et al., 1999) and eCGA (Harik and Goldberg, 2000) can solve both CPFs and Concatenated Parity-Trap Functions (CPTFs) in polynomial time. According to the authors, BOA's model-building was more susceptible to spurious linkages, which deteriorate its effectiveness as a whole. However, Echegoyen et al. (2011) observed that the degree of interactions between decision variables was, in fact, the real source of difficulties for model-building. The authors showed that BOA's capacity for identifying linkages decreases considerably as the number of interactions increases

and concluded that the representativity of a model is not enough to guarantee the effectiveness of EDAs if accurate information cannot be extracted from the population.

Despite the important role of the BB hypothesis for the EDA field, the retrospect shows the relation between BBs and probabilistic modeling has also produced controversial results. The literature on EDAs initially progressed based on the assumption that experiments performed with additively separable deceptive functions would help to understand the performance of EDAs in more complex optimization problems (Goldberg, 2002). However, theoretical and experimental studies concerning the limitations of model-building have questioned the validity of such an assumption. In general, many recent EDAs barely rely on the assumptions defined by Holland (1975), in the sense that (1) current developments are much more related to statistics and model-building than to BBs, as initially envisaged, and (2) the effectiveness of EDAs is much more related to their capacity of extracting useful information from the population than to the BB hypothesis.

1.1 Scope and Objectives

This thesis addresses the analysis of linkage-learning in the context of single- and bi-objective problems, regarding the optimization of pseudo-Boolean functions (Hammer and Peled, 1972). Theoretical results are, in most cases, described in terms of additively separable functions whereas general conclusions are assessed through experimentation with the NK -landscape model (Kauffman, 1993) and the Multidimensional Knapsack Problem (MKP) (Weingartner and Ness, 1967). Although some multivariate probabilistic models are considered, we restrict the analysis to those built upon bivariate statistics, which in fact encompasses most EDAs.

Given the scope described above, the aim of this thesis is to elucidate some of the controversial results on the employment of linkage-learning in evolutionary optimization. Next, the items of concern in this thesis are pinpointed regarding single- and multi-objective optimization problems; the goals to be pursued in each of them are also described.

In *single-objective optimization*, there is a gap between the theoretical results obtained by analysis of benchmark functions and the current practice in EDA development. Although serious linkage-learning limitations have been reported by Coffin and Smith (2007) and Chen and Yu (2009), the origins of such limitations are not evident and it is not clear how often they occur in practical optimization problems. Echegoyen et al. (2011), for example, investigated such limitations from the perspective of the degree of interactions between decision variables, which seems to be a promising direction of study. This thesis assumes that a clear understanding of the main causes underlying linkage-learning limitations might be a bridge linking theoretical and practical knowledge in EDAs. Therefore, our goal in *single-objective optimization* is to investigate such causes, infer the characteristics of optimization problems in which they occur and describe how they affect the accuracy of linkage-learning and, consequently, the efficiency of EDAs.

In *multi-objective optimization*, controversial results still remain in the field. The role of linkage-learning, for example, might have different interpretations. The traditional view (inherited from BB-hypothesis), assumes that linkage models should represent substructures composing high-quality solutions; the studies of Sastry et al. (2005), Zydallis (2003) and Pelikan et al. (2005) fit in this category. From such a perspective, linkage-learning in MOEDAs would suffer from the same limitations of EDAs. However, the existence of effective MOEDAs for some NP-hard MOPs have been reported in the literature (see Shah and Reed (2011)) and contradicts such assumption. Therefore, there must be a difference between the information modeled by linkage-learning in single- and multi-objective problems. This thesis investigates linkage-learning in *multi-objective optimization* from the perspective of the correlation among the objective functions and describe how it impacts the effectiveness of a MOEDA. This perspective would justify why EDAs might perform poorly in single-objective MKP (Martins et al., 2014) whereas MOEDAs still provide relevant results in multi-objective MKPs (Shah and Reed, 2011).

1.2 Thesis Contributions

This section overviews the contributions produced in the context of this thesis. They are not described in chronological but instead in a semantical order which relates to the chapters. The publications originated in each of the discussed topics are indicated as footnotes.

In *single-objective optimization*, our first contributions concern the verification of the necessary conditions for accurate learning. Such conditions were formalized by using information theoretical measures and assuming simple additively separable functions as a study case. The analysis evidenced multimodality as the main source of limitations for linkage-learning and showed that *allelic-pairwise independence* might occur as a result of massive multimodality.^{1,2}

Such results predict that multimodality imposes limitations for the usefulness of linkage-learning. However, the degree of multimodality of complex optimization problems is not available and it is difficult to know beforehand if a certain problem will benefit from the use of linkage-learning. To estimate the limits of multimodality tolerated by linkage-learning methods, the NK -landscapes model was employed (Kauffman, 1993). By varying the parameter K , instances with different degrees of multimodality were generated and experiments were performed. The results confirmed the theoretical conclusions, i.e. linkage-learning provided benefits only for instances within the $1 \leq K \leq 6$ range, approximately. Such conclusions helped to explain why EDAs might be effective/efficient in additively separable functions (low multimodality) and still unsuccessful when solving general NP-hard problems (high multimodality).³

¹ Martins, J. P. and Delbem, A. C. B. (2014), Multimodality and the Linkage-learning Difficulty of Additively Separable Functions, in: *Genetic and Evolutionary Computation Conference (GECCO'14)*, pp. 365–372

² Crocorno, M. K. et al. (2012), Decomposition of Black-box Optimization Problems by Community Detection in Bayesian Networks, *International Journal of Natural Computing Research*, **3** (4): 1–19

³ Martins, J. P. and Delbem, A. C. B. (2015a), Assessment of Linkage learning Limits in Highly Multimodal NK -landscapes, *Swarm and Evolutionary Computation (IN REVIEW)*,

Although such results do not evidence what one should expect of linkage-learning when solving NP-hard optimization problems, they make it clear that the limitations of linkage-learning are more restrictive than previously thought. To verify such results in a broader context, experiments were performed with many instances of the NP-hard MKP. The linkage models produced were analyzed and compared to random linkage models and it was observed that most linkages identified were not accurate representations of the underlying problem, i.e. they did not provide reliable information about the population and, as consequence, did not benefit the search progress. This result suggests the MKP is outside the range of usefulness of linkage-learning.^{4,5,6}

Since multimodality is a concern for linkage-learning in *single-objective optimization*, we may expect an even worse situation in *multi-objective optimization*. MOPs are inherently multimodal and even simple conflicting objective functions can produce massive multimodality if combined in a MOP.⁷ Despite this fact, linkage-learning has also been considered as a method for BB identification in the multi-objective context (see Zydallis (2003)). In an opposite direction, we argue that in the multi-objective context the statistics describing the correlation between objective functions are more evident and easier to model than consensual substructures relating the objectives. This is the perspective used in this thesis to study linkage-learning in MOPs.

To understand how linkage models can help reproduction in multi-objective optimization, we defined the effectiveness of crossover operators in relation to the idealistic concept of *uniformity-preserving multi-objective crossover*. We say that a crossover is an effective reproduction operator for MOPs if the offspring population it produces is uniformly distributed along the current Pareto-front. We show, theoretically and experimentally, that traditional crossover operators, like *n*-point and *p*-uniform, have an intrinsic bias towards the “middle region” of the objective space that hampers the efficacy of diversity preservation mechanisms and makes such operators ineffective.⁸ A linkage-guided crossover, on the other hand, was shown to produce only a relatively small bias and, consequently, more well-spread Pareto-fronts were obtained. The reproductive bias of crossover is an useful concept to explain the behavior of many different approaches used to improve diversity in MOEAs, such as, for example, *restricted matings*, *niching* and *decomposition-based reproduction*. Furthermore it leads to new insights about how to implement effective reproduction operators for multi-objective evolutionary optimization.⁹

⁴ Martins, J. P. et al. (2014), On the performance of linkage-tree genetic algorithms for the multidimensional knapsack problem, *Neurocomputing*, **146** (0): 17–29

⁵ Martins, J. P. et al. (2013), A Comparison of Linkage-learning-based Genetic Algorithms in Multidimensional knapsack Problems, in: *Conference on Evolutionary Computation (CEC'13)*, pp. 502–509

⁶ Martins, J. P. and Delbem, A. C. B. (2013), The influence of linkage-learning in the linkage-tree GA when solving multidimensional knapsack problems, in: *Genetic and Evolutionary Computation Conference (GECCO'13)*, pp. 821–828

⁷ Martins, J. P. et al. (2011), Multi-objective Phylogenetic Algorithm: Solving Multi-objective Decomposable Deceptive Problems, in: *Evolutionary Multi-Criterion Optimization (EMO'11)*, vol. 6576, pp. 285–297

⁸ Martins, J. P. and Delbem, A. C. B. (2015c), Linkage learning and Crossover bias in Bi-objective pMNK-landscapes, *Swarm and Evolutionary Computation*,

⁹ Martins, J. P. and Delbem, A. C. B. (2015b), Handling Crossover Bias to Improve Diversity in Multiobjective Evolutionary Optimization, in: *Genetic and Evolutionary Computation Conference (GECCO'15)*

1.3 Thesis Outline

This section outlines the subjects discussed in each of the chapters composing this thesis.

Chapter 2 (Evolutionary optimization) Reviews general aspects of the optimization field and the more specific details regarding the use of evolutionary algorithms as general problem solvers. Special focus is given to the simple GA (Holland, 1975) and the use of EDAs in single- and multi-objective optimization.

Chapter 3 (Fundamentals of linkage-learning) Introduces some important notions on the use of linkage-learning to guide crossover and defines the necessary conditions for accurate linkage-learning. Theoretical results are discussed in terms of well-known additively separable functions.

Chapter 4 (Multimodality and the limits of linkage-learning) Investigates how multimodality affects the effectiveness of the Linkage Tree Genetic Algorithm (LTGA) in solving instances of the NK -landscapes model. The results confirm the existence of a range for K in which the LTGA is effective.

Chapter 5 (Linkage-learning and the Multidimensional Knapsack Problem) Evaluates the efficacy of linkage-learning when solving instances of the MKP. The LTGA was compared with its randomized version, the Random Linkage Tree Genetic Algorithm (rLTGA), and the linkage models were analyzed. From the results, we conclude that the MKP is usually outside the range of effectiveness of linkage-learning. Indeed, no significant benefits were obtained by using the LTGA instead of the rLTGA.

Chapter 6 (Reproductive bias and Linkage-learning in MOPs) Evaluates linkage-learning in multi-objective optimization from the perspective of *reproductive bias*. At first, it is shown that linkage-learning models the correlation between objectives. Next, a linkage-guided crossover is compared with traditional crossover operators, which is shown to possess a much weaker bias towards the middle region of the Pareto-optimal front (bi-objective NK -landscapes instances were used for the experiments). The implications of reproductive bias are also discussed in a broader perspective.

Chapter 7 (Conclusions): Summarizes the results for single- and multi-objective evolutionary optimization and discusses future directions of research.

Chapter 2

Evolutionary Optimization

Evolutionary optimization is an area of research concerned with the study and development of algorithms to solve optimization problems. In a broad perspective, Evolutionary Algorithms (EAs) are metaheuristics in which the search mechanisms mimic the behavior of natural evolutionary processes. This chapter reviews and defines fundamental ideas underlying EAs in the context of single- and multi-objective optimization.

2.1 Introduction

This section introduces some basic concepts from the optimization field regarding single- and multi-objective optimization. Furthermore, it describes an unified model of evolution that is adequate to describe methods inspired by the natural evolutionary process.

2.1.1 Optimization Problems

A general optimization problem is defined by a feasible set $\mathcal{G}$, an objective set $\mathcal{P}$ and a mapping $\varphi : \mathcal{G} \mapsto \mathcal{P}$. The feasible set $\mathcal{G}$ is a subset $\mathcal{G} \subseteq \mathbb{R}^N$ (the decision space), whereas the objective set is the codomain of φ , $\mathcal{P} \subseteq \mathbb{R}^M$ (the objective space). Additionally, $f : \mathcal{P} \mapsto \mathcal{F}$ is defined as a function which maps objective vectors in $\mathcal{P}$ to a partially ordered set $\mathcal{F}$. Without loss of generality, the solution of an optimization problem consists in finding feasible solutions $\mathbf{x} \in \mathcal{G}$ which maximizes $f(\varphi(\mathbf{x}))$.

$$\max \quad f(\varphi(\mathbf{x})), \tag{2.1}$$

$$\text{subject to} \quad \mathbf{x} \in \mathcal{G}. \tag{2.2}$$

A solution $\mathbf{x}^* \in \mathcal{G}$ is said to be optimal if it produces a maximal $f(\varphi(\mathbf{x}^*))$, which is equivalent to say there is no “better” solutions than $\mathbf{x}^*$ in $\mathcal{G}$. Since the comparison of solutions is performed in $\mathcal{F}$, the function f must be defined in such a way to enable a partial ordering for $\mathcal{F}$. In single-objective optimization, solutions $\mathbf{x} \in \mathcal{G}$ are associated with unidimensional

objective vectors, i.e. $\varphi(\mathbf{x}) \in \mathbb{R}$. Since $\mathbb{R}$ is a *totally ordered* set under the binary relation ' $\geq$ ', the $f(\varphi(\mathbf{x})) = \varphi(\mathbf{x})$ mapping can be applied. However, this is not always appropriate, as discussed in posterior sections. A *totally ordered* set is a special case of *partially ordered* set, for which there are binary relations with the following properties.

Definition 2.1 (Partial order). It is a binary relation R among elements $\forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{P}$, that guarantees: *i.*(reflexivity) $\mathbf{u}R\mathbf{u}$; *ii.*(antisymmetry) if $\mathbf{u}R\mathbf{v}$ and $\mathbf{v}R\mathbf{u}$ then $\mathbf{u} = \mathbf{v}$ and *iii.*(transitivity) if $\mathbf{u}R\mathbf{v}$ and $\mathbf{v}R\mathbf{w}$ then $\mathbf{u}R\mathbf{w}$.

Definition 2.2 (Total order). It is a binary relation R between elements $\forall \mathbf{u}, \mathbf{v} \in \mathcal{P}$, that induces a partial order and guarantees: (totality) $\mathbf{u}R\mathbf{v}$ or $\mathbf{v}R\mathbf{u}$.

In Multi-objective Optimization Problems (MOPs) ($M > 1$) solutions are associated with multidimensional objective vectors $\varphi(\mathbf{x}) \in \mathcal{P} \subseteq \mathbb{R}^M$ and the binary relation ' $\geq$ ' does not induce a *total ordering* in $\mathcal{P}$. Therefore, specific binary relations must be adopted to define f . Usual alternatives are based on the *Pareto-dominance* binary relation (Definition 2.3).

Definition 2.3 (Pareto-dominance, $>_P$). It is a binary relation $>_P$ between elements $\mathbf{u}, \mathbf{v} \in \mathcal{P}$, such: $\mathbf{u} >_P \mathbf{v}$ if and only if $u_i \geq v_i, \forall i \in [M] \wedge \exists i \in [M] : u_i > v_i$; with $[M] = \{1, \dots, M\}$.

However, since ' $>_P$ ' is not reflexive, Pareto-dominance does not induces a strict *partial order* in $\mathcal{P}$ and the *weak Pareto-dominance* (Definition 2.4) is usually preferred.

Definition 2.4 (Weak Pareto-dominance, $\geq_P$). It is a binary relation $\geq_P$ between elements $\mathbf{u}, \mathbf{v} \in \mathcal{P}$, such: $\mathbf{u} \geq_P \mathbf{v}$ if and only if $u_i \geq v_i, \forall i \in [M]$.

By using ' $\geq_P$ ', a function f can be defined and a partial order induced in $\mathcal{P}$ producing $\mathcal{F}$. However, ' $\geq_P$ ' is not the only possible binary relation, therefore, from a theoretical point-of-view, multi-objective optimization is better described in terms of a more general preorder relation ' $\succeq$ '.

Definition 2.5 (Preorder, $\succeq$). It is a binary relation $\succeq$ between elements $\forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{P}$, such that: *i.*(reflexivity) $\mathbf{u} \succeq \mathbf{u}$; *ii.*(transitivity) if $\mathbf{u} \succeq \mathbf{v}$ and $\mathbf{v} \succeq \mathbf{w}$ then $\mathbf{u} \succeq \mathbf{w}$.

Note, *partial orders* and *total orders* are also *preorders*. Through the use of ' $\succeq$ ' and a function $f : \mathcal{P} \mapsto \mathcal{F}$, the notion of “better solutions” can be formalized and a general definition of global-optima solutions (Definition 2.6) in single- and multi-objective optimization is enabled. For a simple notation, assume $\mathbf{z}_u = \varphi(\mathbf{u}), \forall \mathbf{u} \in \mathcal{G}$.

Definition 2.6 (Global optimum). A solution $\mathbf{x} \in \mathcal{G}$ is a global optimum if it is not worst than any other solution in $\mathcal{G}$, i.e. $\nexists \mathbf{y} \in \mathcal{G} : f(\mathbf{z}_y) \succeq f(\mathbf{z}_x)$ and $f(\mathbf{z}_y) \neq f(\mathbf{z}_x)$.

Since ' $\succeq$ ' is reflexive in $\mathcal{F}$, optimization problems might have multiple global optima. In single-objective optimization, global optima solutions can differ only in $\mathcal{G}$, i.e. they must have the same objective value in $\mathcal{P}$. In multi-objective optimization, on the other hand, global optima

solutions may differ in the decision and objective spaces, but must be equivalent in $\mathcal{F}$. Indeed, in an idealistic scenario, the solution of a MOP consists in finding a set of globally optimal solutions with diversified characteristics in both the decision and objective spaces.

To summarize the concepts described in this section, let us assume a bi-objective problem ($M = 2$). Figure 2.1 illustrates solutions $\{x^*, x\} \in \mathcal{G}$ and their respective objective vectors $\{z^*, z\} \in \mathcal{P}$. The function f maps the objective vectors z^* and z to a preordered set $\mathcal{F}$, where $f(z^*) \succeq f(z)$. Since $f(z^*)$ is maximal in $\mathcal{F}$, x^* is a global-optimum. In the specific case of the weak Pareto-dominance relation ' $\succeq_P$ ', solutions like z^* are called *Pareto-optimal* and compose the *Pareto-optimal set*. The image of the Pareto-optimal set according to $\varphi(\cdot)$ composes the *Pareto-optimal front*.

Figure 2.1: Pareto-optimal solution x^* , a dominated solution x , their respective objective vectors z^* , z and their relative ordering in $\mathcal{F}$.

This section reviewed some basic concepts regarding single- and multi-objective optimization. Section 2.1.2 uses these concepts to overview the general principles of evolutionary algorithms and describe how they proceed in the search for global-optima solutions.

2.1.2 Unified Model of Evolutionary Algorithms

In the 1930s researchers already found it useful to visualize real evolutionary systems as a set of agents exploring multi-peaked fitness landscapes (Wright, 1932). However, it was only in the 1960s, with cheaper computational resources available, that evolutionary computation emerged as an automated alternative for problem solving. Early approaches, inspired by evolutionary systems, were proposed almost simultaneously by at least three different research groups in the context of Evolutionary Strategies (ESs) (Rechenberg, 1965), Evolutionary Programming (EP) (Fogel et al., 1966) and Genetic Algorithms (GAs) (Holland, 1975). Along the years many branches of these EAs have become mature fields of research, as for example, Memetic Algorithms (MAs) (Moscato, 1989), Genetic Programming (GP) (Koza, 1992) and Estimation of Distribution Algorithms (EDAs) (Mühlenbein and Paass, 1996).

Although such diversification was fruitful in expanding EAs' scope of application, it also imposed certain difficulties for a better understanding of their working principles. The different branches of evolutionary optimization usually take inconsistent perspectives and focus on different algorithmic aspects, which limits the synergy among them. Some theoretical models have been proposed in an attempt to unify the understanding of EAs in basic elements, which could better illustrate the similarities between the fields. As examples of such models we may cite the topological interpretation proposed by Moraglio and Poli (2004), the black-box framework proposed by Droste et al. (2006) and the unified model of evolution described by Paixão et al. (2014) in the context of the SAGE project.¹ This section follows, to certain extent, the model described by Paixão et al. (2014) to characterize EAs.

In a general sense, EAs are search methods whose resulting dynamics mimic evolutionary processes found in nature. More specifically, EAs define stochastic operators which modify a population P and along a certain number of *generations* produce a sequence

$$(P(1), P(2), \dots, P(t), \dots). \quad (2.3)$$

At every step $t > 0$, $P(t) \in \mathcal{G}^\mu$ is a multiset² composed of μ (population size) *individuals* $x \in \mathcal{G}$ ($\mathcal{G}$ is called the *genotype space*). Through the application of stochastic operators $\psi : \mathcal{G}^\mu \mapsto \mathcal{G}^\mu$, the population $P(t)$ is transformed into a population $P(t+1)$.

$$P(t+1) = \psi(P(t)). \quad (2.4)$$

In practical EAs, ψ is usually composed of multiple operators, which mimic the behavior of *reproduction* and *natural selection*. Such operators might produce a different number of individuals than the population size, denoted as $\lambda > 0$. According to the notation proposed by Paixão et al. (2014), the composition of two operators $\mathbb{O}_1 : \mathcal{G}^\mu \mapsto \mathcal{G}^\lambda$ and $\mathbb{O}_2 : \mathcal{G}^\lambda \mapsto \mathcal{G}^\mu$ is denoted as $(\mathbb{O}_2 \circ \mathbb{O}_1) : \mathcal{G}^\mu \mapsto \mathcal{G}^\mu \equiv \mathbb{O}_2(\mathbb{O}_1(P(t)))$. Observe that these stochastic operators work on the genotype space $\mathcal{G}$ by modifying the individuals in the current population $P(t)$.

If instead of modeling the evolutionary process through the use of concrete populations the probability distributions $D(t) \in \mathcal{D}$ underlying the populations are used, a different model is conceived, where $\mathcal{D}$ is the space of possible distributions.

$$(D(1), D(2), \dots, D(t), \dots). \quad (2.5)$$

Similarly to the case of concrete populations, there must be operators $\phi : \mathcal{D} \mapsto \mathcal{D}$, from which the current probability distribution $D(t)$ can be transformed in $D(t+1)$,

$$D(t+1) = \phi(D(t)). \quad (2.6)$$

¹Speed of Adaptation in Population Genetics and Evolutionary Computation (www.project-sage.eu).

²A multiset is a generalization of the concept of *set* that allows multiple instances of the same element.

EAs based on probability distributions are usually called EDAs. In most of them the implementation of ϕ requires the sampling of the distributions. The sampling operator $\beta_\lambda : \mathcal{D} \mapsto \mathcal{G}^\lambda$ produces λ individuals according to the probability distribution D and it is defined as

$$\beta_\lambda(D) := (X_i)_{i \in [\lambda]}, \quad \text{where } X_1, \dots, X_\lambda \sim D. \quad (2.7)$$

The distribution $D(t+1)$ can be estimated from a population $P(t)$ by an operator

$$\alpha : \mathcal{G}^\lambda \mapsto \mathcal{D}. \quad (2.8)$$

Variation and sampling operators (ψ, β) are classified as *reproduction operators* whereas α denote the probabilistic modeling or *linkage learning* step in EDAs.

So far we have discussed, in abstract terms, that EAs and EDAs work by modifying a population $P(t)$ and/or a distribution $D(t)$ along the generations. All the abstract operators we described act on the genotype space $\mathcal{G}$, in resemblance to genetic variation which occurs during the reproduction of living organisms. In order to allow evolution to emerge, we need to define operators acting on the *phenotype space* $\mathcal{P} \subseteq \mathbb{R}^M$ which mimic *natural selection*.

Survival-of-the-fittest is the idea underlying natural selection, in which the fittest individuals (according to their phenotype) are those who produce more offspring than the others. In evolutionary optimization, the individuals represent candidate solutions for a certain optimization problem. In other words, φ performs the genotype-phenotype mapping which provides the features of each individual $\mathbf{x} \in \mathcal{G}$. From $\varphi(\mathbf{x})$, a fitness $f(\varphi(\mathbf{x}))$ can be attributed to the individuals $\mathbf{x} \in \mathcal{G}$ (*fitness assignment*) and selection operators $S_f : \mathcal{G}^\mu \mapsto \mathcal{G}^\lambda$ can be implemented.

The unified model of EA reviewed in this section can be instantiated to describe most of the existing EAs variations and provides a concise way to generalize the basic elements of evolutionary computation. Sections 2.2-2.3 use this model to describe GAs and EDAs for single-objective optimization. Section 2.4 discusses how the components of EAs can be extended to multi-objective optimization and provides examples of successful approaches.

2.2 Genetic Algorithms

Genetic Algorithms (GAs) constitute one of the main branches of EAs. Proposed by Holland (1975) as a systematic way to model adaptive agent systems, GAs were the first EAs to rely on crossover as the main reproduction operator (Mitchell, 1998). Since the studies of De Jong (1975), many subsequent works analyzed GAs from the perspective of problem solving (Goldberg, 1989). In this context, *crossover* and *mutation* implement the search operators and *selection* guides the search towards better solutions. Such a metaphor is widely accepted in the literature and summarizes the main elements of GAs.

The development of GAs for a specific optimization problem encompasses the definition of

four elements: (1) a solution encoding, (2) a selection operator $S : \mathcal{G}^\mu \mapsto \mathcal{G}^\lambda$, (3) a recombination operator $R : \mathcal{G}^\lambda \mapsto \mathcal{G}^\mu$ and (4) a mutation operator $M : \mathcal{G}^\mu \mapsto \mathcal{G}^\mu$. Therefore one step of the canonical GA can be described as

$$P(t+1) = M \circ R \circ S(P(t)). \quad (2.9)$$

In words, λ solutions are selected from $P(t)$ by the selection operator S , such solutions are used by the recombination operator R to produce other μ candidate solutions, all of which are slightly modified by the mutation operator M and results in the population $P(t+1)$. Note that λ might be equal to the population size μ . By iterating Equation (2.9), Algorithm 2.1 is defined. It starts with an initial population $P(t=0)$, from which λ solutions are selected to compose P_{selected} , which undergoes crossover and mutation and produces the next population $P(t+1) = P_{\text{offspring}}$.

Algorithm 2.1 Canonical genetic algorithm.

Require: An initial population $P(0)$, selection, recombination and mutation operators S, R, M .

1:	while $t < \text{MAX_STEPS}$ do	▷ Repeat until the maximum number of steps.
2:	$P_{\text{selected}} \leftarrow S_f(P(t))$	▷ Select the fittest solutions from P .
3:	$P_{\text{offspring}} \leftarrow R(P_{\text{selected}})$	▷ Generates offspring from parents.
4:	$P_{\text{offspring}} \leftarrow M(P_{\text{offspring}})$	▷ Slightly modifies part of the offspring.
5:	$P(t+1) \leftarrow P_{\text{offspring}}$	▷ $P_{\text{offspring}}$ becomes $P(t+1)$.
-

The scope of this thesis is restricted to binary genotype spaces $\mathcal{G} \in \{0, 1\}^N$ and the remaining of this section describes some commonly used operators in this context. A thorough discussion about other solution encodings and their operators is behind our purposes and the interested reader is referred to Rothlauf (2006).

The next sections describe the operators proposed by Holland (1975) and some alternatives to them. Section 2.2.1 exemplifies *fitness assignment* procedures to define $f : \mathcal{P} \mapsto \mathcal{F}$. Section 2.2.2 describes some *selection operators*. Section 2.2.3 concerns traditional *variation operators* for binary encodings. Section 2.2.4 briefly reviews some of the working principles of GAs in terms of the schema theory and the Building Block (BB) hypothesis.

2.2.1 Fitness Assignment

Fitness assignment is the mapping $f : \mathcal{P} \mapsto \mathcal{F}$, which associates values to phenotype features and produces a preordered set $\mathcal{F}$ in which individuals $\mathbf{x} \in \mathcal{G}$ can be qualitatively compared. The simplest fitness assignment, in single-objective optimization, consists of defining $f(\varphi(\mathbf{x})) = \varphi(\mathbf{x})$. However, this approach is not always appropriate, as shown in the remaining of this section.

As mentioned in Section 2.1.1, optimization problems may have globally optimal solutions in multiple regions of the genotype space, a feature denoted as *multimodality*. In some cases, it would be useful if GAs could find many of such global-optima at once. Unfortunately, there are

mechanisms in the simple GA that hamper such an ability. At every step t of a GA, the whole population is evaluated and the best individuals overall are privileged during reproduction. This characteristic changes the population dynamics by increasing the density of individuals in the regions of higher fitness, but at the same time decreases the density of solutions in regions with fewer individuals and/or lower fitness. Along the generations, many promising regions may become uninhabited, which hinders the exploration of multiple regions at the same time.

Fitness assignment methods have been proposed to enable a “more” uniform exploration of multiple regions of interest in the genotype space. One of the first successful approaches was proposed by Holland (1975) and it is based on the idea of evolving multiple *niches* in parallel (Goldberg and Richardson, 1987). Such a technique is called the *sharing method* and works by decreasing the fitness of individuals located in highly populated regions. In other words, individuals belonging to the same *niche* share their fitness, therefore, individuals in large niches tend to receive a small shared fitness (Sareni and Krahenbuhl, 1998). The shared fitness of an individual $\mathbf{x}_i \in \mathcal{G}$ is given by

$$f'(\mathbf{x}_i) = \frac{\varphi(\mathbf{x}_i)}{m(\mathbf{x}_i)}, \quad \forall \mathbf{x}_i \in P(t) \quad (2.10)$$

where $m(\mathbf{x}_i)$ accounts for the number of individuals with $\mathbf{x}_i$ share its fitness. The computation of $m(\mathbf{x}_i)$ takes in account the distance of $\mathbf{x}_i$ to every other solution $\mathbf{x}_j \in P(t)$

$$m(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in P(t)} sh(d(\mathbf{x}_i, \mathbf{x}_j)), \quad (2.11)$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ measures the similarity between candidate solutions $\mathbf{x}_i, \mathbf{x}_j$ and $sh(d(\mathbf{x}_i, \mathbf{x}_j))$ returns 1 if they are identical, 0 if their distance is greater than the threshold σ_{share} , and otherwise an intermediate value is returned. The best value for ζ is problem-specific, but it is usually set to $\zeta = 1$.

$$sh(d(\mathbf{x}_i, \mathbf{x}_j)) = \begin{cases} 1 - (d(\mathbf{x}_i, \mathbf{x}_j)/\sigma_{\text{share}})^\zeta, & \text{if } d(\mathbf{x}_i, \mathbf{x}_j) < \sigma_{\text{share}} \\ 0, & \text{otherwise} \end{cases} \quad (2.12)$$

The distance $d(\mathbf{x}_i, \mathbf{x}_j)$ can be computed in the genotype or phenotype. In the case of a binary encoding, the Hamming distance is an appropriate choice, whereas in the phenotype, the Euclidean distance may be a better option (Cioppa et al., 2004; Mahfoud, 1995; Yin and Gernay, 1993).

Many other niching techniques were proposed over the years. Miller and Shaw (1996), for example, extended fitness sharing by associating individuals to one (or none) of q dynamically identified niches. Another variation, proposed by Miller and Shaw (1996), called “clearing”, shares fitness as resources among a certain number of individuals in each niche, while the remaining receive zero fitness (Petrowski, 1996). Clustering is also a common way of imple-

menting niching. First, solutions are clustered in q different niches and then the distance between individuals and the centroid of their cluster is used to determine the fitness (Yin and Gernay, 1993). Restricted Tournament Selection (RTS), proposed by Harik (1995), avoids the competition between distant individuals by using an implicitly defined threshold distance. A more complete and updated review of these and other fitness assignment techniques is provided by Shir (2012).

2.2.2 Selection Operators

According to the theory of evolution, fittest individuals reproduce with a higher probability of success. In context of GAs, this is implemented by selection operators, which compare the individuals and give to the fittest more chance to undergo crossover and mutation. Usually, selection operators S_f (or simply S) copy the fittest individuals from $P(t)$ to a new population, in such a way that the fittest individuals in $P(t)$ are more likely to be copied than the others. The number of individuals selected λ might be different of the population size μ .

$$S : \mathcal{G}^\mu \mapsto \mathcal{G}^\lambda \quad (2.13)$$

This section describes concrete examples of selection operators and provides references to other studies. The notation used is inspired in Paixão et al. (2014), where $\Pr(\mathbf{x}_j)$ denotes the probability of an individual $\mathbf{x}_j$ being chosen by the selection operator.

Uniform selection. This operator selects individuals independently of their fitness, and it is useful to define more elaborate operators.

$$S_{\text{Unif}}(P(t)), \quad \text{in which } \Pr(\mathbf{x}_j) = 1/\mu, \forall \mathbf{x}_j \in P(t). \quad (2.14)$$

Fitness proportional Selection. This operator, proposed by Holland (1975), selects individuals with a probability proportional to their fitness in relation to the total fitness of $P(t)$.

$$S_{\text{Prop}(f)}(P(t)), \quad \text{in which } \Pr(\mathbf{x}_i) = \frac{f(\mathbf{x}_i)}{\sum_{\mathbf{x}_j \in P(t)} f(\mathbf{x}_j)}, \forall \mathbf{x}_i \in P(t). \quad (2.15)$$

The main criticism about proportional selection concerns the beginning and the ending stage of the search. In the beginning, the best solutions might be selected with a much higher probability (high selective pressure), taking over the population and favoring premature convergence. On the other hand, in further stages of the search, the difference between the higher and average fitness might be too small to favor the best solutions, leading to genetic drift due to the low selective pressure (Fogel, 1999).

Tournament Selection. This operator does not use the fitness itself but the relative ranking of solutions according to their fitness, which avoids the drawbacks produced by *fitness proportional*

selection. In words, $q \geq 2$ solutions are chosen uniformly at random $P_i = S_{\text{Unif}(q)}(P(t))$. Then, among those solutions the one with better fitness, say $\mathbf{x}_i$, is called the tournament winner and selected. Such procedure is repeated until all λ solutions have been selected (Goldberg, 2002).

$$S_{\text{Tour}(f,q)}(P(t)), \quad \text{with } \mathbf{x}_i = \arg \max_{\mathbf{x} \in P_i} f(\mathbf{x}), \quad \forall i \in [\lambda] \quad (2.16)$$

where $P_i = S_{\text{Unif}(q)}(P(t))$.

Since only tournament winners are allowed to reproduce, selection pressure can be increased proportionally to q . A small q favors exploration, whereas a large q favors fast convergence.

Truncation Selection. Proposed by Mühlenbein and Schlierkamp-Voosen (1993), the truncation selection operator also uses the ranking of solutions instead of their fitness. The population $P(t)$ is sorted in decreasing order of fitness

$$S_{\text{Sort}(f)}(P(t)) = (\mathbf{x}_{r(1)}, \dots, \mathbf{x}_{r(\mu)}), \quad \text{s.t. } f(\mathbf{x}_{r(1)}) \succeq \dots \succeq f(\mathbf{x}_{r(\mu)}). \quad (2.17)$$

Where $r : i \mapsto j$ is the permutation in which $r(i)$ is the individual with the i^{th} better fitness. Next, the population is trimmed and only the first $T\mu$ solutions are kept in $P(t)$, with $T \in [0, 1]$.

$$S_{\text{Trim}(f,T)} \circ S_{\text{Sort}(f)}(P(t)) = (\mathbf{x}_{r(1)}, \dots, \mathbf{x}_{r(T\mu)}) \quad (2.18)$$

From the resulting population, λ solutions are selected uniformly at random.

$$S_{\text{Trunc}(f,T)}(P(t)) = S_{\text{Unif}(\lambda)} \circ S_{\text{Trim}(f,T)} \circ S_{\text{Sort}(f)}(P(t)) \quad (2.19)$$

A small $T > 0$ would produce a high selective pressure, which decreases as $T \approx 1$. Although being one of the most simple selection operators, there are cases in which it is the preferable choice (Lima et al., 2007).

Many other selection schemes have been proposed and compared along the years. Back (1994) analyzed theoretically *proportional selection*, *tournament selection* dentre outros. Additionally, Blickle and Thiele (1996) compared their reproduction rate, loss of diversity, selection intensity and selection variance, including *proportional selection* and *truncation selection*. A thorough discussion of different aspects concerning selection operators in GAs is provided by Hingee and Hutter (2008), Lehre and Yao (2012), Miller and Goldberg (1996), Sokolov and Whitley (2005), and Xie et al. (2007).

2.2.3 Variation Operators

Variation operators are responsible for modifying the current population $P(t)$ and provide the diversity on which selection operators act. Paixão et al. (2014) identified two main classes of

variation operators in EAs: (1) mutation and (2) recombination. Mutation operators, $M : \mathcal{G} \mapsto \mathcal{G}$, work on individuals by introducing random modifications in their genotype, thereby creating diversity. Recombination operators, $R : \mathcal{G}^\ell \mapsto \mathcal{G}$, on the other hand, recombine the genotype of $\ell \geq 2$ solutions (parents) to produce an individual (offspring) but do not change the frequency of alleles. In the context of black-box optimization, in which *a priori* knowledge about the optimization problem is not available (Droste et al., 2006), Paixão et al. (2014) suggested that variation operators should be *uniformity-preserving*.

Definition 2.7 (Uniformity-preserving operators). A variation operator V is uniformity-preserving if, given a population $P \sim \text{Unif}(\mathcal{G}^{|P|})$, it produces a population $P' = V(P)$ such that $P' \sim \text{Unif}(\mathcal{G}^{|P'|})$.

In other words, *uniformity-preserving* operators do not have any inherent bias towards particular regions of the genotypic space. All operators described in this section are uniformity-preserving.

Uniform mutation

Uniform mutation is a random operator $M_p : \mathcal{G} \mapsto \mathcal{G}$, which acts on an individual $\mathbf{x} = (x_1, \dots, x_N)$ and produces an individual

$$M_p(\mathbf{x}) = (Y_1, \dots, Y_N), \quad \text{s.t } \Pr(Y_i \neq x_i) = p, \quad \forall i \in [N] \quad (2.20)$$

M_p can also operate in the context of whole populations, $M_p : \mathcal{G}^\mu \mapsto \mathcal{G}^\mu$, where $M_p(P(t)) = (M_p(\mathbf{x}_1), \dots, M_p(\mathbf{x}_\mu))$. With $p = 1/N$, the operator $M_{1/N}$ is usually called *standard mutation*.

Recombination operators

Recombination is the main operator in GAs. Holland (1975) and Goldberg (1989) considered 1-point crossover, but n -point ($n > 1$) and uniform crossover have also been extensively investigated (De Jong and Spears, 1992; Eshelman, 1991; Jansen and Wegener, 2005; Moraglio and Poli, 2004; Sastry and Goldberg, 2004; Sudholt, 2005; Watson and Jansen, 2007)

One-point crossover. This operator works on a pair of parents $\mathbf{x}, \mathbf{y} \in \mathcal{G}$ and produces a new individual $\mathbf{z} \in \mathcal{G}$, i.e. $R_{1\text{-point}} : \mathcal{G} \times \mathcal{G} \mapsto \mathcal{G}$. A random crossover point $j \in \{1, \dots, N\}$ is chosen and the offspring $\mathbf{z}$ is composed by the first segment of $\mathbf{x}$ and the second segment of $\mathbf{y}$.

$$z_i = \begin{cases} x_i, & \text{if } i \leq j \\ y_i, & \text{otherwise.} \end{cases} \quad (2.21)$$

n -point crossover. $R_{n\text{-point}} : \mathcal{G} \times \mathcal{G} \mapsto \mathcal{G}$ generalizes $R_{1\text{-point}}$ with n random crossover points $j \in \{1, \dots, N\}$, chosen without replacement. Assume an ordered sequence $J =$

$(j_0, j_1, \dots, j_n, j_{n+1})$, with $j_0 = 0$ and $j_{n+1} = N$. The offspring z is created as follows, with $\forall a \in \{0, \dots, n\}$ and $b = a + 1$.

$$z_i = \begin{cases} x_i, & \text{if } (j_a < i \leq j_b) \text{ and } (b \bmod 2 \neq 0) \\ y_i, & \text{if } (j_a < i \leq j_b) \text{ and } (b \bmod 2 = 0). \end{cases} \quad (2.22)$$

Uniform crossover. In this operator, $R_{\text{Unif}} : \mathcal{G} \times \mathcal{G} \mapsto \mathcal{G}$, there are no crossover points, instead, each allele is copied from one of the parent according to a probability p . In the traditional uniform crossover $p = 0.5$.

$$\Pr(Z_i = x_i) = p \quad (2.23)$$

$$\Pr(Z_i = y_i) = 1 - p.$$

Figure 2.2 illustrates the application of n -point ($R_{n\text{-point}}$) and uniform crossovers (R_{Unif}), with parameters n and p , respectively. All the above crossover operators are *uniformity-preserving* according to Definition 2.7.

Figure 2.2: Example of one-point, two-point, n -point and uniform crossover performed between parents x and y , producing a solution z .

GAs use crossover as main operator, however, much has been discussed about the benefit of crossover. Mitchell and Holland (1993) were among the first to provide an experimental comparison between crossover and mutation. More recent studies provided by Jansen and Wegener (2005), Watson and Jansen (2007), Sudholt (2005, 2012), Kötzing et al. (2011) have shown cases where crossover really benefits the search. However, conclusions for the general case are not supported yet.

2.2.4 Working Principles

This section briefly describes GAs' working principles as proposed by Holland (1975). The discussion assumes the simple GA (SGA) as starting-point, defined as

$$P(t+1) = M_{1/N} \circ R_{1\text{-point}} \circ S_{\text{Prop}(f)}(P(t)) \quad (\text{SGA})$$

At each generation t of the SGA, solutions are selected and used to generate a new population. Holland (1975) argued that, when selecting solutions, GAs are at the same time privileging the common characteristics of high-quality solutions composing the current population. Therefore, in order to guarantee a fast improvement of the population along the generations, such characteristics should not be frequently destroyed by crossover and mutation operators. Holland (1975) named such “common characteristics” the *schema* of high-quality solutions.

Definition 2.8 (Schema). A schema $H = s_1 \dots s_N$ is a string of N characters $s_j \in \{0, 1, \star\}$, with $\forall j = 1, \dots, N$. H represents all solutions $\mathbf{x} \in \mathcal{G} : (x_j \neq s_j) \Leftrightarrow (s_j = \star)$.

Definition 2.9 (Order). The order of a schema $o(H)$ is defined as the number of fixed positions in it, $o(H) = |\{j \in [N] : s_j \neq \star\}|$.

Definition 2.10 (Defining length). The defining length of a schema $\delta(H)$ is the distance between the first and the last non- $\star$ character positions.

Along the generations some schemata are preserved whereas others become extinct. It would be the role of selection operators to guarantee that the most promising schema survive. However, GAs do not know which schemata should survive and their randomized variation operators are likely to disrupt previous assembled schemata. Holland (1975) and Goldberg (1989) argued the exchange of low-order schemata as the main mechanism in producing high-quality offspring. Such components were called BBs and used to conjecture the BB-hypothesis as an explanation for GAs working principles.

Definition 2.11 (Building-block). A BB is a low-order, low defining-length schema that defines a set of solutions with average fitness above the population average.

The BB-hypothesis then suggests that GAs would show their best performance if using BBs appropriately and avoiding high rates of disruption. Such ideas brought intense repercussions in the EA field, but also a lot of controversies, as well discussed by Reeves and Rowe (2002). Although such controversies still live on, they did not prevent the BB-hypothesis from influencing the field positively, indeed many studies have emerged from its foundations.

2.3 Estimation of Distribution Algorithms

The efficiency of GAs strongly relies on the choice of appropriate reproduction operators. However, in many situations it is difficult to develop suitable operators and, consequently, GAs

might not perform so well. Estimation of Distribution Algorithms (EDAs) were proposed as an alternative to overcome such difficulties, in which a distribution is estimated to model the current population and from which new solutions can be generated without, necessarily, requiring crossover and mutation operators (Larraanaga and Lozano, 2001; Lozano, 2006). In general, each step of a canonical EDA can be defined as

$$P(t+1) = \beta_\lambda \circ \alpha \circ S(P(t)), \quad (2.24)$$

$$D(t+1) = \alpha \circ S \circ \beta_\lambda(D(t)) \quad (2.25)$$

Algorithm 2.2 illustrates the canonical structure with concrete populations instead of distributions. At generation t , a temporary population P_{selected} is selected from $P(t)$. A distribution D_{model} is then estimated from P_{selected} and sampled $\beta_\lambda(D_{\text{model}})$ to generate new solutions.

Algorithm 2.2 Canonical estimation of distribution algorithm.

Require: An initial population $P(t=0)$.

1: while $t < \text{MAX_STEPS}$ do
2: $P_{\text{selected}} \leftarrow S_f(P(t))$
3: $D_{\text{model}} \leftarrow \alpha(P_{\text{selected}})$
4: $P_{\text{offspring}} \leftarrow \beta_\lambda(D_{\text{model}})$
5: $P(t+1) \leftarrow P_{\text{offspring}}$ | ▷ Different stop criteria can be used.
▷ Select the fittest solutions from P .
▷ Estimate the distribution of P_{selected} .
▷ Sample λ solutions from D_{model} .
▷ $P_{\text{offspring}}$ becomes $P(t+1)$. |
|---|--|
-

Although early EDAs relied on univariate factorizations of the probability distribution, bivariate and multivariate factorizations were also investigated as the connections between EDAs and the BB-hypothesis were recognized. Statistical dependencies are used to define “linkages” among decision variables and the process of linkage identification has been called *linkage-learning*. This section reviews some EDAs of historical importance, with univariate, bivariate and multivariate distributions.

2.3.1 Univariate Factorizations

The simplest EDAs assume the independence of decision variables, therefore $\alpha : \mathcal{G}^\lambda \mapsto D_{\text{Univariate}}$. Two random variables are independent if $p(X_1, X_2) = p(X_1) \cdot p(X_2)$, in the general case of N random variables, the joint probability distribution is factorized in N univariate marginal distributions

$$D_{\text{Univariate}} := p(X_1, \dots, X_N) = \prod_{i=1}^N p(X_i). \quad (2.26)$$

Univariate factorizations are used in many different EDAs, so we describe some of them next. Note that in the binary case $p(X_i = 1) = p(X_i)$.

Univariate Marginal Distribution Algorithms (UMDAs)

The UMDA, proposed by Mühlenbein (1997), uses the operator α_{UMDA} to compute marginal frequencies from a selected population $S(P(t))$ and estimate a distribution $p_{t+1}(X_1, \dots, X_N)$.

$$p_{t+1}(X_i) = \frac{1}{\lambda} \sum_{x \in S(P(t))} x_i, \forall i \in [N]. \quad (\alpha_{\text{UMDA}})$$

The temporary population $P(t)$ is sampled by the operator β_λ , in summary:

$$D(t+1) = \alpha_{\text{UMDA}} \circ S \circ \beta_\lambda(D(t)). \quad (2.27)$$

The UMDA was further extended by González et al. (2002), for example, who analyzed its behavior on linear and quadratic functions when using tournament selection; Hong et al. (2005) evaluated the benefits of adaptive population sizes; Yang (2005) evaluated its performance in dynamic environments and proposed some memory improvements.

Population-Based Incremental Learning (PBIL)

The PBIL, as proposed by Baluja (1994), also represents the population implicitly by its univariate marginal distributions, from which it samples new solutions and updates the model instead of generating a new one. At each generation, μ individuals are sampled and $\lambda \leq \mu$ are selected. Such individuals are, then, used to update the model by the operator α_{PBIL} , as follows

$$p_{t+1}(X_i) = (1 - \gamma)p_t(X_i) + (\gamma/\lambda) \sum_{x \in S(P(t))} x_i, \forall i \in [N], \quad (\alpha_{\text{PBIL}})$$

where $\gamma \in (0, 1]$ is a parameter defining the learning rate, where a small value determines that the previous model $p_t(X_i)$ should be only slightly modified by the new solutions sampled. By using the operator α_{PBIL} , PBIL can be described as

$$D(t+1) = \alpha_{\text{PBIL}} \circ S \circ \beta_\mu(D(t)) \quad (2.28)$$

The PBIL has been extensively applied and analyzed along the years. Sebag and Ducoulombier (1998) extended it to continuous domains; Yang and Yao (2005, 2008) evaluated its performance in dynamic environments; Hong et al. (2008) applied it for unsupervised feature selection, among others.

Compact Genetic Algorithm (CGA)

The CGA, as proposed by Harik et al., (1999), also relies on the implicit populations defined by univariate marginal distributions. At each generation t , two individuals $\mathbf{x}$ and $\mathbf{y}$ are sampled, $P(t) = \beta_2(D(t))$. The population $P(t)$ is then sorted in decreasing order of fitness, $S_{\text{Sort}(f)}(P(t))$,

with $\mathbf{u} = \mathbf{x}_1$ being the best and $\mathbf{v} = \mathbf{x}_2$ being the worst solution. The operator α_{CGA} estimates an univariate marginal distribution as follows

$$p_{t+1}(X_i) = p_t(X_i) + \gamma(u_i - v_i), \quad \forall i \in [N], \quad (\alpha_{\text{CGA}})$$

where, $\gamma \in (0, 1]$ is a constant defining the learning rate, usually set to $\gamma = 1/N$. In the cases which $u_i = v_i$ nothing can be learned from the samples, therefore the model is not updated. Using the operator α_{CGA} , the CGA can be defined in as

$$D(t+1) = \alpha_{\text{CGA}} \circ S_{\text{Sort}(f)} \circ \beta_2(D(t)) \quad (2.29)$$

The CGA is a lightweight EDA suitable for memory-constrained applications, Reza and Arash (2006) proved that the CGA surely converges to local-optima; Kramer and Gallagher (2003) and Kramer et al. (2004) evaluated different versions of the CGA to evolvable hardware architectures and Goldberg et al. (2007) performed a extensive evaluation of a parallel implementation of the CGA in large-scale problems (billions of variables), among others.

2.3.2 Bivariate Factorizations

Although univariate models can be computed efficiently, in optimization problems with non-linear interactions among variables they are not representative enough to provide better performance than GAs. In such problems, decision variables are commonly dependent, therefore, samples from univariate marginal distributions are not expected to produce high-quality solutions. In order to overcome such a drawback, the use of bivariate factorizations was proposed in the EDA community, in which dependencies between pairs of variables could be modeled. A bivariate factorization of the multivariate probability distribution is the product of bivariate marginals, where π_i contains a variable on which X_i depends (in this case $|\pi_i| = 1$).

$$D_{\text{Bivariate}} := p(X_1, \dots, X_N) = \prod_{i=1}^N p(X_i | \pi_i). \quad (2.30)$$

Bivariate and multivariate distributions are usually represented as Probabilistic Graphical Models (PGMs) (graphs), in which edges denote statistical dependencies (or conditional probabilities) and vertices denote variables. The process of learning (from data) the structure of a PGM is called linkage-learning in this context.

Mutual Information Maximizing Input Clustering (MIMIC)

The MIMIC, as proposed by (De Bonet et al., 1997), factorizes the joint probability distribution in a chain-like model representing successive dependencies between variables. It finds a permutation of the decision variables, $r : i \mapsto j$, such that $x_{r(1)}x_{r(2)}, \dots, x_{r(N)}$ minimizes the Kullback-

Leibler divergence in relation to the true probability distribution. In this context $\pi_{r(i+1)} = \{X_{r(i)}\}$. De Bonet et al. (1997) proposed a simple greedy algorithm to find a permutation r , in which the first variable is the one with smallest entropy, the next variable is the one with smallest conditional entropy in relation to the previous one and this process is repeated until all variables have been selected. The operator α_{MIMIC} factorizes the multivariate distribution as follows

$$p_{t+1}(X_1, \dots, X_N) = p_t(X_{r(N)}) \prod_{i=1}^{N-1} p_t(X_{r(i)} | X_{r(i+1)}). \quad (\alpha_{\text{MIMIC}})$$

New solutions are sampled from the leftmost to the rightmost variable, the first is generated independently and the others according to conditional probabilities. Since the estimated distribution must be recomputed each generation, MIMIC uses concrete populations in the following way

$$P(t+1) = \beta_\mu \circ \alpha_{\text{MIMIC}} \circ S(P(t)). \quad (2.31)$$

Bivariate Marginal Distribution Algorithm (BMDA)

The BMDA, as proposed by Pelikan and Muehlenbein (1999), factorizes the joint probability distribution in bivariate marginal distributions. First, a randomly chosen variable is added as a node in a graph, then the most dependent variable to one of those in the graph is chosen among those not yet in the graph. This procedure is repeated until no remaining variable depends on any variable in the graph (verified according to a threshold value).

The resulting BMDA model is a forest with multiple trees rooted at nodes Υ_t . Considering $\mathcal{I}_t$ the non-root variables, α_{BMDA} estimates a factorized distribution, in which the root variables can be sampled independently, whereas all the others must be conditioned on the parent variable π_i .

$$p_{t+1}(X_1, \dots, X_N) = \prod_{X_i \in \Upsilon_t} p_t(X_i) \cdot \prod_{X_i \in \mathcal{I}_t} p_t(X_i | \pi_i). \quad (\alpha_{\text{BMDA}})$$

Each step of BMDA is defined as follows

$$P(t+1) = \beta_\mu \circ \alpha_{\text{BMDA}} \circ S(P(t)). \quad (2.32)$$

2.3.3 Multivariate Factorizations

The next stage of EDA development was the use of multivariate factorizations of the probabilistic distribution. In this case, the joint probability distribution is factorized into a number of multivariate distributions of K bounded order: $|\pi_i| \leq K, \forall i \in [N]$.

$$p(X_1, \dots, X_N) = \prod_{i=1}^N p(X_i | \pi_i) \quad (2.33)$$

Since the learning of PGMs encoding multivariate distributions is a computationally expensive task, it is usual for EDAs to estimate the multivariate statistics from bivariate statistics.

Extended Compact Genetic Algorithm (eCGA)

The eCGA, as proposed by Harik (1997), was one of the first EDAs to employ multivariate factorizations, in which high-order dependencies among decision variables could be modeled. Its approach factorizes the joint probability distribution in the product of multivariate marginal distributions. Assume $\mathcal{T}_{\text{eCGA}} = \{\tau_1, \dots, \tau_\Psi\}$ is a set of disjoint subsets, in which every $\tau \in \mathcal{T}_{\text{eCGA}}$ is a *linkage set*, containing $|\tau| \leq K$ variables. The factorized joint probability distribution estimated by the operator α_{eCGA} is represented as follows

$$p(X_1, \dots, X_N) = \prod_{\tau \in \mathcal{T}_{\text{eCGA}}} p(\tau). \quad (\alpha_{\text{eCGA}})$$

The eCGA popularized the term *linkage-learning* as denoting procedures that identify *linkage sets*. Its linkage-learning procedure relies on two measures: (1) the *Model Complexity* (MC) and (2) the *Compressed Population Complexity* (CPC). The MC quantifies the model representation size in terms of number of bits required to store all the marginal probabilities

$$MC = \log_2(\lambda + 1) \sum_{\tau \in \mathcal{T}_{\text{eCGA}}} (2^{|\tau|-1}), \quad (2.34)$$

The CPC, on the other hand, quantifies the data compression in terms of entropy of the marginal distribution over all partitions, where λ is the selected population size, $|\tau|$ is the number of decision variables in the linkage set τ and $H(\tau)$ is the joint entropy of the variables in τ

$$CPC = \lambda \sum_{\tau \in \mathcal{T}_{\text{eCGA}}} H(\tau). \quad (2.35)$$

The linkage-learning in eCGA works as follows: (1) Insert each variable in a cluster, (2) compute $\text{CCC} = \text{MC} + \text{CPC}$ of the current linkage sets, (3) verify the increase on CCC provided by joining pairs of clusters, (4) effectively joins those clusters with highest CCC improvement. This procedure is repeated until no CCC improvements are possible and produces a *linkage model* $\mathcal{T}_{\text{eCGA}}$. The eCGA works with concrete populations, therefore, using the factorized distribution modeled by α_{eCGA} , it can be described as

$$P(t+1) = \beta_\mu \circ \alpha_{\text{eCGA}} \circ S(P(t)) \quad (2.36)$$

The eCGA has been applied to some practical problems, for instance, forest management planning (Ducheyne et al., 2005), ground-state optimization of clusters (Sastry et al., 2007), community detection in complex networks (Li and Song, 2013), topology optimization (Tominaga et al., 2013) among others.

Bayesian Optimization Algorithm (BOA)

The BOA, as proposed by Pelikan et al. (1999), uses Bayesian networks to model and sample promising solutions. Bayesian networks are directed acyclic graphs, with nodes representing variables and edges representing conditional probabilities between pair of variables. The value of a variable x_i can be conditioned on a maximum of K other variables, defined in π_i . The operator α_{BOA} builds a PGM encoding a factorized joint distribution, in which the parameters of the network, i.e. the conditional probabilities, are estimated from the selected population using the maximum likelihood estimator.

$$p(X_1, X_2, \dots, X_N) = \prod_{i=1}^N p(X_i | \pi_i). \quad (\alpha_{\text{BOA}})$$

The Bayesian network structure, on the other hand, must be built iteratively (linkage-learning). It starts with a network without edges and, at each step, adds the edge which better improves some scoring metric (e.g. Bayesian information criterion (BIC) or Bayesian-Dirichlet metric with likelihood equivalence (BDe)) (Larrañaga et al., 2012). The scoring metric evaluates the network structure according to its accuracy in modeling the selected population. From the built network, BOA samples new promising solutions as follows: (1) it computes the ancestral ordering for each variable, each node being preceded by its parents; (2) each variable is sampled conditionally to its parents. Given such scenario and the operator (α_{BOA}), the BOA can be defined in one-step as

$$P(t+1) = \beta_\mu \circ \alpha_{\text{BOA}} \circ S(P(t)) \quad (2.37)$$

To improve the efficiency and computational costs of BOA, many variants were investigated along the years. Among them, we highlight the hierarchical BOA (Pelikan, 2005; Pelikan and Goldberg, 2003a,b), which has been extensively analyzed (Hauschild et al., 2009; Hauschild et al., 2007; Hauschild et al., 2008; Lima et al., 2009). The BOA has been applied to the nurse scheduling problem (Li and Aickelin, 2006), antenna design (Ha et al., 2014; Santarelli et al., 2006), ground states optimization of Ising sping glasses (Pelikan and Hartmann, 2006; Pelikan et al., 2008), task assignment problems (Li et al., 2014) among others.

Linkage Tree Genetic Algorithm (LTGA)

The LTGA, as proposed by Thierens (2010), differs from most EDAs in the sense it does not explicitly model a probabilistic distribution but only a *linkage model*, called linkage-tree. A linkage model $\mathcal{T}$ is a set of *linkage sets* with no probabilistic distribution associated, therefore, there is no way to sample new solutions directly from $\mathcal{T}$. In the LTGA, the linkage model is a linkage-tree produced by α_{LTGA} and stored as a Family of Subsets (FOS).

$$\mathcal{T}_{\text{LT}} = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_1, x_2\}, \{x_3, x_4\}\}. \quad (\alpha_{\text{LTGA}})$$

The operator α_{LTGA} is a hierarchical clustering algorithm, which work as follows. At each step the two “closest” clusters i and j are merged, this procedure repeats until only one cluster remains. Figure 2.3 illustrates a linkage-tree produced by such a procedure, in which each subtree is stored as a subset $\tau \in \mathcal{T}_{\text{LT}}$.

Figure 2.3: A linkage tree and the possible subsets that can be obtained.

The closeness of two clusters is measured by a distance function $d(i, j)$. If i or j are not singleton clusters, their distance is approximated from the distance of their elements. There are many ways to approximate such distances, but the LTGA uses the *average linkage* approach implemented by the Unweighted Pair Grouping Method with Arithmetic-mean (UPGMA) (Gronau and Moran, 2007).³ If the pair of clusters (C_p, C_q) is merged, creating a new cluster $C_1 = \{C_p, C_q\}$, the distance between C_1 and any other cluster C_k is computed as

$$d_{\text{avg}}(C_1, C_k) := \frac{|C_p| \times d(C_p, C_k)}{|C_1| + |C_k|} + \frac{|C_q| \times d(C_q, C_k)}{|C_1| + |C_k|}, \quad (2.38)$$

Since *linkage sets* should reflect statistical dependencies, variations of the *mutual information* are usually applied as distance measure. Following the results of Bosman and Thierens (2012b), we apply the *normalized variation of information* $NVI(X, Y) = 1 - I(X; Y)/H(X, Y)$. The NVI ranges from total dependence ($NVI(X, Y) = 0$) to total independence ($NVI(X, Y) = 1$) and serve the purposes of linkage-learning.

The LTGA uses $\mathcal{T}_{\text{LT}}$ to guide an *optimal mixing* procedure which resembles a recombination operator but only accepts improving moves. We denote it as R_{LTGA} (see Algorithm 2.3), where the notation $\mathbf{x}[\tau] \leftarrow \mathbf{y}[\tau]$ indicates the transfer of the genetic material indexed by τ from $\mathbf{y}$ to $\mathbf{x}$.

Algorithm 2.3 Optimal mixing recombination (R_{LTGA}).

Require: a family of subsets $\mathcal{T}_{\text{LT}}$ and a population $P(t)$

- 1: **for all** $\mathbf{x}_i$ in $P(t)$ **do**
 - 2: **for all** τ in $\mathcal{T}_{\text{LT}}$ **do**
 - 3: choose a random $\mathbf{x}_j \in P(t) : \mathbf{x}_i \neq \mathbf{x}_j$
 - 4: $f_{\mathbf{x}_i} \leftarrow f(\mathbf{x}_i)$ ▷ Store the current $f(\mathbf{x}_i)$
 - 5: $\mathbf{x}_i[\tau] \leftarrow \mathbf{x}_j[\tau]$ ▷ Crossover variables τ
 - 6: **if** $f(\mathbf{x}_i) < f_{\mathbf{x}_i}$ **then**
 - 7: $\mathbf{x}_i[\tau] \leftarrow \mathbf{x}_j[\tau]$ ▷ If $\mathbf{x}_i$ has worsened, undo
 - return** $P(t)$
-

³The UPGMA implementation used in this study was based on the C clustering library, available at <http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm>.

The LTGA does not implement typical selection operators. Instead, selection is performed during recombination. Similar ideas have been usually applied into local-search heuristics and, in this sense, the LTGA may be seen as an hybrid method. In summary, relying on the operators α_{LTGA} and R_{LTGA} , the LTGA is defined as

$$P(t+1) = R_{\text{LTGA}}(P(t)) \circ \alpha_{\text{LTGA}}(P(t)) \quad (2.39)$$

If an accurate linkage-model is available, the LTGA will probably perform well due to its more intense local exploitation. Otherwise, the search still progresses, but less effectively. Such robustness is not common in model sampling and, indeed, the LTGA has outperformed many of the previous state-of-the-art algorithms (Bosman and Thierens, 2011, 2012a,b; Thierens and Bosman, 2011, 2012a,b,c, 2013).

2.4 Multiobjective Optimization Extensions

Since the first Multi-objective Evolutionary Algorithms (MOEAs), in the 1990's, much of the interest has been focused on *fitness assignment*. In multi-objective problems, objective vectors are not sufficient to define if a solution is desired or not and the fitness must be adapted to consider new features. A candidate solution can only be assigned to a high fitness if it helps to improve both the convergence and the diversity of the population in relation to the Pareto-optimal front. Many *fitness assignment* methods have been proposed along the years, usually associated with MOEAs of historical importance for the field. Single-objective variation operators and EDAs have also been extended to better suit multi-objective optimization.

This section reviews some important techniques implemented by well-known MOEAs. However, it does not intend to be a complete survey of the field, and focuses on the groundings of MOEAs instead of overall developments.

2.4.1 Fitness Assignment

In single-objective optimization the main goal is to find one of the globally optimal solutions in the search space. In cases where multimodality is a concern, a set of diverse high-quality solutions may be desirable and this motivated the use of *niching methods*. Multiobjective problems are intrinsically multimodal in the sense there are many different solutions composing the Pareto-optimal set. In order to find such solutions, the convergence and diversity of solutions must be considered by *fitness assignment*. This section reviews some fitness assignment methods and elucidates how they circumvent the lack of *total ordering* (Definition 2.2) in objective spaces with $M \geq 2$.

Multi-objective Genetic Algorithm (MOGA)

The MOGA, as proposed by Fonseca, Fleming, et al. (1993), extends the idea proposed by Goldberg (1989) that suggests a fitness assignment method based on the ranking of objective vectors. In MOGA, the ranking of any solution $\mathbf{x} \in P(t)$ is defined from the number of solutions which dominate it

$$\Lambda_{>_P}^t(\mathbf{x}, P(t)) = |\{\mathbf{y} \in P(t) : \varphi(\mathbf{y}) >_P \varphi(\mathbf{x})\}|. \quad (2.40)$$

Fonseca, Fleming, et al. (1993) suggested the ranking should be directly related to $\Lambda_{>_P}^t(\mathbf{x}, P(t))$ whereas fitness should be inversely related to $\Lambda_{>_P}^t(\mathbf{x}, P(t))$

$$\text{rank}(\mathbf{x}, t) = 1 + \Lambda_{>_P}^t(\mathbf{x}, P(t)). \quad (2.41)$$

Figure 2.4 illustrates the rankings of some objective vectors according to Equation (2.41). The grid lines delimit the regions dominated by each objective vector.

Figure 2.4: Ranking assignment as implemented in MOGA. Objective vectors get ranking proportional to the number of objective vectors dominating them.

After ranking assignment, fitness is assigned as follows: (1) sort the population according to rank, (2) assign fitnesses to individuals by interpolating from the smallest ranking to the largest, (3) the fitness of individuals with the same ranking is averaged so as to allow them the same reproduction chance. Next, sharing is applied in the objective space to favor diversity.

Non-dominated Sorting Genetic Algorithm (NSGA)

The NSGA, proposed by Srinivas and Deb (1994), also followed the idea proposed by Goldberg (1989), but relies on the non-dominated sorting of fronts. First, all non-dominated solutions are selected to compose the first front F_1 and $\forall \mathbf{x} \in F_1$ receives the same fitness value. In order to keep diversity of solutions, *sharing* is performed within F_1 . From the remaining population $P(t) \setminus F_1$, the non-dominated solutions are again extracted to compose the second front F_2 .

Solutions in F_2 receive a fitness value smaller than the minimum shared fitness in F_1 and *sharing* is also performed. This procedure is repeated until all solutions have been assigned to a front and fitness (see Figure 2.5).

Figure 2.5: Fitness assignment by non-dominance sorting, as implemented by the NSGA. Solutions in the first fronts receive higher fitness values.

After *fitness assignment* the population is reproduced by common variation operators and selection is performed by *stochastic remainder proportionate selection* (Booker, 1982). Since a higher fitness is assigned to individuals in the first fronts and/or in less populated regions, more offspring are generated in such regions. In summary, the non-dominance sorting favors solutions closer to the Pareto-optimal front whereas the *fitness sharing* mechanism maintains the diversity of solutions along the front.

Strength Pareto Evolutionary Algorithm (SPEA)

The SPEA, proposed by Zitzler and Thiele (1999), introduces to MOEAs two ideas that were extensively adopted in next generation algorithms: (1) elitism and (2) the use of an external archive A to store non-dominated solutions. Elitism is guaranteed by performing selection on the union of population and archive $P(t) \cup A(t)$. Since the archive stores the non-dominated solutions found so far, good solutions are not lost, speeding up the convergence.

Concerning *fitness assignment*, the SPEA uses dominance ranking in the opposite way of MOGA, it calls the *strength* of $\mathbf{x} \in P(t)$, the value $s(\mathbf{x}, t) = \Lambda_{\geq P}^t(\mathbf{x}, P(t))/\mu$, in which

$$\Lambda_{\geq P}^t(\mathbf{x}, P(t)) = |\{\mathbf{y} \in P(t) : \varphi(\mathbf{x}) \geq_P \varphi(\mathbf{y})\}|. \quad (2.42)$$

The fitness of $\mathbf{x}$ is computed by summing the strengths of all solutions $\mathbf{y} \in A(t)_{\geq P}$, where $A(t)_{\geq P} = \{\mathbf{y} \in A(t) : \varphi(\mathbf{y}) \geq_P \varphi(\mathbf{x})\}$. This procedure guarantees solutions in $A(t)$ to have

better fitness than those in $P(t)$.⁴

$$f(\mathbf{x}) = - \left(1 + \sum_{\mathbf{y} \in A(t) \geq_P} s(\mathbf{y}, t) \right). \quad (2.43)$$

After *fitness assignment*, binary tournament selection is performed on $P(t) \cup A(t)$ and traditional variation operators produce $P(t + 1)$. The non-dominated solutions in $P(t + 1)$ are then copied to $A(t + 1)$ and the dominated solutions excluded. If the size of $A(t)$ exceeds a certain limit, the most diversified subset must be selected. Solutions in $A(t)$ are clustered according to their distance (in objective space) in μ' clusters (maximum size of $A(t)$), then one solution is chosen from each cluster so as to obtain a well spread set.

Non-dominated Sorting Genetic Algorithm II (NSGA-II)

The NSGA-II was proposed by Deb et al. (2002) as an improved version of the NSGA, in which the following criticisms to the NSGA were solved: (1) the high computation cost of non-dominated sorting, (2) the lack of elitism and (3) the need for specifying the sharing parameter (Deb et al., 2002). A faster non-dominated sorting was developed, elitism was implemented following the idea of SPEA and fitness sharing was substituted by the *crowding method*.

The NSGA-II classifies the population in multiple fronts more efficiently than NSGA. The *crowding distance* measures the volume of the box containing each objective vector in relation to its neighbors. It is used to identify how isolated objective vectors are in the objective space. Figure 2.6 exemplifies the boxes containing objective vectors z_1 and z_2 , where z_1 has a larger crowding distance than z_2 .

Figure 2.6: Examples of the crowding distance for objective vectors z_2 and z_3 . A small crowding distance (a small box in the figure) means the objective vector is in a more populated region in objective space.

⁴We set $f(\mathbf{x}) = -f(\mathbf{x})$ to consider the maximization of fitness, instead of minimization.

From the dominance ranking ($\text{rank}(\mathbf{x})$) and the crowding distances ($\text{crow}(\mathbf{x})$), the binary relation ' $\succeq$ ' can be defined for $\forall \mathbf{u}, \mathbf{v} \in \mathcal{P}$ as follows

$$\mathbf{u} \succeq \mathbf{v} = \begin{cases} \text{true}, & \text{if } (\text{rank}(\mathbf{u}) < \text{rank}(\mathbf{v})) \\ & \text{or } (\text{rank}(\mathbf{u}) = \text{rank}(\mathbf{v})) \wedge (\text{crow}(\mathbf{u}) > \text{crow}(\mathbf{v})) \\ \text{false}, & \text{otherwise.} \end{cases}$$

Differently from the NSGA, the NSGA-II performs selection by binary tournaments. Each pair of selected parents undergo crossover and mutation producing μ candidate solutions in $P_{\text{offspring}}$. From the combined set $P(t) \cup P_{\text{offspring}}$ non-dominated sorting is performed and the best μ solutions are maintained to compose $P(t+1)$.

Indicator-based Evolutionary Algorithms

The MOGA, SPEA and NSGA-II created the foundations for the main advances reached by MOEAs in the 1990's. However, in many-objective optimization ($M > 3$), most of these approaches do not work properly and alternative approaches have been proposed in order to provide better scalability in terms of M (Zitzler and Künzli, 2004).

The hypervolume $H(\mathcal{F}, \mathbf{z}_{\text{ref}})$ is one of such alternatives and it measures the dominated volume covered by a non-dominated front $\mathcal{F}$ in relation to the reference point $\mathbf{z}_{\text{ref}}$. Hypervolume indicators quantify convergence and diversity of a set of solutions simultaneously. Besides, it is compatible with Pareto-dominance relations in the sense it also can be used to approximate the Pareto-optimal front (Bader, 2009). Figure 2.7 gives an example of the volume covered by a non-dominated front, the hatched region represents the contribution of $\mathbf{z}_1$ to the hypervolume, $C(\mathbf{z}_1, \mathbf{z}_{\text{ref}})$.

$$C(\mathbf{z}_1, \mathbf{z}_{\text{ref}}) = H(\mathcal{F}, \mathbf{z}_{\text{ref}}) - H(\mathcal{F} \setminus \{\mathbf{z}_1\}, \mathbf{z}_{\text{ref}}) \quad (2.44)$$

Knowles and Corne (2003) and Knowles (2002) were the first to propose the use of the hypervolume indicator to implement selection in MOEAs. Huband et al. (2003) proposed a version of the Strength Pareto Evolutionary Algorithm 2 (SPEA2) in which a hypervolume-related measure substitutes the original density estimation technique. Zitzler and Künzli (2004) used a binary hypervolume indicator to compare solutions and assign fitness. One of the first MOEA to propose an alternative use of the hypervolume indicator was the steady-state algorithm proposed by Beume et al. (2007), which combines non-dominated sorting and hypervolume measure (see also Igel Christian et al. (2007)).

Although many different approaches have been proposed using the hypervolume indicator, most of them rely on the contribution of individual solutions to the total indicator value (Bringmann and Friedrich, 2012; Emmerich and Fonseca, 2011). From a set-based perspective (Zitzler et al., 2010), given a set composed of $\lambda > \mu$ solutions, the selection step must chose the μ

Figure 2.7: The Dominated hypervolume in relation to z_{ref} (shaded region) measures convergence and diversity simultaneously. The hatched region represents the contribution of z_1 to the dominated hypervolume.

solutions which yield the maximum indicator value (the subset selection problem). The same problem can be defined in the context of other quality indicators (Vaz et al., 2013).

2.4.2 Reproduction Operators

This section describes some of the extensions of single-objective *reproduction operators* investigated in the context of multi-objective optimization. We call a *reproduction operator* any operator responsible for generating new solutions, e.g. *variation operators*, *sampling operators*.

Crossover-based Reproduction

Most of the studies regarding the effectiveness of crossover operators have emerged in the context of mating restrictions (Goldberg, 1989). It has been shown that a restricted crossover, performed only between similar parents, is a simple way to generate more diverse sets of non-dominated solutions. However, most of the MOEAs previously described still use unrestricted n -point and p -uniform crossovers, with usual parameters $n = 1$ and $p = 0.5$ (Deb et al., 2002; Srinivas and Deb, 1994; Zitzler and Thiele, 1999; Zitzler et al., 2002). This section reviews some of the main studies on crossover and restricted mating.

Mate restrictions have been studied since the early days of MOEA development (Fonseca and Fleming, 1995). In the context of fitness sharing, for example, the parameter σ_{share} can also be applied to define a maximum distance between parents σ_{mate} allowed to reproduce. Many other authors have treated this subject at different levels of depth (Jaszkiewicz, 2002b; Veldhuizen and Lamont, 2000; Zitzler and Thiele, 1999). However, what seems to be the most complete experimental investigation started with Ishibuchi and Shibata (2003b), who proposed a flexible scheme for the implementation of mating restrictions in the decision and/or objective spaces. The method works as follows: (1) a parent x is chosen by tournament selection ($q = 2$); (2) $\Omega > 0$ other solutions are chosen by tournament selection; (3) among the Ω solutions, the closest one to

x , in the objective space, is chosen as a mate. The authors evaluated such mating restrictions with different values of Ω in multiobjective knapsack and flowshop scheduling problems and positive results regarding diversity or convergence were achieved, depending on Ω .

Ishibuchi and Shibata (2003a) also proposed a different version of the procedure, which biased the choice of the first parent towards extreme regions in the objective space. In this case, ω possible parents are chosen by tournament selection and the one most distant from their centroid is chosen as the first parent. The second parent is chosen among other Ω possible parents, as originally proposed. The results showed improvements in diversity and convergence for appropriate values of (ω, Ω) . In another study, Ishibuchi and Narukawa (2005) showed increases in the tournament size (q) would lead to more dissimilar solutions, which improved diversity but deteriorated convergence. Experiments with the SMS-EMOA (Beume et al., 2007) were also performed and showed the effectiveness of their method for problems with many-objectives (Ishibuchi et al., 2008a, 2012).

Ishibuchi et al. (2007) also argued that geometric crossover decreases the diversity of solutions in the objective space. In order to avoid such a drawback, Ishibuchi et al. (2010) proposed a non-geometric binary crossover which acted simultaneously with mating restrictions to guarantee a good balance between diversity/spreading and convergence. More recently, Sato et al. (2013) and Ishibuchi et al. (2014) analyzed the importance of generating offspring close to the parents, from the perspective of restricted mating, as a way of obtaining better diversity.

Probabilistic Model-based Reproduction

Single-objective EDAs usually differ according to the probability distribution estimated and the approach used to sample new solutions (see Section 2.3). The extension of such algorithms to the multi-objective context requires the implementation of appropriate *fitness assignment*. The first Multi-objective Estimation of Distribution Algorithms (MOEDAs), borrowed the *fitness assignment* of well-known MOEAs, such as the NSGA-II and SPEA2, enabling a straightforward evaluation of the benefits of MOEDAs.

Laumanns and Ocenasek (2002), for example, proposed the Bayesian Multi-objective Optimization Algorithm (BMOA) using an special selection operator based on the ϵ -archive. The BMOA was shown to outperform the NSGA-II and SPEA2 on instances of the multiobjective knapsack problem. Essentially, the BMOA combines the mixed BOA proposed by Ocenasek et al. (2004) and the SPEA2 proposed by Zitzler et al. (2002). Khan et al. (2002) also proposed a multiobjective BOA by implementing the fitness assignment of the NSGA-II, results for multiobjective deceptive problems were reported. Shah and Reed (2011) confirmed the results of Ocenasek et al. (2004) in multiobjective multidimensional knapsack problems using other version of the hierarchical Bayesian Optimization Algorithm (hBOA), which implemented the ϵ -nondominated archiving and produced the best results among other MOEAs.

Controversially to such promising results, theoretical studies regarding artificial deceptive functions suggested the massive multimodality in the middle of Pareto-optimal fronts could

spoil the benefits of model-building (Sastry et al., 2005). Such an allegation led to the use of multiple models in MOEDAs. Pelikan et al. (2005), for example, proposed the Multi-objective Hierarchical Bayesian Optimization Algorithm (mhBOA) which applies clustering in the objective space and builds a model for each cluster, a similar idea was used by Bosman et al. (2006) with the Multi-objective Iterated Density Estimation Algorithm (MIDEA). In such MOEDAs, solutions in different regions are generated by sampling different models.

Martí et al. (2011a,b) went further and analyzed the weaknesses of the previous MOEDAs in terms of model-building and computational complexity. The authors evidenced the need for a different treatment of *outliers* and investigated the use of less expensive model-building algorithms. According to the authors, outliers represent solutions in underexploited regions of the objective space, hence should be emphasized during model-building. Following such ideas, Martí et al. (2008a) proposed the Multi-objective Neural Estimation of Distribution Algorithm (MONEDA) (Martí et al., 2011a).

Decomposition-based Reproduction

Decomposition-based MOEAs, as the name suggests, decompose the MOP into multiple single-objective problems. Each of the single-objective problems is defined by aggregations of the M objectives. In the simplest case, the *weighted sum approach*, an aggregation $g_j(\mathbf{x})$ is generated through the choice of a weight vector $\mathbf{w} \in \mathbb{R}^M$, such that $\sum_{i=1}^M w_i = 1$. Assume $\mathbf{z} = \varphi(\mathbf{x})$, with $\mathbf{z} \in \mathbb{R}^M$, a single-objective optimization problem is defined by the aggregation

$$g_j(\mathbf{x}) = \sum_{i=1}^M w_i \cdot z_i. \quad (2.45)$$

Any other aggregation function could be applied, see (Miettinen and Mäkelä, 2002). The weight vectors $\mathbf{w} \in \mathbf{W}$ define specific directions in the objective space; therefore, if they are properly chosen, the search can produce a well-distributed Pareto-front (Li and Zhang, 2009; Zhang and Li, 2007; Zhang et al., 2009).

One of the first MOEAs to implement decomposition was the Multi-objective Genetic Local Search (MOGLS) proposed by Jaszkiewicz (2002a,b). MOGLS considers all possible aggregations during a search, but randomly chooses only one in each generation. The Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D), proposed by Zhang and Li (2007), on the other hand, uses a more structured hybridization by properly managing the aggregations. For each weigh vector $\mathbf{w} \in \mathbf{W}$, MOEA/D computes the closest T weigh vectors $B(\mathbf{w})$. Each aggregation $\mathbf{w}$ leads to a single-objective problem that is solved separately by using a population composed of solutions generated by neighboring aggregated functions in $B(\mathbf{w})$. Regarding reproduction, traditional crossover and mutation are used (Li and Zhang, 2009; Ma et al., 2014; Tan et al., 2013; Zhang et al., 2010). We might say that MOEA/D implements restricted mating in the context of neighboring aggregation functions in $B(t)$.

Much has been investigated about the MOEA/D, Li and Zhang (2009), for example, compared MOEA/D with the NSGA-II on MOPs on complicated Pareto-optimal sets. Zhang et al. (2010) proposed a version of the MOEA/D for MOPs with expensive function evaluations. Konstantinidis and Yang (2011) develop a problem-specific MOEA/D for finding energy-efficient wireless sensor network design, whereas Tan et al. (2013) implemented MOEA/D for many-objectives optimization.

Since decomposition-based MOEAs do not rely directly on Pareto-dominance and non-dominated sorting, they circumvent most of the difficulties faced by Pareto-dominance in many-objective optimization (Ishibuchi et al., 2009). Selection occurs in the context of the single-objective aggregations and the diversity of the Pareto-fronts are handled by properly choosing the weight vectors. Many different decomposition-based MOEAs have been proposed, see for example, the studies of Liu et al. (2014) and Giagkiozis et al. (2013, 2014), and it seems to be one of the current promising fields of research in MOEAs.

2.5 Conclusions

This chapter reviewed general concepts of single- and multi-objective evolutionary optimization. Through the unified model of evolution proposed by Paixão et al. (2014), Evolutionary Algorithms (EAs) could be described from a unique framework which put in evidence the similarities and peculiarities of specific implementations. The main focus of this chapter was to provide the background required to understand the context in which this thesis was proposed.

Regarding *single-objective optimization*, Genetic Algorithms (GAs) and some operators for binary encodings, proposed by Holland (1975), were reviewed and recent developments referenced. The Building Block (BB) hypothesis was briefly described and grounded a discussion about the early developments in Estimation of Distribution Algorithms (EDAs). Many EDAs have been developed along the years so as to test evidences for the BB hypothesis. Their model-building procedures (linkage-learning) were usually associated with the idea of BB identification as described in this chapter. Due to this reason, recent EDAs usually estimate multivariate distributions in order to handle practical optimization problems. However, they rely on the assumption that multivariate dependencies can be assessed from bivariate dependencies. Such premise enables more efficient linkage-learning methods but might not produce accurate models. Chapters 3-5 investigate such situations in order to explain the origins of the limitations of linkage-learning in EDAs.

Regarding *multi-objective optimization*, many methods were described which implement *fitness assignment* for Multi-objective Optimization Problems (MOPs) and extend single-objective *reproduction operators*. In the context of linkage-learning, some controversies concerning the use of probabilistic models in Multi-objective Estimation of Distribution Algorithms (MOEDAs) were clarified, providing the background for the investigation of the role of linkage models in multi-objective problems. Chapter 6 builds upon the results for single-objective problems in

order to explain the limitations of linkage-learning in bi-objective problems and explain the benefits of using MOEDAs.

Chapter 3

Fundamentals of Linkage-learning

Linkage-learning is a fundamental step of bivariate and multivariate Estimation of Distribution Algorithms (EDAs), in which the structure of a Probabilistic Graphical Model (PGM) (linkage model) is defined from the statistical dependencies observed in the population. By using the information encoded into PGMs, different reproduction operators have been proposed in the literature, most of them relying on model-sampling instead of crossover and mutation. Although linkage-models have also been investigated in the context of crossover by Harik et al. (2006) and Thierens (2010), few studies have concerned its benefits and/or drawbacks in a broader scenario. This chapter formalizes the notion of linkage-guided crossover operators that ground the analysis of the limits of usefulness for linkage-learning in single- and multi-objective optimization.

3.1 Introduction

Consider the genotype space $\mathcal{G} \in \{0, 1\}^N$, composed of N -dimensional binary vectors that represent candidate solutions for a black-box optimization problem. A population $P(t) \subset \mathcal{G}$, at generation $t > 0$, is a multiset composed of μ samples $\mathbf{x} \in \mathcal{G}$ and it is usually represented as a matrix of dimensions (μ, N) , where each row represents a decision vector $\mathbf{x}_i \in \mathcal{G}$ and each column contains μ samples for decision variables x_j , $\forall i \in [N]$.

$$P = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_\mu \end{pmatrix} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots \\ x_{\mu,1} & \cdots & x_{\mu,N} \end{pmatrix} \quad (3.1)$$

The population $P(t) \sim D(t)$ provides evidence about the distribution of optimal solutions in the genotype space (D^*). By estimating the distribution $D(t+1)$ from the best solutions in $P(t)$, EDAs aims at guaranteeing that $D(t) \approx D^*$ as t grows. Distributions $D(t)$ are usually encoded as PGMs which describe dependencies among variables and their conditional probabilities. Given accurate PGMs, EDAs are expected to enable a better synergy between selection and reproduction operators, hence increasing the efficiency of the search.

Without loss of generality, it can be assumed that high-quality solutions are not uniformly distributed in the search space $\mathcal{G}$. Additionally, close solutions in decision space are expected to have similar fitness, a characteristic denoted as *locality* (Rothlauf, 2006). Since selection operators prioritize high-quality solutions, some regions of the search space might become more populated than others. This fact induces statistical dependencies among random variables and can be used to build a PGM. From the perspective of crossover, only the structures of PGMs are required, which are produced by linkage-learning and usually called *linkage models*, e.g. in the Bayesian Optimization Algorithm (BOA) the linkage model is a directed acyclic graph.

Following from the ideas of Holland (1975), statistical dependencies (linkages) are associated with the concept of Building Block (BB). Therefore, reproduction operators that disrupt previously enforced dependencies would be likely to produce non-promising solutions. In this context, the effectiveness of reproduction operators depends on the accuracy of the linkage models employed in their guidance. In the case of crossover operators, the linkage-models must be composed of subsets of variables. In the general case, there is a large number of possible linkages to be identified from N decision variables, indeed such number is equivalent to the powerset of N

$$\mathcal{T}^* = 2^{\mathcal{I}},$$

where $\mathcal{I} = \{1, \dots, N\}$ contains the indexes for every variable and $\mathcal{T}^*$ contains all possible subsets of variables. $\mathcal{T}^*$ is the general search space for linkage-learning methods, from which a linkage-model can be identified. However, for practical reasons, linkage-learning is usually performed in smaller search spaces $\mathcal{T} \subset \mathcal{T}^*$, in which subsets sizes are bounded to $K \geq 2$. The linkage-model then defines a multivariate factorization of the probability distribution describing the current population.

A linkage-model is composed of $\Psi > 0$ linkage sets $\tau \in \mathcal{T}$, which can be used to guide crossover, as described in Algorithm 3.1. The linkage-guided crossover produces a solution $\mathbf{x}$ by using the genetic material of parents $\mathbf{u}, \mathbf{v}$ according to the linkage sets τ . For example, consider the subsets $\tau_1 = \{1, 3, 4\}$, $\tau_2 = \{2, 5, 6\}$ and the parents $\mathbf{u} = (0, 1, 1, 1, 0, 0)$, $\mathbf{v} = (1, 1, 0, 0, 0, 1)$. Therefore, $\mathbf{x}$ is produced by copying values from $\mathbf{u}$ and $\mathbf{v}$, respectively: $\mathbf{x}[\tau_1] \leftarrow \mathbf{u}[\tau_1]$ and $\mathbf{x}[\tau_2] \leftarrow \mathbf{v}[\tau_2]$, which results in a candidate solution $\mathbf{x} = (\underline{0}, 1, \underline{1}, \underline{1}, 0, 1)$.

Algorithm 3.1 Linkage-guided crossover.

Require: Two solutions $\mathbf{u}, \mathbf{v}$ and a linkage-model $\mathcal{T}$.

Ensure: An offspring $\mathbf{x}$

for all $i \in [\Psi]$ **do**

if $i \bmod 2 = 0$ **then**

$\mathbf{x}[\tau_i] \leftarrow \mathbf{u}[\tau_i]$

else $\mathbf{x}[\tau_i] \leftarrow \mathbf{v}[\tau_i]$

return $\mathbf{x}$

▷ All linkage sets in $\tau_i \in \mathcal{T}$.

▷ Subsequent τ_i from different parents.

In the simplest case, which assumes the independence of decision variables (without linkage-learning) the resulting “linkage” model $\mathcal{T}_{\text{univariate}} = \{\{1\}, \dots, \{N\}\}$ is not useful for a linkage-

guided crossover since it induces an $(N - 1)$ -point crossover. Due to this reason, $\mathcal{T}$ used by Algorithm 3.1 it is assumed to be composed of linkage sets of bounded size $K \geq 2$.

The linkage-models used by the Extended Compact Genetic Algorithm (eCGA) consider disjoint subsets of variables, i.e. $\mathcal{T}_{\text{eCGA}} \subset \mathcal{T}^*$ such that

$$\left(\bigcap_{\forall \tau \in \mathcal{T}_{\text{eCGA}}} \tau = \emptyset \right) \wedge \left(\bigcup_{\forall \tau \in \mathcal{T}_{\text{eCGA}}} \tau = \mathcal{I} \right). \quad (3.2)$$

However, there is no impediment for using non-disjoint linkage-sets on the linkage-guided crossover defined by Algorithm 3.1.

The accuracy of $\mathcal{T}$ in modeling the structure underlying a population $P(t)$ influences the quality of the offspring produced. This fact is more evident in the context of additively separable functions. A function is defined additively separable if its maximum (minimum) can be computed from the maximum (minimum) value of m sub-functions f_i , defined on hyperplanes of the whole N -dimensional binary space, i.e. each f_i evaluates only a subset of $\{x_1, \dots, x_N\}$, as follows:

$$\varphi(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x}, \pi_i), \quad \mathbf{x} \in \{0, 1\}^N \quad (3.3)$$

where π_i defines the hyperplane in which the i -th sub-function operates. The abuse of notation in terms of π (see Section 2.3) is intentional to make clear the relation between the objective function and the distribution underlying the population.

Deceptive functions are well-known examples of additively separable functions commonly used to benchmark EDAs, in which, additionally, $\cap_{i=1}^m \pi_i = \emptyset$ (Goldberg, 2002). Traditional crossover and mutation operators are able to find locally optimal by exploring the hyperplanes separately, i.e. finding local or global optima configurations for each sub-function. However, as soon as solutions for the sub-functions have been found, the progress tends to stop due to the disruptive effects of crossover. At such a stage, crossover is only effective if properly mixes the BBs already found. However, they are not known beforehand and Genetic Algorithms (GAs) usually require an exponential number of fitness evaluations to find the global optimum. By employing linkage-learning, these BBs can be identified and much better results have been reported in the literature (Harik et al., 2006). For non-separable functions, where $\cap_{i=1}^m \pi_i \neq \emptyset$, there are no possible linkages that completely avoid disruption of previously built substructures, but useful linkages can still be identified (Pelikan and Goldberg, 2003b).

Theoretical analysis and empirical results have shown that, even in additively separable functions, there are limits for the usefulness of linkage-learning. Coffin and Smith (2007), for example, reported that linkage-learning based on bivariate statistics cannot find high-order linkages for the Concatenated Parity Functions (CPFs) due to an intrinsic *allelic-pairwise independence* between decision variables. However, the authors did not explain which properties of an optimization problem might lead to *allelic-pairwise independence*. According to Echegoyen et al. (2011),

the degree of interactions between decision variables affects the linkage-learning accuracy. In the remaining of this chapter the limits of linkage-learning are theoretically investigated from the perspective of multimodality, in the context of additively separable functions.

Section 3.1.1 briefly reviews some basic concepts from information theory that grounds the analysis about the limits of linkage-learning. Section 3.1.2 defines the necessary conditions for a correct linkage identification. Section 3.1.3 explains how such conditions are affected by multimodality, which may be one of the main sources of linkage-learning limitations. Section 3.2 uses the conditions to assess the difficulty of learning some well-known additively separable functions. Section 3.4 summarizes the results and concludes the chapter.

3.1.1 Information Theory Background

One of the fundamental concepts in information theory is *entropy*, which measures the uncertainty associated with a random variable. Consider, for example, a random variable X that can assume two distinct values $\xi = \{0, 1\}$, each with probability denoted by $p_X(x)$, $x \in \xi$.

Definition 3.1 (Entropy (Kelbert and Suhov, 2013)). The entropy $H(X)$ of X measures the expected amount of information gained from the observation of X and is defined in the support set $\mathcal{S}_X = \{x \in \xi : p_X(x) > 0\}$

$$H(X) := - \sum_{x \in \mathcal{S}_X} p_X(x) \log_2 p_X(x). \quad (3.4)$$

By this definition, the information gain is proportional to the uncertainty, therefore, the entropy increases as the probability distribution of X becomes closer to uniform, i.e. with $p_X(x) = |\xi|^{-1}$, $\forall x \in \xi$, the entropy is maximum. Considering the binary alphabet ξ , the entropy is maximal if $p_X(x) = 1/2$, $\forall x \in \xi$ and the entropy is minimal if $p_X(x) = 1$, for some $x \in \xi$.

Since the focus is on linkage-learning, we must consider at least two random variables and the previous definition must be extended for joint probability distributions and joint entropies.

Definition 3.2 (Joint entropy (Kelbert and Suhov, 2013)). The joint entropy of a binary random vector (X, Y) with joint probability distribution $p_{XY}(x, y)$ is defined by

$$H(X, Y) := - \sum_{x \in \mathcal{S}_X} \sum_{y \in \mathcal{S}_Y} p_{XY}(x, y) \log_2 p_{XY}(x, y). \quad (3.5)$$

Again, the joint entropy increases as the joint probability distribution of X becomes closer to uniform, i.e. with $p_{XY}(x, y) = |\xi|^{-2}$, $\forall x, y \in \xi$ the joint entropy is maximum. From Equations (3.4) and (3.5), the amount of information a random variable X conceives about another random variable Y can be measured by their mutual information.

Definition 3.3 (Mutual Information (Kelbert and Suhov, 2013)). Given binary random variables X and Y , their mutual information is defined by $I(X; Y) := H(X) + H(Y) - H(X, Y)$.

The mutual information is $I(X; Y) \geq 0$, with equality holding iff $p_{XY}(x, y) = p_X(x)p_Y(y)$, i.e. when X and Y are independent random variables. Although the mutual information measures statistical dependence, it is not strictly a metric and the *variation of information* is preferred (Thierens and Bosman, 2011).

Definition 3.4 (Variation of information). Given binary random variables X and Y , the variation of information is defined by

$$\begin{aligned} VI(X; Y) &:= H(X, Y) - I(X; Y) \\ &= 2H(X, Y) - H(X) - H(Y). \end{aligned} \quad (3.6)$$

The variation of information is $VI(X; Y) \geq 0$ and decreases as the mutual information $I(X; Y)$ increases, i.e. $VI(X; Y) \approx 0$ indicates X and Y are statistically dependent variables. In other words, the variation of information measures the statistical independence between random variables. Although $VI(X; Y)$ is preferable as a metric, it is easily derived from the mutual information and the joint entropy. Therefore, for the purpose of our theoretical analysis it is sufficient to use $I(X; Y)$ and $H(X, Y)$.

3.1.2 Necessary Conditions for Learning

Assuming the context of additively separable functions, Equation (3.3), of order $k = |\pi_i|$, $\forall i = 1, \dots, m$, it is shown that as the number of local optima, for each function f_i , grows, it becomes harder to detect if variables are statistically dependent or not. To illustrate such an idea and define a basic separability condition, assume $k = 2$, where X and Y are variables from the same subspace π_i and Z is a variable from another subspace π_j , $\forall j \neq i$. From the *mutual information* (Definition 3.3), the statistical dependence between X and Y will be correctly identified if the following general condition is observed:

$$I(X; Y) > I(X; Z).$$

It is important to evidence that $I(X; Y) > I(Y; Z)$ must also be true, but since linkage learning iteratively evaluates each pair of variables, if the first condition holds the second is ensured as a consequence. It follows from Definition 3.3,

$$\begin{aligned} H(X) + H(Y) - H(X, Y) &> H(X) + H(Z) - H(X, Z) \\ H(Y) - H(X, Y) &> H(Z) - H(X, Z). \end{aligned} \quad (3.7)$$

In the simplest scenario, in which all sub-functions f_i are equal, $H(Y) = H(Z)$ and it follows:

$$H(X, Y) < H(X, Z). \quad (3.8)$$

In a linkage-learning procedure based on measurements of bivariate statistics, it is clear that (3.7) and (3.8) are necessary conditions for a correct modeling, i.e. the pairwise joint entropy should reflect the separability of the fitness function. To a certain extent, they also define valid necessary conditions for any linkage-learning algorithm which estimates high-order dependencies through bivariate statistics.

3.1.3 Origins of Learning Difficulty

In a population generated uniformly at random, inequalities (3.7)-(3.8) do not hold. The joint probability distribution of (X, Y) is uniform, therefore the joint entropy is maximum and there is no *mutual information*. As selection and reproduction are performed, individuals in regions of better fitness survive, which changes the distribution of solutions in the search space. By definition, selection decreases diversity, consequently, the joint entropy among decision variables may decrease and statistical dependencies become more evident.

For some functions, even after selection and reproduction, the decrease in the joint entropy might be too small. In such cases, linkage-learning would not identify useful linkages and, by trusting the linkage-model, the search would become less effective (Chen and Yu, 2009; Coffin and Smith, 2007). Figure 3.1 shows two continuous fitness landscapes used to illustrate the effect of multimodality on linkage-learning. Figure 3.1(a) shows an example in which the dependence between X and Y can be easily observed from statistics: there are only five local attractors and, after selection, a large part of the search space becomes uninhabited. Since many pairs of values do not occur in the population anymore, the mutual information between X and Y is increased. Figure 3.1(b) illustrates a more difficult situation, with a larger number of local attractors: even after selection, the pairs of values are very diversified, which implies a small decrease in the joint entropy and a more difficult observation of statistical dependencies.

Figure 3.1: Population before (crosses) and after selection (circles). According to the number of local attractors statistical dependencies might be harder to detect.

These small examples illustrate the hypothesis that multimodality imposes serious limitations on linkage-learning due to the difficulty in observing useful linkages. As the degree of multimodality increases, linkage-learning is expected to become less useful. The remainder of this section deals with the analysis of different separable functions so as to verify the validity of this hypothesis and understand its consequences on the limits of usefulness of linkage-learning in more general pseudo-Boolean functions.

3.2 Methodology

To investigate the impact of multimodality on linkage-learning, there must be ways to measure the linkage-learning difficulty and compare the measurements in functions of different degrees of multimodality. The linkage-learning difficulty can be estimated from conditions (3.7)-(3.8). If the function in question is composed of equal sub-functions, condition (3.8) can be explored to define Equation (3.9), with $\delta_H > 0$ if condition (3.8) is true.

$$\begin{aligned} H(X, Y) &< H(X, Z) \\ 0 &< H(X, Z) - H(X, Y) \\ \delta_H &= \begin{cases} 1 - H(X, Y)/H(X, Z), & \text{if } H(X, Z) > 0 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (3.9)$$

If, on the other hand, the function is composed of different sub-functions, only condition (3.7) is applicable, from which Equation (3.10) can be derived.

$$\begin{aligned} I(X; Y) &> I(X; Z) \\ 0 &< I(X; Y) - I(X; Z) \\ \delta_I &= I(X; Y) - I(X; Z). \end{aligned} \quad (3.10)$$

In functions with a large δ_H or δ_I , accurate linkages are statistically observable, hence effective EDAs are possible. However, in order to compute δ_H or δ_I the probability distribution of pairs (X, Y) (intra-block variables) and (X, Z) (inter-block variables) must be known beforehand. Section 3.2.1 addresses the analysis of additively separable functions regarding the measurement of δ_H and δ_I from infinite populations specifically built to evidence the linkage-learning benefits and its limitations.

3.2.1 Additively Separable Functions

Additive separable functions can be more easily solved by focusing on their components instead of attacking the whole problem. However, in black-box optimization, GAs do not know such components (linkages); it is the role of linkage-learning to identify them and the reproduction

operators should enable their proper use in the composition of better solutions (Goldberg, 2002).

For the purposes of the theoretical analysis described in this section, only additive separable functions with $\pi_i = \{x_i, x_{i+1}, \dots, x_{i+k-1}\}, \forall i \in \{1, 1+k, \dots, m-k\}$ were considered. The $f_i(\mathbf{x}, \pi_i)$ functions composing $f(\mathbf{x})$, Equation (3.3), are based in the sum of the values taken by the variables: $u(\mathbf{x}, \pi_i) = \sum_{j=1}^k x_{\pi_{ij}}$ (unitation). Each function uses $u(\mathbf{x}, \pi_i)$ differently, producing different optimization problems. Four well-known functions: k -trap, k -bipolar, k -

Figure 3.2: All possible outcomes of a trap function with $k = 5$ (a), a bipolar function with $k = 6$ (b), a parity function with $k = 6$ and a parity/trap function with $k = 6/k = 5$.

parity and k -parity/trap were chosen, Figure 3.2 illustrates their outcomes according to their specific use of the unitation values.

mk -traps The mk -trap problem consists of m concatenated k -traps. It contains one global optimum and one local optimum solution (where variables are all-ones and all-zeros, respectively) and is defined as the maximization of a function composed of m components of size k . The fitness of a whole solution $\mathbf{x}$ is given by the summation of the results of each k -trap. Each k -trap takes the number of ones (unitation u) in its respective set of k variables as an argument and

returns:

$$\text{trap}(u) = \begin{cases} k & \text{if } u = k, \\ k - 1 - u & \text{otherwise.} \end{cases}$$

Figure 3.2(a) illustrates the possible outcomes of $\text{trap}(u)$ for $k = 5$. Solutions uniformly generated at random are likely to be in a suboptimal local-attractor.

mk -bipolar A different situation is observed for the mk -bipolar problem with blocks of size $2k$. For a problem with $m = 5, k = 3$, there are 32 globally optimal solutions (combinations of all-one and all-zero components) and approximately three million locally optimal solutions. The k -bipolar function is defined as follows:

$$\text{bipolar}(u) = \begin{cases} 1 & \text{if } |u - k| = k, \\ 0.9 & \text{if } |u - k| = 0, \\ 0.8 & \text{if } |u - k| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Figure 3.2(b) shows the possible outcomes for a 6-bipolar function. Each component has two global optimum configurations and $\binom{6}{3}$ local optimum configurations. Section 3.3 show how such a difference impacts on linkage-learning and makes the mk -bipolar difficult to learn.

mk -parity Like mk -trap and mk -bipolar, Concatenated Parity Function (CPF) are also composed by the sum of m separable components. However, CPFs do not rely on deceptive functions; instead, it uses m parity functions, each one defined as follows:

$$\text{parity}(u) = \begin{cases} C_{\text{even}}, & \text{if } u \text{ is even,} \\ C_{\text{odd}}, & \text{otherwise, with } C_{\text{even}} > C_{\text{odd}}. \end{cases}$$

For each parity function of size k , there are 2^{k-1} global optimum configurations, with ones and zeros having the same probability distribution in a population. Since there are no observable pairwise dependencies, linkage-learning from bivariate statistics cannot identify the k -order dependencies in CPFs, as pointed out by Coffin and Smith (2007).

mk -parity/trap This function is composed of concatenated k -parity and k -trap functions. It mixes two different difficulties, i.e. the high multimodality of parity functions with easier bimodal traps. Figure 3.2(d) illustrates two concatenated sub-functions, each of them contributing independently to the whole fitness.

The configurations highlighted with circles in Figure 3.2 represent local and global optimum

solutions for each sub-function. After deterministic Hill-Climbing (dHC), solutions will be composed of combinations of such partial-solutions, i.e. a solution will be globally optimal only if each of its components is also globally optimal. In this scenario we know the marginal and joint probability distributions of random variables X , Y and Z , therefore δ_H and δ_I values can be estimated.

3.2.2 Population Size Assumption

To determine probability distributions from samples (population) is a difficult task (see Goldberg (2002)). Fortunately, for our purposes it is sufficient to analyze the situation considering infinite populations. Besides, we should focus on situations in which linkage-learning is essential for the search and not in the whole process of searching for solutions. Regarding additively separable deceptive functions, linkage-learning becomes essential from a very specific stage, which can be simulated by using hill-climbing.

Consider the case in which a number of generations has passed, i.e. a number of selection, crossover and mutation steps has been performed. At this stage all disjoint subsets of variables would take only optimal or locally optimal configurations and globally optimal solutions would only be found by appropriate mixing. Without knowledge about the components, crossover would not be effective in mixing, therefore the population average fitness would most probably stop to increase due to the disruptive effects of crossover and mutation. At such a stage, linkage-learning would highly influence the progress of the search by increasing crossover effectiveness. This is the situation in which we are interested to know the difficulty of learning the linkages.

In case of additively separable deceptive functions such a scenario can be simulated by applying deterministic Hill-Climbing (dHC) to the population. The initial population is created in two steps: (1) candidate solutions are generated uniformly at random, (2) each candidate solution undergoes dHC (Algorithm 3.2).

Algorithm 3.2 Deterministic hill-climbing (dHC).

Require: a solution $\mathbf{x}$

```

1: for all  $i$  in  $1, \dots, N$  do
2:    $f_{\mathbf{x}} \leftarrow f(\mathbf{x})$  ▷ Store the current  $f(\mathbf{x})$ 
3:    $x_i \leftarrow 1 - x_i$ 
4:   if  $f(\mathbf{x}) < f_{\mathbf{x}}$  then  $x_i \leftarrow 1 - x_i$  ▷ If  $\mathbf{x}$  has worsened, undo
   return  $\mathbf{x}_i$ 
```

The dHC decreases the intra-block entropy $H(X, Y)$ of the population, evidencing possible linkages. Moreover, for being deterministic, it produces a population with known marginal and joint distributions of variables X , Y and Z , that are necessary for the calculation of δ_H and δ_I values. The comparison of δ_H and δ_I values in functions with different degrees of multimodality enables the assessment of their relative linkage-learning difficulty, which can be used to confirm, or not, the influence of multimodality in linkage-learning accuracy. The use of dHC is not restrictive to the analysis as proved by Theorem 3.1.

3.3 Results and Discussions

This section describes the results regarding the difficulty of learning accurate linkage models for additively separable functions. The measurement of δ_H and δ_I values are performed on two different ways: (1) the probability distributions of several functions are analyzed and their difficulty compared; (2) such measurements are estimated on upper-bounds of the entropy whose values are also compared. Although upper-bounds are sufficient to elucidate the relation between multimodality and linkage-learning difficulty, we show they do not provide completely descriptive results, in the sense that subtle differences in difficulty are not evidenced. Section 3.3.1 analyses the exact δ_H and δ_I values, whereas Section 3.3.2 analyses their upper-bounds.

3.3.1 Linkage-learning Difficulty

Each block composed of k binary variables can initially take 2^k different configurations. In a random population, each of these configurations occurs with the same probability $1/2^k$. However, after dHC these probabilities change due to a decrease in the number of possible configurations for each pair of variables, such a decrease depends on the number of local- and global optima.

The marginal and joint probabilities also depend on the number of global and local-optima for each sub-function. Assume that G and L represent the events of global and local optima configurations. Then the probability distribution of X, Y, Z obeys the following equations; $x, y, z \in \{0, 1\}$:

$$p_{XY}(y|x) = p_{XY}(y|G \wedge x) + p_{XY}(y|L \wedge x) \quad (3.11)$$

$$p_X(x) = p_X(x|G) + p_X(x|L) \quad (3.12)$$

$$p_{XY}(x, y) = p_X(x)p_{XY}(y|x) \quad (3.13)$$

$$p_{XZ}(x, z) = p_X(x)p_Z(z) \quad (3.14)$$

In words, a random variable X may take the value x in a local- or global-optimum configuration, $p_X(x|G)$ and $p_X(x|L)$. Therefore, $p_X(x)$ is the sum of these two probabilities, as shown in Equation (3.12). Similarly, $p_{XY}(x, y)$ depends on the probability of X assuming the value x and the probability of Y assuming the value y given $X = x$, see Equation (3.13). At last, Equation (3.14) follows from the fact X and Z are, by definition, independent random variables.

From the general Equations (3.11)-(3.14), the probability distributions, after dHC, are calculated for all the functions previously defined.

k -trap: In a random population, the global optimal configuration for a k -trap function occurs with probability $1/2^k$. Since dHC is performed from the leftmost to the rightmost variable (from x_1 to x_N), only the configuration 01111 (for $k = 5$) will become optimal, therefore, global optima configuration occurs in two cases leading to a probability $G_{\text{trap}} = 2/2^k$. Inversely, the

probability of a local optima configuration is $L_{\text{trap}} = 1 - G_{\text{trap}}$ and the probability distributions for any $X, Y \in \pi_j$ are defined as follows

$$\begin{aligned} p_X(1) &= G_{\text{trap}} \times 1 + L_{\text{trap}} \times 0 = G_{\text{trap}} \\ p_X(0) &= G_{\text{trap}} \times 0 + L_{\text{trap}} \times 1 = L_{\text{trap}} \\ p_{XY}(1, 1) &= G_{\text{trap}} \times (1 + 0) = G_{\text{trap}} \\ p_{XY}(1, 0) &= G_{\text{trap}} \times (0 + 0) = 0 \\ p_{XY}(0, 1) &= L_{\text{trap}} \times (0 + 0) = 0 \\ p_{XY}(0, 0) &= L_{\text{trap}} \times (0 + 1) = L_{\text{trap}} \end{aligned}$$

Since only in the global optimum configuration $X = 1$, then $p_X(1) = G_{\text{trap}}$, the opposite occurring for $p_X(0) = L_{\text{trap}}$. By a similar reasoning, all $p_{XY}(x, y)$ are defined. In the k -trap, multimodality does not depend on k , there are always only one global and one local-optimum. The above distributions are derived from the dHC working principles. Next, we show this is not restrictive to the analysis by proving that a randomized Hill-Climbing (rHC) produce the same distribution of globally-optima configurations (similar conclusions can be derived for all the other functions considered in this chapter).

Theorem 3.1. *One pass of deterministic Hill-Climbing (dHC) or randomized Hill-Climbing (rHC) induces the same probability distribution over solutions of the mk -trap function.*

Proof. Consider $k = 5$ in a k -trap function. The globally optimum configuration for a block is randomly generated with probability $1/2^k$. Since the rHC evaluates variables in a random order, all configurations 01111, 10111, 11011, 11101, 11110 might become globally optima. However, this only occurs if the first variable evaluated by rHC is the one assuming the zero value. The probability of randomly generating these configurations is $(k/2^k)$, whereas the probability of picking the correct variable first is $(1/k)$. Therefore rHC produces additional global optima from them with probability $(k/2^k)(1/k) = 1/2^k$, which is the same of dHC. $\square$

k -bipolar: In this case, two global optima configurations can be randomly generated: 000000 and 111111 (for $k = 3$), and two configurations will become optimum after dHC, 100000 and 011111. Therefore, there would be four global optimum configurations leading to a probability $G_{\text{bipolar}} = 4/2^{2k}$, all other configurations would be local-optima, then $L_{\text{bipolar}} = 1 -$

g_{bipolar} . Note that global optima configurations contains only ones or only zeros, while local optima configurations always contain three variables taking the value “one”. Therefore, in global optima configurations, if $X = 1$ then $p_{XY}(y = 1|x = 1) = 1$. In local optima configurations, if $X = 1$ then $p_{XY}(y = 1|x = 1) = (k - 1)/(2k - 1)$. From these considerations, the marginal and joint probability distributions are defined as follows

$$p_X(x = 1) = p_X(x = 0) = G_{\text{bipolar}} \left(\frac{1}{2} \right) + L_{\text{bipolar}} \left(\frac{k}{2k} \right)$$

$$\begin{aligned}
&= \frac{G_{\text{bipolar}}}{2} + \frac{L_{\text{bipolar}}}{2} = \frac{1}{2}, \\
p_{XY}(x=1, y=1) &= \frac{G_{\text{bipolar}}}{2} p_{XY}(y|G_{\text{bipolar}} \wedge x) + L_{\text{bipolar}} \left(\frac{k}{2k} \right) p_{XY}(y|L_{\text{bipolar}} \wedge x) \\
&= \frac{G_{\text{bipolar}}}{2} + \frac{L_{\text{bipolar}}}{2} \left(\frac{k-1}{2k-1} \right), \\
p_{XY}(x=1, y=0) &= \frac{L_{\text{bipolar}}}{2} \left(\frac{k}{2k-1} \right), \\
p_{XY}(x=0, y=1) &= p_{XY}(1, 0) \\
p_{XY}(x=0, y=0) &= p_{XY}(1, 1)
\end{aligned}$$

For a k -bipolar function, the number of global and local-optima grows as $2 + \binom{2k}{k}$.

k -parity: For this function all configurations with even unitation are global optima. In other words, for a block of size k , there are 2^{k-1} possible optimum configurations after dHC, therefore, $G_{\text{parity}} = 1$ and $L_{\text{parity}} = 0$. Considering even unitations, $p_X(1)$ is defined by summing the probabilities for unitations $u = 2i$, with $i = 0, 1, \dots, \lfloor \frac{k}{2} \rfloor$, which converge to

$$\begin{aligned}
p_X(1) &= \frac{1}{2^{k-1}} \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{2i} \frac{2i}{k} = \frac{1}{2} \\
p_X(0) &= \frac{1}{2^{k-1}} \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{k-2i} \frac{k-2i}{k} = \frac{1}{2} \\
p_{XY}(1, 1) &= \frac{1}{2} \sum_{i=1}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{2i} \frac{2i}{k} = \frac{1}{4} \\
p_{XY}(1, 0) &= \frac{1}{2} \sum_{i=1}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{k-2i} \frac{k-2i}{k} = \frac{1}{4} \\
p_{XY}(0, 1) &= p_{XY}(1, 0), \quad p_{XY}(0, 0) = p_{XY}(1, 1)
\end{aligned}$$

Note that all series are derived analogously to $p_X(1)$. For the k -parity, the multimodality depends on k and, as k increases, the number of global and local-optimum grows according to 2^{k-1} .

k -parity/trap: The probability distribution for this concatenated function is defined from the distributions of its components, described above.

Such benchmark functions represent different levels of multimodality. By analyzing them in terms of conditions (3.7)-(3.8), the influence of multimodality on the difficulty of linkage-learning can be investigated in terms of the δ_H and δ_I values. Large δ_H (δ_I) values for an additively separable function imply that it can be easily modeled. Next, δ_I values for all

functions previously described are derived, assuming an specific k . Afterwards, different values for k are evaluated and their influence on the learning difficulty of those functions is evaluated.

To compute δ_I , values $H(X, Y)$ and $H(X, Z)$ are required for all functions (see Definition 3.5). They can be obtained from the probability distributions, as follows:

δ_{I_t} **(6-trap)**: Considering $k = 6$, $G_{\text{trap}} = 2/2^6 = 0.03125$ and $L_{\text{trap}} = 0.96875$, one has:

$$\begin{aligned}
 H(X, Y) &= -G_{\text{trap}} \log_2 G_{\text{trap}} - L_{\text{trap}} \log_2 L_{\text{trap}} = 0.2006223 \\
 I(X; Y) &= H(X) + H(Y) - H(X, Y) = 0.2006223 \\
 H(X, Z) &= -G_{\text{trap}}^2 \log_2(G_{\text{trap}}^2) - 2G_{\text{trap}}L_{\text{trap}} \log_2(G_{\text{trap}}L_{\text{trap}}) - L_{\text{trap}}^2 \log_2(L_{\text{trap}}^2) \\
 &= 0.4012446 \\
 I(X; Z) &= H(X) + H(Z) - H(X, Z) \\
 &= 0.2006223 + 0.2006223 - 0.4012446 = 0 \\
 \delta_{I_t} &= I(X; Y) - I(X; Z) = \mathbf{0.2006223}
 \end{aligned}$$

δ_{I_b} **(3-bipolar)**: Considering $k = 3$, $G_{\text{bipolar}} = 4/2^6 = 0.0625$ and $L_{\text{bipolar}} = 0.9375$, one has:

$$\begin{aligned}
 H(X, Y) &= -2p_{XY}(1, 1) \log_2 p_{XY}(1, 1) \\
 &\quad - 2p_{XY}(1, 0) \log_2 p_{XY}(1, 0) = 1.989727 \\
 I(X; Y) &= H(X) + H(Y) - H(X, Y) \\
 &= 1 + 1 - 1.989727 = 0.010273 \\
 H(X, Z) &= -2p_X(1)^2 \log_2 p_X(1)^2 \\
 &= -2p_X(1)p_Z(0) \log_2 p_X(1)p_Z(0) = 2 \\
 I(X; Z) &= H(X) + H(Z) - H(X, Z) \\
 &= 1 + 1 - 2 = 0 \\
 \delta_{I_b} &= I(X; Y) - I(X; Z) = \mathbf{0.010273}
 \end{aligned}$$

δ_{I_p} **(6-parity)**: Considering $k = 6$, $G_{\text{parity}} = 1$ and $L_{\text{parity}} = 0$, one has:

$$\begin{aligned}
 H(X, Y) &= -2p_{XY}(1, 1) \log_2 p_{XY}(1, 1) \\
 &\quad - 2p_{XY}(1, 0) \log_2 p_{XY}(1, 0) = 2 \\
 I(X; Y) &= H(X) + H(Y) - H(X, Y) = 0 \\
 H(X, Z) &= -2p_X(1)^2 \log_2 p_X(1)^2 \\
 &\quad - 2p_X(1)p_Z(0) \log_2 p_X(1)p_Z(0) = 2 \\
 I(X; Z) &= H(X) + H(Z) - H(X, Z) = 0 \\
 \delta_{I_p} &= I(X; Y) - I(X; Z) = \mathbf{0}
 \end{aligned}$$

$\delta_{I_{pt}}$ (**6-parity/trap**): Considering $k = 6$, X, Y variables from a trap function and Z a variable from a parity function. As two different functions are considered, we use δ_I

$$\begin{aligned}
 I(X; Y) &= H(X) + H(Y) - H(X, Y) = 0.2006223 \\
 H(X, Z) &= -p_X(0)p_Z(0)\log_2 p_X(0)p_Z(0) \\
 &\quad - p_X(0)p_Z(1)\log_2 p_X(0)p_Z(1) \\
 &\quad - p_X(1)p_Z(0)\log_2 p_X(1)p_Z(0) \\
 &\quad - p_X(1)p_Z(1)\log_2 p_X(1)p_Z(1) = 1.200622 \\
 I(X; Z) &= H(X) + H(Z) - H(X, Z) \\
 &= 0.2006223 + 1 - 1.200622 = 0 \\
 \delta_{I_{pt}} &= I(X; Y) - I(X; Z) = \mathbf{0.2006223}
 \end{aligned}$$

Comparing the δ_I values the difficulty in learning each of these functions is evident. The separability of the 6-trap is easy to learn from an appropriate population, i.e. variables X and Z will be wrongly considered as statistically dependent with a low probability. In fact, from the perspective of learning, k -trap is one of the easiest functions; after dHC each of the sub-functions take only two different configurations, leading to a low intra-block entropy $H(X, Y)$.

For k -bipolar, on the other hand, such a decrease is very different. Since there are $\binom{2k}{k}$ possible configurations for the k decision variables, even after dHC δ_{I_b} will be very small making difficult to identify the linkages with a certain confidence. Obviously, in practice, there is more than one opportunity to obtain an accurate linkage model, besides, many of the possible configurations for variables X and Y might disappear from the population due to selection, which could lead to an additional decrease of intra-block entropy.

The worst situation occurs for the k -parity function. Half of the 2^k configurations are possible after dHC and $\delta_{I_p} = 0$, which demonstrates the *allelic-pairwise independence* of intra-block variables noticed by Coffin and Smith (2007). From the perspective of linkage-learning based on pairwise dependencies, learning the k -order dependencies of the parity function is infeasible.

The $\delta_{I_{pt}}$ for 6-parity/trap function confirms the results for the 6-parity and the 6-trap. In fact, it is easy to learn that variables from a trap function are dependent and variables from parity functions are pairwise-independent, which makes $\delta_{I_{pt}} = \delta_{I_t}$. Instead, if we had chosen X and Y from parity and Z from trap, then $\delta_{I_{pt}} = \delta_{I_p}$.

Figure 3.3(a) illustrates how δ_I values behave according to k . A large δ_I means that it is easy to correctly identify linked variables. In summary, the difficulties in learning trap functions and parity/trap functions are the same, while the bipolar is harder to learn and the parity linkage-learning is infeasible ($\delta_I = 0$). For k -bipolar, as k grows, the number of configurations implying $\text{bipolar}(u) = 0$ also grows, this fact decreases the accuracy of our model, as shown in Figure 3.3(a). However, for $k = 3$ (the most common value used in literature) the model and results agree, therefore the discrepancy shown do not affect our conclusions.

As the search advances, some configurations might be lost, which decreases the intra-block entropy $H(X, Y)$ and increases the mutual information $I(X; Y)$. If all locally optimal configurations have the same fitness contribution, this loss of diversity does not hamper the search, i.e. some configurations are lost but the global optimal can still be found by *mixing*. However, in a general context, pairwise independence can impose serious difficulties for EDAs.

Figure 3.3: Figure (a) shows that accurate linkage-learning from mutual information is harder for larger values of k (small δ_I values). Figure (b), on the other hand, shows that δ_H for the k -trap is stable as k grows, whereas the k -bipolar is not (due to the increase of multimodality according to k).

Figure 3.3(b) shows δ_H for all functions except parity/trap. Since δ_H is normalized we can see how k affects the difficulty of learning. As k -traps always have the same degree of multimodality, their difficulty does not increase with k . For k -bipolar and k -parity, on the other hand, the difficulty increases and relatively fast reaches $\delta_H \approx 0$. For the bipolar function, experimental values slightly differ from theoretical values. This fact occurs due to an unexpected characteristic of such functions, in which the number of plateau solution increases along with k .

The stability of δ_H corroborates the use of normalized measures of information during linkage-learning, a subject discussed by Bosman and Thierens (2012b).

3.3.2 Upper-bounds on Linkage-learning Difficulty

This section shows, at the expense of some accuracy, that entropy upper-bounds can be used to estimate δ_I and δ_H , which facilitates the analysis. Although such approximations do not enable us to observe small differences, all main conclusions of our concern can be drawn from them (Martins and Delbem, 2014).

Upper-bounds can be obtained by Jensen (1906)'s inequality, which is employed as follows:

$$\left[H(X) = \sum_{x \in \mathcal{S}_X} p_X(x) \log_2 \left(\frac{1}{p_X(x)} \right) \right] \leq \log_2(|\mathcal{S}_X|)$$

where the right-hand side is an upper-bound for the entropy. Therefore, to compute such upper-bound we need to know only the size of the support set of (X, Y) and (X, Z) (Definition 3.1), from which the result is easily obtained. Next, we compute δ_I values for the mk -parity/trap and δ_H values for all the other functions. To evidence they are approximations of δ_I and δ_H , we use the notation $\tilde{\delta}_H$ and $\tilde{\delta}_I$.

$\tilde{\delta}_{H_t}$ (**k -trap**) In mk -trap functions, after dHC has been performed, only two configurations remain in the population, e.g. blocks 000000 and 111111 ($k = 6$), therefore, all other $2^k - 2$ configurations have zero probability of occurrence, i.e. $|\mathcal{S}_{XY}| = 2$. The application of Jensen's inequality to intra-block variables (X, Y) , results in

$$H(X, Y) \leq \lceil \log_2(2) = 1 \rceil.$$

Concerning inter-block variables (X, Z) , all pairs of values can occur, therefore, $|\mathcal{S}_{XZ}| = 4$ and Jensen's inequality results in the upper-bound

$$H(X, Z) \leq \lceil \log_2(4) = 2 \rceil.$$

Using these values $\tilde{\delta}_{H_t}$ can be computed,

$$\tilde{\delta}_{H_t} = 1 - H(X, Y)/H(X, Z) = \mathbf{0.5},$$

which, in fact, equals the exact δ_{H_t} value (Martins and Delbem, 2014). However, the conclusions are very different if 6-bipolar functions are considered, as shown next.

$\tilde{\delta}_{H_b}$ (**k -bipolar**) In this case, there are two global optima and $\binom{6}{3} = 20$ local optima for intra-block configurations. Since (X, Y) can assume any pair of values, $|\mathcal{S}_{XY}| = 4$. Inter-block variables (X, Z) can also assume any pair of values, therefore, $|\mathcal{S}_{XZ}| = 4$ and

$$(H(X, Y) = H(X, Z)) \leq \lceil \log_2(4) = 2 \rceil,$$

Using these values $\tilde{\delta}_{H_b}$ can be computed,

$$\tilde{\delta}_{H_b} = 1 - H(X, Y)/H(X, Z) = \mathbf{0}.$$

In this case, $\tilde{\delta}_{H_b} = 0$ only approximates the exact value which is $\delta_{H_b} = 0.005$. However, the great difference between $\tilde{\delta}_{H_t}$ and $\tilde{\delta}_{H_b}$ confirms a well-known result, in which mk -bipolar is considered harder for linkage-learning than mk -traps (Goldberg, 2002).

$\tilde{\delta}_{H_p}$ (**k -parity**) The parity function is in the limit of separability, i.e. although there are well-defined blocks, they are not detectable from low-order statistics. There are 2^{k-1} different possible

configurations (all optima, with even unitation) and no decrease in the intra-block entropy after dHC. Therefore, there is no *mutual information* or ways to distinguish between statistically dependent and independent variables. Using Jensen's inequality, the joint entropy between any pair of variables and the $\tilde{\delta}_{H_p}$ are approximated by

$$\begin{aligned} (H(X, Y) = H(X, Z)) &\leq \lceil \log_2(4) = 2 \rceil, \\ \tilde{\delta}_{H_p} &= 1 - H(X, Y)/H(X, Z) = 0. \end{aligned}$$

Thus, there is no observable pairwise dependencies between intra-block variables. However, it does not mean there are no higher-order linkages. For example, assume $k - 1$ dependencies of a block of size k are already known and referred as a set A . It must be decided which other variable is also statistically dependent to variables in A , assume such variable is Y .

By the definition of parity functions, the cardinality of the support set according to the correct linkage set $A \cup Y$ is $|\mathcal{S}_{A \cup Y}| = 2^{k-1}$, which leads to an upper-bound

$$H(A \cup Y) \leq \lceil \log_2(2^{k-1}) = k - 1 \rceil.$$

On the other hand, if a wrong variable Z had been chosen to compose a linkage set $A \cup Z$, there would be configurations with both even and odd unitations. Since $A \cup Y$ can only contain configurations with even unitations, $|\mathcal{S}_{A \cup Z}| > |\mathcal{S}_{A \cup Y}|$. By assuming a $\epsilon > 0$, the size of the support set of $A \cup Z$ can be defined as $|\mathcal{S}_{A \cup Z}| = 2^{k-1+\epsilon}$, then

$$H(A \cup Z) \leq \lceil \log_2(2^{k-1+\epsilon}) = (k - 1 + \epsilon) \rceil.$$

Since $H(A \cup Y) < H(A \cup Z)$ in this case, $\tilde{\delta}_{H_p}$ would be greater than zero. This result shows there are k -order statistical dependencies that cannot be observed from bivariate statistics.

$\tilde{\delta}_{I_{pt}}$ (**k -parity/trap**) The concatenated parity/trap function is very similar to the trap in terms of linkage-learning difficulty. Assuming X, Y from a trap and Z from a parity function, there would be $|\mathcal{S}_{XY}| = 2$ possible configurations for (X, Y) , therefore,

$$H(X, Y) \leq \lceil \log_2(2) = 1 \rceil.$$

Since inter-block variables can assume any pair of values, the support set has size $|\mathcal{S}_{XZ}| = 4$ and the joint entropy is

$$H(X, Z) \leq \lceil \log_2(4) = 2 \rceil.$$

In this case, the sub-functions are different (k -trap and k -parity), therefore, δ_I must be used instead of δ_H . Using Jensen's inequality, and considering the support sets of size $|\mathcal{S}_Y| = 2$ and

$|\mathcal{S}_Z| = 2$, there are entropies $H(Y) \leq 1$ and $H(Z) \leq 1$, therefore

$$\begin{aligned}\tilde{\delta}_{I_{pt}} &= I(X; Y) - I(X; Z), \\ \tilde{\delta}_{I_{pt}} &= H(Y) - H(X, Y) - H(Z) + H(X, Z) = 1.\end{aligned}$$

Since $\tilde{\delta}_{I_{pt}}$ is maximum, it can be concluded it is easy to identify statistically dependent variables from trap functions. However, if X and Y are from a parity function and Z from a trap function, the opposite situation would have occurred, with $H(X, Y) = 2$ and $H(X, Z) = 2$. Therefore $\tilde{\delta}_{I_{pt}} = 0$, representing the difficulty of learning parity functions. In summary, in a problem with parity and trap sub-functions, each of them keep their original linkage-learning difficulty, k -trap variables are detected as linkages while k -parity variables still cannot be identified due to the allelic-pairwise independence.

3.4 Conclusions

This chapter addressed the use of concepts from information theory to analyze the limitations of linkage-learning. It was argued that an increasing multimodality can lead to an observable allelic-pairwise independence even in additively separable functions.

Using mutual information and joint entropy as reference, the necessary condition which makes linkage-learning from bivariate dependencies feasible were defined. This condition considers the degree of interactions among intra-block and inter-block variables, according to the probability distributions observed in the population. It was shown that additive separable functions with different degrees of multimodality possesses different degrees of separability, measured by δ_H and δ_I values.

Different benchmark functions were compared according to their linkage-learning difficulty, the results clearly showed that the k -trap is the easiest function for linkage-learning, followed by k -parity/trap and k -bipolar. It was also confirmed that the learning of k -parity functions from bivariate statistics is infeasible due to its intrinsic allelic-pairwise independence. Pairwise independence between decision variables may be very common in binary search spaces, where, besides multimodality, overlapping interactions influence the observable statistics.

In a broader perspective, EDAs may have serious limitations on their capacity of learning in binary optimization problems. Fortunately, as in the case of parity functions, there must be ways to circumvent such difficulties. Hybridization or use of linkage-guided crossover must be taken into account in order to guarantee that EDAs work when linkage-models are not reliable. EDAs like the Linkage Tree Genetic Algorithm (LTGA) (Thierens, 2010), already exploits such an idea and, even when inaccurate linkage-models are available, the search can still progresses. In the next chapter these ideas are extended to verify experimentally the limits of linkage-learning in non-separable functions.

Chapter 4

Multimodality and the Limits of Linkage-learning

Chapter 3 showed multimodality may considerably affect the difficulty of observing statistical dependencies between random variables that should indeed be seen as dependent in additively separable functions. Such a difficulty implies limitations for the accuracy of linkage models and restricts the usefulness of linkage-learning to problems whose multimodality degree does not exceed a certain limit. This chapter evaluates the validity of the theoretical analysis in a broader scenario, considering non-separable instances of the NK -landscapes model with different degrees of multimodality (defined by K).

4.1 Introduction

Studies concerning the difficulties and limits of linkage-learning emerged with Coffin and Smith (2007) in the context of additively separable Concatenated Parity Functions (CPFs). The authors have shown that Bayesian Optimization Algorithm (BOA) (Pelikan, 2005), one of the main Estimation of Distribution Algorithms (EDAs) by that time, scaled exponentially in the size N of the CPFs, an unexpected result for the community. They argued that any linkage-learning method based on bivariate statistics would not succeed in finding the high-order linkages of the CPFs due to the absence of pairwise dependencies between decision variables, a characteristic they called *allelic-pairwise independence*. Since most EDAs rely on bivariate statistics to implement model-building, this fact brings many concerns about the limitations of EDAs. Many studies regarding EDAs and their efficiency in exploiting substructures to improve the performance of the search have been reported along the years. Here, the question is how likely such substructures are observed in more practical optimization problems, or, in the opposite way, how likely is for *allelic-pairwise independence* to occur and linkage-learning become less useful.

Chapter 3 is a step further in understanding *allelic pairwise independence*, in which the necessary (but not the sufficient) conditions for accurate linkage-learning from bivariate statistics

were demonstrated. The results showed that highly multimodal functions tend to be harder for linkage-learning and that *allelic pairwise independence* may occur as consequence of massive multimodality. Unfortunately the theoretical analysis of additively separable functions cannot be extended for the more general case of non-separable functions and a new approach must be taken towards generalization.

This chapter evaluates the role of multimodality on the linkage-learning difficulty of non-separable functions. The NK -landscapes model provides a simple way to assess the impact of multimodality on the benefits of using linkage-learning. If K is increased, a decrease on the efficacy of EDAs is expected and this chapter verifies such assumption experimentally. Section 4.2 describes the method used to assess the usefulness of linkage-learning according to multimodality and define the experiment settings. Section 4.3 describes the results in terms of effectiveness, efficiency and the usefulness of linkage-learning.

4.2 Methodology

The benefits of EDAs are usually verified by comparison with more traditional Genetic Algorithms (GAs). However, to identify the range of usefulness of linkage-learning, the experiments must be designed in such a way to isolate (as well as possible) the benefits of using linkages from the effects of other operators. Therefore, instead of drawing conclusions from the comparison of EDAs and GAs, the experiments regard the comparison of linkage-models and random linkage-models in the context of the same EDA: (1) the Linkage Tree Genetic Algorithm (LTGA) and (2) the Random Linkage Tree Genetic Algorithm (rLTGA), which uses random linkage-tree models.

If linkage-learning is assumed as an effective tool for any kind of optimization problems, the LTGA should never be outperformed by the rLTGA. This chapter assumes this is not the case, instead, it follows the results of Chapter 3 which suggests the usefulness of linkage-learning would decrease as multimodality increases. Therefore, by verifying the degrees of multimodality in which LTGA significantly outperforms the rLTGA, it is possible to identify the limits of usefulness for linkage-learning. Such an hypothesis is grounded on the theoretical results for additively separable functions, but the comparison of LTGA and rLTGA enables the evaluation of general optimization problems.

The comparison between LTGA and rLTGA must be made in a context where multimodality can be precisely estimated, the NK -landscapes model is an appropriate choice for this purpose. It generates instances of non-separable pseudo-Boolean functions with tunable dimension (N) and ruggedness/multimodality (K). Moreover, for certain configurations, the NK -model produces NP-hard optimization problems closely related to practical optimization problems. Therefore, such a comparison enables the assessment of the validity of the hypothesis developed theoretically in Chapter 3 in a much more realistic scenario.

4.2.1 The NK-landscapes Model

The NK -landscapes (Kauffman, 1993) is a well-known model for binary unconstrained pseudo-Boolean functions with parameterized difficulty (ruggedness, multimodality). In general, they are defined in N -dimensional binary search spaces $\mathbf{x} \in \mathcal{G} = \{0, 1\}^N$, where each decision variable x_i , $i = 1, \dots, N$, interacts with K other variables $\pi(x_i)$:¹

$$\pi(x_i) = \{x_j : j \in \{1, \dots, N\} \wedge j \neq i\}.$$

Instances of the NK -landscapes can be randomly generated for fixed parameters N and K , as follows: (1) for each variable x_i , a set $\pi(x_i)$ of size K is randomly chosen from $\{x_1, \dots, x_N\}$. The subset of variables $x_i \cup \pi(x_i)$ has size $K + 1$, therefore, in the binary case, it can take 2^{K+1} different configurations; (2) for each configuration, a fitness contribution $f_i(x_i, \pi(x_i))$ is randomly chosen within the $(0, 1]$ interval. The average contribution of all variables provides the objective value $f_{NK}(\mathbf{x})$ of a candidate solution $\mathbf{x} \in \mathcal{G}$. The optimization of a NK problem consists in finding the maximum

$$\max_{\mathbf{x}} f_{NK}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N f_i(x_i, \pi(x_i)) \quad (4.1)$$

According to the aspects discussed in this chapter, K is the most important parameter, as it defines the non-linearity degree (number of epistatic interactions among decision variables) that can be used to increase the multimodality and the ruggedness of the fitness landscape. In summary, as K grows, the instances produced become harder to solve and the effectiveness in finding global optima solutions should decrease considerably (Kauffman, 1993). Since we have associated the linkage-learning difficulty of optimization problems with multimodality, we expect the performances of LTGA and rLTGA to become more similar as K grows.

Another relevant aspect commonly explored in the literature is the choice of linked variables. In some studies the set $\pi(x_i)$ is not randomly generated; instead it is chosen from consecutive variables starting from x_i , i.e. $\pi(x_i) = \{x_{i+1}, \dots, x_{i+K}\}$ (Kauffman, 1993; Pelikan, 2010). These two approaches are commonly referenced as NK -landscapes with *random linkages* and NK -landscapes with *nearest-neighbors* linkages, respectively. Instances with *nearest-neighbors* linkages have shown similar characteristics to random linkages, but they have deterministic exact algorithms to find global optimal solutions for small N , which is useful for experimentation. However, as originally showed by Kauffman (1993), *nearest-neighbors* produce easier instances than *random linkages*, as evidenced by the results in the next sections.

¹ K in the NK -model influences multimodality and epistasis, differently of the k used in Chapter 3, which defines only epistasis.

4.2.2 Settings of the Experiments

To compare the performances of LTGA and rLTGA as multimodality increases, NK -landscape instances with $K = 2, 5, 10, 15$ are evaluated ($K \approx 5$ is the usual value found in the literature regarding EDAs). The influence of problem size is also evaluated, with $N = 100, 500, 1000$, and population sizes $\mu = 100, 500, 1000$. The following algorithms are compared:

1. GA (with uniform crossover and mutation);
2. LTGA;
3. LTGA+ (LTGA with mutation);
4. rLTGA (LTGA with random linkage-models)
5. rLTGA+ (rLTGA with mutation).

In all cases, the standard mutation is employed; besides, the insertion of duplicate solutions in the population was prohibited so that the diversity of solutions could be maintained (preliminary experiments showed it only improves performance in long-runs).

For each triple (K, N, μ) , the algorithms were allowed to perform $\theta = N \times 10^3$ fitness evaluations. Since θ does not depend on μ , it can be evaluated if the LTGA performs better with large or small populations. In each configuration, one hundred runs were executed with different seeds. The resulting samples were compared by the Wilcoxon-Mann-Whitney hypothesis test, concerning differences between LTGA and rLTGA average performances. We say the LTGA has outperformed the rLTGA if $p\text{-value} < \alpha$, with $\alpha = 0.001$.²

Although five algorithms were evaluated, we provide p-values only for the comparison between the LTGA and the rLTGA. The other algorithms were included because the GA serves as a standard reference for performance and the LTGA+ and rLTGA+ serve as a reference for the influence of mutation. We seek for a common effectiveness pattern for the LTGA and the rLTGA to estimate the range of K in which linkage-learning is useful. Results for all pairs (K, N) are described for each $\mu = 100, 500, 1000$, according to the following criteria: (1) Which algorithm found the best solution: LTGA or rLTGA? (2) How did mutation impact on the results? (3) How did population-size impact on the results?

4.3 Results and Discussion

4.3.1 Effectiveness

This section describes the results for different instances of the NK -model. The p-values obtained from one-sided Wilcoxon-Mann-Whitney hypothesis tests are shown above each figure and a symbol $\dagger$ is written in the respective caption if $p\text{-value} < \alpha$, which means the LTGA outperformed the rLTGA with statistical significance.

²The p-values were rounded in a maximum of six decimal places and are shown above each figure.

Figure 4.1 summarizes the results for a population size $\mu = 100$. For the smaller instances ($N = 100$), shown in Figures 4.1(a)-4.1(d), the LTGA significantly outperformed the rLTGA only for $K = 5$, but both LTGA and rLTGA found considerably better solutions than GA. Mutation had positive effects only in low multimodality ($K = 2, 5$). The GA was the fastest algorithm to reach high-quality solutions, however, it also suffered from premature convergence where solution improvements ceased to occur. For the medium-sized instances ($N = 500$), shown in Figures 4.1(e)-4.1(h), the LTGA outperformed the rLTGA in all cases, although with a small margin in highly multimodal instances ($K = 10, 15$). The positive effects of mutation ceased to occur for large K and exerted a worse effect on the rLTGA rather than on the LTGA. The GA reached high-quality solutions very fast but ceased to progress. For highly multimodal instances ($K = 10, 15$), GA's results were similar to those achieved by the other algorithms. For the largest instances ($N = 1000$), shown in Figures 4.1(i)-4.1(l), the pattern of the results was very similar, i.e. the LTGA outperformed the rLTGA in all cases. Mutation impacted positively for $K = 2, 5$ but it lost effectiveness in high multimodality ($K = 10, 15$). The GA behaved similarly as in the previous case ($N = 500$), but its results were much closer to the LTGA results for $K = 10, 15$. The LTGA lost some power as N scaled, which did not occur with the GA.

Figure 4.1: Average trajectory of the best fitness of each algorithm along $\theta = N \times 10^3$ fitness evaluations. All experiments with $K = 2, 5, 10, 15$ and $N = 100, 500, 1000$ were performed with a fixed $\mu = 100$.

In general, for a population size $\mu = 100$, the benefits of using linkage-learning and mutation decreased considerably with the multimodality increase (K). While the GA performed stably and mostly independently of problem size, all LTGA variants showed a decrease in the solution quality as the problem size increased (i.e. their advantage in relation to the GA decreased). Such a decrease was more evident in highly multimodal instances ($K = 10, 15$), as shown in Figures 4.1(d), 4.1(h), 4.1(l). Another interesting fact is the difference between evolutionary patterns of GA and LTGAs; the first relied on small improvements while the second relied on large jumps of fitness to evolve solutions.

Figure 4.2: Average trajectory of the best fitness of each algorithm along $\theta = N \times 10^3$ fitness evaluations. All experiments with $K = 2, 5, 10, 15$ and $N = 100, 500, 1000$ were performed with a fixed $\mu = 500$.

Figure 4.2 summarizes the results for a population size $\mu = 500$. For the smaller instances ($N = 100$), shown in Figures 4.2(a)-4.2(d), the LTGA outperformed the rLTGA only in low multimodality ($K = 2, 5$) and lost its benefits in highly multimodal instances ($K = 10, 15$). Mutation did not exert positive effects even for small K and both LTGA+ and rLTGA+ found low-quality solutions. The GA had difficulties in finding high-quality solutions in highly multimodal instances ($K = 10, 15$). For the medium-sized instances ($N = 500$), shown in Figures 4.2(e)-4.2(h), the LTGA outperformed the rLTGA, except for $K = 15$. Mutation, again, did not provide positive results. The great surprise was the performance of the GA, which surpassed all LTGA

variants for $K = 5, 10, 15$. For the largest instances ($N = 1000$), shown in Figures 4.2(i)-4.2(l), the same situation occurred, i.e. the LTGA outperformed the rLTGA in all cases where $K < 15$. Mutation was of no help for $K > 2$ and the GA was the best overall for $K > 2$.

In general, for a population size $\mu = 500$, mutation did not help the search, except in the easier instances ($K = 2$). Although the LTGA outperformed the rLTGA in most cases (except when $K = 15$), both algorithms found very low-quality solutions in comparison to the results of GA. The LTGA showed again its stairway-like evolutionary fitness pattern, but due to the increase in the population size, the time required for improvements in the best solution (“jumps”) also increased. When such “jumps” did not occur, the LTGA performed very poorly, as shown in Figures 4.2(c) and 4.2(d).

From the results showed so far, it seems reasonable to conclude the LTGA variants do not work well with large populations, which is a surprising fact due to the common requirements of other EDAs Yu et al. (2007). In other words, with a limited number of fitness evaluations ($\theta = N \times 10^3$ in this case), the LTGA performs better with small populations. Regarding the GA, it was less sensitive to the increase in the population size and found solutions of similar quality in both configurations $\mu = 100, 500$, but achieved the best overall results with $\mu = 500$.

Figure 4.3: Average trajectory of the best fitness of each algorithm along $\theta = N \times 10^3$ fitness evaluations. All experiments with $K = 2, 5, 10, 15$ and $N = 100, 500, 1000$ were performed with a fixed $\mu = 1000$.

Figure 4.3 summarizes the results for a population size $\mu = 1000$. As expected from the previous results, the relative performance of the LTGA and rLTGA was the worst. First of all, there was no case in which both algorithms differed significantly and the GA outperformed all LTGA variants by a considerable margin. As in the experiments with $\mu = 500$, mutation did not help the search in any configurations, i.e. LTGA+ and rLTGA+ behaved similarly and worse than versions without mutation. We may argue that, with large populations, there is no need for additional diversity, therefore instead of improving the search mutation only slowed down the progress. The LTGA's stairway-like pattern of fitness increase did not appear, consequently the LTGA variants did not found high-quality solutions.

The results described so far suggest linkage-learning and mutation decrease their effectiveness as multimodality grows. They have also shown that, differently from other EDAs, the LTGA does not seem to benefit from using a large population; instead, its performance clearly deteriorates with larger populations. The next section describes results concerning the efficiency (running-time) of the algorithms. The conjunction of both effectiveness and efficiency should provide reliable indicators to estimate the range of usefulness of linkage-learning.

4.3.2 Efficiency

This section reports the results for the same instances of the previous section, but now regarding the efficiency of the algorithms in reaching their best solutions. A symbol $\ddagger$ has been added to the respective caption in experiments whose p-value $< \alpha$ (i.e. the LTGA outperformed the rLTGA). According to previous results, where a $\dagger$ symbol represented the LTGA found better solutions, now a pair of symbols $\dagger\ddagger$ means the LTGA found better solutions faster than the rLTGA. Additionally, the boxes where p-value $\geq \alpha$ have been colored in red to facilitate the identification of experiments in which the LTGA performed poorly. Since the LTGA was not effective with a population size $\mu = 1000$, the efficiency results for this case have been suppressed.

Figure 4.4 shows the distribution of the first-hitting³ time of each algorithm for a population size $\mu = 100$. For smaller instances ($N = 100$), shown in Figures 4.4(a)-4.4(d), it was difficult for the LTGA to be faster than the rLTGA (except for $K = 2$). However, such slow-down was expected, since building a linkage-tree is more expensive than building a random linkage-tree. It took the LTGA variants with mutation longer to find their final solutions, which was also expected, due to the additional diversity imposed by mutation. The GA was the most unstable among the algorithms and its first-hit time distribution showed a large standard deviation. For medium-sized instances ($N = 500$), shown in Figures 4.4(e)-4.4(h), the results are better.

The LTGA was the fastest with $K = 2, 10$, hence the most effective and efficient in these instances (note the $\dagger\ddagger$ in the caption). Such an increase in the problem size (N) led to a considerable reduction of variance in the LTGA and rLTGA results. GA and LTGA+ were very unstable with small and large standard deviations in different situations. Mutation exerted a

³The first-hitting time marks the moment in which the best solution was found.

much more subtle effect on the rLTGA than on the LTGA. For the largest instances ($N = 1000$), shown in Figures 4.4(i)-4.4(l), the situation was even better. The LTGA outperformed the rLTGA in all cases and found better solutions faster (note the †† in the captions). LTGA+ and GA were the most unstable and showed large variances in some cases.

Figure 4.4: Distributions of the first-hitting time with $K = 2, 5, 10, 15$, $N = 100, 500, 1000$ and $\mu = 100$.

In general, for a population size $\mu = 100$, the LTGA was better in large instances ($N = 500, 1000$). In these cases, large linkage-trees are produced, which also lead to large Family of Subsets (FOS). As the number of subsets in an FOS grows, the number of fitness evaluations performed by the LTGA variants in each generation also increases. As a consequence, within the limit of θ fitness evaluations, the number of generations decreases and fewer model-building steps are performed, which lead to a decrease in the running-time. The results in terms of solution quality and running-time indicates that linkage-learning is more useful in large instances (see Figures 4.1(i)-4.1(l) and 4.4(i)-4.4(l), respectively).

Figure 4.5 shows the distribution of the first-hitting time with a population size $\mu = 500$. As mentioned in Section 4.3.1, the LTGA variants make progresses by large increases of the best fitness. However, with large populations such a pattern requires more time to occur and progresses in fitness slowed down. With a population size $\mu = 500$ this fact was evident and in

most cases the LTGA did not outperform the rLTGA (p-values ≈ 1 were obtained).

Figure 4.5: Distributions of the first-hitting time with $K = 2, 5, 10, 15$, $N = 100, 500, 1000$ and $\mu = 500$.

For instances with $K = 2, 10, 15$ and problem sizes $N = 100, 500, 1000$, the differences in the running-time between the LTGA and the rLTGA did not support the use of linkage-learning. Moreover, such differences increased with the increase of multimodality, i.e. rLTGA became even faster than LTGA. Mutation slowed down the first-hitting time in all cases and the GA showed a high variance in its first-hitting time.

For instances of moderate multimodality ($K = 5$), the LTGA was the fastest among the algorithms in all problem sizes. Furthermore, it was also most effective than the rLTGA, as shown in Figures 4.2(i)-4.2(l). In such cases the LTGA was better than rLTGA, i.e. more effective and more efficient, which means linkage-learning plays an important role in the solution of such instances. Section 4.3.3 summarizes the results concerning such aspects for the estimation of a range of usefulness for linkage-learning in terms of K and N .

4.3.3 Range of Usefulness

All experiments aimed at verifying the existence of a range of usefulness for linkage-learning, i.e., some degrees of multimodality (represented by K) in different problem sizes (N), in which

linkage-learning would be more effective. The results have shown the validity of the hypothesis and the performance of the LTGA deteriorated, in relation to the rLTGA, as K increased and we have evidences to conclude linkage-learning is more effective in instances with $K \approx 5$. Coincidentally, most of the previous studies that evaluated EDAs in NK -landscapes used values of K around this threshold (Hauschild and Pelikan, 2010; Pelikan, 2008, 2010, 2011; Pelikan et al., 2011; Thierens and Bosman, 2011, 2012a). This section summarizes and generalizes such evidences by analyzing results for all $1 \leq K \leq 15$. Moreover, since NK -landscapes with *nearest-neighbors linkages* have also been used to evaluate EDAs, they are also considered here.

The algorithms were run 100 times for each triple (N, K, μ) . General information about linkage-learning usefulness was collected by counting how many times the LTGA outperformed the rLTGA in terms of solution quality. The balance was achieved by subtracting from the total the number of cases in which the rLTGA found the best solutions. Therefore, for every pair (N, K) there is a number that summarizes the results. Figure 4.6(a) shows the counting (normalized within the $[0, 1]$ interval) concerning *random linkages* and Figure 4.6(b) shows the results concerning *nearest-neighbors linkages*.

Figure 4.6: For each pair (K, N) the values indicate the probability of the LTGA outperforming the rLTGA (according to the experiments). Linkage-learning was more useful in light-colored regions.

The values on the maps (Figure 4.6) can be interpreted as the probability of the LTGA outperforming the rLTGA in each configuration of (N, K) and considering all population sizes $\mu = 100, 500, 1000$ (the color is lighter if the probability of LTGA winning is high). Regarding *random linkages*, as shown in Figure 4.6(a), the range in which linkage-learning provided the best benefits encompasses the very restricted region defined by the $1 \leq K \leq 6$ interval. On the other hand, there is a large region in which linkage-learning produced reasonable benefits. For example, for large instances ($N = 1000$), linkage-learning was reasonably useful in the large $1 \leq K \leq 13$ interval. However, for small instances ($N = 100$), linkage-learning was useful only in the small region delimited by the $1 \leq K \leq 5$ interval. This fact might be related to the density of interactions among variables, which grows according to the proportion K/N . Regarding *nearest-neighbor linkages*, as shown in Figure 4.6(b), regions were well delimited, independently of the problem size (N), and in all cases linkage-learning was of great utility

in the $1 \leq K \leq 9$ interval. These results demonstrate that *random linkages* induces harder instances than *nearest-neighbor linkages*.

Although the maps in Figure 4.6 elucidate the limits of linkage-learning, they do not quantify how well LTGA outperformed the rLTGA. To obtain such information we use a different procedure. For example, if the LTGA finds a much better solution (higher fitness) than the rLTGA, a large positive contribution is accumulated (difference between the fitness of solutions found by each algorithm). If, instead, the rLTGA finds a better solution, the difference will be negative and subtracted from the total. Through this procedure, we estimate how intensely linkage-learning has contributed to the search, see Figure 4.7.

Figure 4.7: For each pair (K, N) , the values indicate how intensely the LTGA has outperformed the rLTGA (according to experiments). Light-colored regions indicate values of K in which significant benefits were obtained.

This procedure is more restrictive than the first, therefore, the regions where linkage-learning was more effective have shrunk for both random and nearest-neighbor linkages. On the other hand, the agreement between such regions has increased considerably and, in both cases, the sweet-spot was delimited by the $2 \leq K \leq 6$ interval. Similarly to the map in Figure 4.6(a), the linkage-learning usefulness in Figure 4.7(a) (random linkages) depends on the problem size. For nearest-neighbor linkages, as shown in Figures 4.6(b) and 4.7(b), the situation was different and the range of usefulness was mostly independent of the problem size N .

4.4 Conclusions

EDAs emerged as an alternative to GAs in complex optimization problems. Most of the early studies were influenced by the Building Block (BB) hypothesis (from a reductionist perspective) and assumed the existence of detectable subcomponents that could be useful during the search for global optima solutions. During the pursuit of effective/efficient EDAs, many different linkage-learning algorithms (model-building) were proposed and evaluated, mostly in the context of benchmark functions. Although very efficient in such benchmarks, in many cases the promising results did not hold in well-known NP-hard optimization problems.

Some theoretical studies contributed to the skepticism regarding the applicability of EDAs as general problem solvers. The main concern is about the limitations of linkage-learning in producing useful linkage-models. This chapter analyzed such limitations from the perspective of multimodality. It was verified that from a certain degree of multimodality it becomes hard to extract useful statistics from the population of candidate solutions, situation in which linkage-learning is not able to produce accurate linkage-models.

The impact of multimodality in linkage-learning was evaluated by using the NK -model, from which instances of non-separable functions can be generated. By a comparison between the LTGA and rLTGA, regarding effectiveness and efficiency, in instances of different sizes ($N = 100, 500, 1000$) and degrees of multimodality ($1 \leq K \leq 15$) we had a method to assess the limitations of linkage-learning. Since the rLTGA uses random linkage-trees during the search, it is expected to find solutions of quality similar to the LTGA only in cases where linkage-learning is not so useful. The results of such comparison confirmed our hypothesis and in highly multimodal instances the LTGA and rLTGA performed very similarly, i.e. the results did not support the use of linkage-learning for a large K .

Many other collateral results were also evaluated during the experiments. Regarding mutation, although there were benefits of using it in some instances, they were not significant enough to suggest the implementation of mutation in the LTGA. In terms of population size, the LTGA variants showed a preference to small populations, an unusual fact among most of the EDAs.

These conclusions shed some light on promising, and crucial, research directions for EDAs in binary search spaces. The limitations verified both theoretically and experimentally in this study, suggest linkage-learning should not be seen as general tool applicable to any optimization problem (which is usually the scenario seen in the literature). In summary, we would consider linkage-learning as an interesting tool when solving large-scale black-box optimization problems with moderate multimodality. At last, it is worth noting the importance of fallback reproduction operators for situations in which linkage-learning is not accurate.

Chapter 5

Linkage-learning and the Multidimensional Knapsack Problem

Chapter 4 used the NK -landscape model to assess the limitations of linkage-learning according to the multimodality (represented by K) and size (N) of the instances. From the results, the range of multimodality in which linkage-learning excels was estimated evidencing the fact that in some optimization problems the accuracy of linkage models might not be enough to help the search for global optima solutions. However, there is no simple way to know beforehand if linkage-learning suits, or not, a certain optimization problem. This chapter considers the well-known Multidimensional Knapsack Problem (MKP) in order to exemplify the difficulties of applying linkage-learning in a practical problem. Results, in terms of effectiveness, are discussed and the linkage models built analyzed accordingly to specific properties of the MKP.

5.1 Introduction

The MKP is a well-known strongly NP-Hard combinatorial optimization problem (Freville, 2004; Puchinger et al., 2010). The objective is, from a set of items, to choose a subset which gives maximum total profit, restricted by knapsack-capacity constraints. Differently from the unidimensional case, in the MKP the knapsack capacity is bounded by $d > 1$ constraints and formulated as follows:

$$\max \quad \varphi(\mathbf{x}) = \sum_{j=1}^N p_j x_j, \quad (5.1)$$

$$\text{subject to} \quad \sum_{j=1}^N w_{ij} x_j \leq c_i \quad i = 1, \dots, d, \quad (5.2)$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, N. \quad (5.3)$$

where, there are N items available, with profits $p_j > 0$ ($j = 1, \dots, N$), and d resources available in amounts $c_i > 0$ ($i = 1, \dots, d$). Each item chosen contributes p_j to the total profit and consumes w_{ij} from each resource i . The number of resources available define the knapsack dimensionality and in *feasible* solutions the total resource consumption do not surpass any capacity c_i , $\forall i \in [d]$, with $[d] = 1, \dots, d$.

The MKP is considerably harder than its unidimensional counterpart, not admitting an efficient polynomial-time approximation scheme even for $d = 2$ (Kulik and Shachnai, 2010). Furthermore, MKP hardness increases with d and larger instances still cannot be effectively solved to optimality. Due to these characteristics a large number of papers have been published regarding the MKP, concerning both exact and heuristic methods.

One of the earliest methods for MKPs was a dynamic programming approach proposed by Gilmore and Gomory (1966), which was further extended by Weingartner and Ness (1967) by embedding heuristics in the algorithm. Branch-and-bound was also used to improve dynamic programming approaches, as proposed by Marsten and Morin (1977) and Marsten and Morin (1978). In the same context, Shih (1979) proposed the first branch-and-bound implementation based on linear programming. However, due to high space requirements and limited computational capacity, such approaches were limited to very small problems. In order to circumvent such limitations, Gavish and Pirkul (1985) developed a branch-and-bound method using rules to reduce the problem size, obtaining better results than Shih's method. A more recent approach based on an approximate dynamic programming was proposed by Bertsimas and Demir (2002), whereas a hybridization with a branch-and-cut-procedure was proposed by Boyer et al. (2009).

In the 1990's, many important results were obtained through a more extensive use of meta-heuristics. Tabu-search methods, as developed by Glover and Kochenberger (1996) and further improved by Freville and Plateau (1994), were able to solve to optimality all the public instances available so far (Freville, 2004). At the same year, Chu and Beasley (1998) proposed the first competitive Genetic Algorithm (GA) for the MKP and they also made available a well-designed instance set, containing large correlated instances ($n = 100, 250, 500$ and $d = 5, 10, 30$). Several of their results for these instances were considerably improved by Vasquez and Hao (2001) and Vasquez and Vimont (2005), through a tabu-search algorithm. After that, many studies proposed new tabu-search methods for the MKP (Li and Curry, 2005; Li, 2005; Li et al., 2012). In the GA field, Chu & Beasley Genetic Algorithm (CBGA) has influenced a series of studies concerning new GA operators and fitness-landscapes analysis (Gottlieb, 2000b, 2001; Raidl, 1998; Tavares et al., 2006, 2008), and still has been used as reference for more recent works with univariate Estimation of Distribution Algorithms (EDAs) (Kong et al., 2008; Martins and Delbem, 2013; Martins et al., 2013; Wang et al., 2012b), Particle Swarm Optimization (Azad et al., 2014; Chih et al., 2014) and Differential Evolution (Wang et al., 2012a). From now on, we use the CBGA as reference framework. for a more complete review of MKP's peculiarities see Freville (2004), Hanafi and Wilbaut (2011), Kellerer et al. (2004), and Puchinger et al. (2010). For state-of-the-art results we suggest Boussier et al. (2010), Croce and Grosso (2012), and Hill et al. (2012).

5.1.1 Efficiencies and the Core Concept

In the most successful algorithms for the MKP, both exact and heuristics, the concept of *core* is usually employed. The main idea of *core* is very simple, some items are more valuable than others due to the relation between their profit and size (*pseudo-utility ratio* or *efficiency*). Items with high/low efficiencies are commonly set to 1/0 in high-quality solutions, therefore, a dimensionality reduction is possible and the search only needs to be performed on items with medium efficiencies. Although finding the exact *core* is also a difficult problem, many algorithms estimate it in order to effectively find globally optimal solutions. For unidimensional knapsacks this idea has led to very effective algorithms (Balas and Zemel, 1980; Martello et al., 2000).

The key-point for *core*-based algorithms is the definition of *efficiency*. In the unidimensional case, every item j has a profit p_j and a weight w_j and their *efficiencies* are given by $e_j^1 = p_j/w_j$. Considering the decision variables in non-increasing order of e_j , a relevant greedy solution can be obtained by iteratively trying to insert each item. In the multidimensional case, however, there is no obvious *efficiency* measure anymore, since there are now d different weights for each item. Puchinger et al. (2010) have discussed the effectiveness of different efficiency measures for the MKP, next we review some of their results.

The first efficiency measures, as proposed by Dobson (1982), Equation (5.4), did not account for possible differences of magnitude of constraints, usually leading to poor results.

$$e_j^{\text{simple}} = \frac{p_j}{\sum_{i=1}^d w_{ij}}, \quad (5.4)$$

A natural way to overcome this difficulty it was to scale the weights according to capacities in each dimension (Puchinger et al., 2010).

$$e_j^{\text{scaled}} = \frac{p_j}{\sum_{i=1}^d \frac{w_{ij}}{c_i}} \quad (5.5)$$

Senju and Toyoda (1968) proposed a different way to account for the relative contribution of constraints by computing the difference between the capacity c_i and the total weight of all items in dimension i .

$$e_j^{\text{senju}} = \frac{p_j}{\sum_{i=1}^d w_{ij} \cdot ((\sum_{j=1}^N w_{ij}) - c_i)} \quad (5.6)$$

All these different formulations were further generalized by Fox and Scudder (1985), which considered relevance values r_i for every constraint. These relevance values describe how promising is to pack an item in a dimension i considering the quantity of resources available in that dimension. The use of different relevance values produces different efficiencies.

$$e_j^{\text{general}} = \frac{p_j}{\sum_{i=1}^d r_i \cdot w_{ij}} \quad (5.7)$$

Following this framework, Freville and Plateau (1994) proposed the relevances defined by Equation 5.8, yielding efficiencies e_j^{freville} .

$$r_i = \frac{\sum_{j=1}^N w_{ij} - c_i}{\sum_{j=1}^N w_{ij}}. \quad (5.8)$$

According to Puchinger et al. (2010), the relevance values r_i can be interpreted as surrogate multipliers. Therefore, dual variables $\lambda_i, \forall i \in [d]$, for an optimal solution of the linear-relaxation of the MKP can be used as relevance values.

$$e_j^{\text{dual}} = \frac{p_j}{\sum_{i=1}^d \lambda_i \cdot w_{ij}}, \forall j \in [N]. \quad (5.9)$$

Indeed, Kan et al. (1993) showed the connections between solutions produced by a greedy heuristic using dual relevance values and the primal solution of the linear relaxation of the MKP. A similar greedy heuristic was applied by Chu and Beasley (1998) (as discussed in Section 5.1.2). Later on, Puchinger et al. (2010) verified the superiority of e_j^{dual} by performing an extensive comparison with all the different efficiency measures previously described (e_j^{simple} , e_j^{scaled} , e_j^{senju} , e_j^{freville}). From such results and considering CBGA implementation, from now on, we consider e_j^{dual} as default efficiencies. For a more recent discussion about constructive heuristics for the MKP we suggest Bertsimas and Demir (2002), Croce and Grosso (2012), Fleszar and Hindi (2009), and Hanafi and Wilbaut (2011).

5.1.2 Chu and Beasley Genetic Algorithm

Many GAs were proposed for the MKP in the 1990's. Some of them used *penalty functions* in order to guarantee feasibility while others used *repair operators* based on different *efficiencies*. However, what seems to be most promising approach was proposed by Chu and Beasley (1998) (CBGA), and its framework has been intensively used as reference for development of new GAs (Raidl, 1998). Indeed, more recent analysis have shown the quality of the CBGA. First, Gottlieb (2000a) showed difficulties in finding feasible solutions when using various penalty functions and proposed a new penalty to alleviate this problem; however, in a posterior study Gottlieb (2001) showed the superiority of repair methods in biasing the search to the feasible region. Second, Tavares et al. (2006, 2008) analyzed MKP's fitness landscapes and quantified (by using different measures) the superiority of the CBGA framework when compared to different combinations of representations and repair-operators. Assuming the improvements suggested by Gottlieb (2000a), this section describes the CBGA.

The CBGA uses a direct representation (binary strings), binary tournament selection, standard uniform crossover, simple mutation of two bits and a repair mechanism (based on e_j values) in a steady-state fashion. To increase diversity, the CBGA avoids the insertion of duplicated solutions in the population (see Algorithm 5.1).

Algorithm 5.1 Chu and Beasley genetic algorithm (CBGA).**Require:** The population size μ and efficiencies e^{dual} .

```

1:  $\mathcal{O} = \{o(1), \dots, o(N)\} : e_{o(1)}^{\text{dual}} \leq e_{o(2)}^{\text{dual}} \leq \dots \leq e_{o(N)}^{\text{dual}}$ 
2:  $P \leftarrow \text{randomPopulation}(\mu)$  ▷ Generates  $\mu$  random candidate solutions
3:  $\text{repairAllSolutions}(P)$  ▷ Makes all initial solutions feasible
4: while !stopCriteria() do
5:    $u \leftarrow S_{\text{Tour}}(f, 2)(P)$ 
6:    $v \leftarrow S_{\text{Tour}}(f, 2)(P)$ 
7:    $y \leftarrow R_{\text{Unif}}(u, v)$ 
8:    $y \leftarrow M_{1/N}(y)$ 
9:    $y \leftarrow \text{repairSolution}(y, \mathcal{O})$  ▷ Makes  $y$  feasible
10:  if isUnique( $y, P$ ) then ▷ If  $y$  is not a duplicate in  $P$ 
11:     $x^{\text{worst}} \leftarrow y$  ▷ Substitutes the worst solution in  $P$ 
12: return the best solution in  $P$ .
```

In Algorithm 5.1, the only MKP specific step is the repair operator. It is responsible for guaranteeing that exclusively feasible solutions are kept in the population. The repair operator is implemented by a simple two-step heuristic (see Algorithm 5.2).

Algorithm 5.2 Repair feasibility**Require:** A solution x and an ordering $\mathcal{O} = \{o(1), \dots, o(N)\} : e_{o(1)}^{\text{dual}} \leq e_{o(2)}^{\text{dual}} \leq \dots \leq e_{o(N)}^{\text{dual}}$.

```

1: // DROP phase:
2: for all  $j \in \{o(1), \dots, o(N)\}$  do
3:    $x_j \leftarrow 0$ 
4:   breakIfFeasible( $x$ ) ▷ Stops the loop if  $x$  has become feasible
5: // ADD phase:
6: for all  $j \in \{o(N), \dots, o(1)\}$  do ▷ Evaluates items in non-increasing order of  $e_j^{\text{dual}}$ .
7:   if  $j$  fits in the knapsack then
8:      $x_j \leftarrow 1$  ▷ Inserts item  $j$  in the knapsack
```

First, the items are sorted according to their efficiencies e_j^{dual} , such ordering is used in two phases called DROP and ADD. Starting from items with small e_j^{dual} , the DROP phase removes items from the knapsack until a feasible solution is obtained. In the ADD phase, the items are evaluated in the opposite order, i.e. items with high efficiencies first. Those items which fit in the knapsack are inserted, all items are tested in this phase. This procedure is guaranteed to generate a feasible solution.

5.2 Methodology

Algorithms like the Linkage Tree Genetic Algorithm (LTGA) were proposed, in most cases, assuming nearly-decomposable systems as inspiration (Simon, 1996). Although such a characteristic is commonly found in nature, we cannot extrapolate this fact to optimization problems without a thorough analysis. If an optimization problem does not present evident substructures or

they are highly-overlapping, multimodality becomes an impediment for linkage-learning (Chapter 4). In this section experiments are set in order to show that MKP seems to be an example of this situation, in which significant substructures are hard to identify, making linkage-learning useless in the scenario adopted for the experiments.

In order to verify the usefulness of linkage-learning for the MKPs, the previous analysis provided by Martins and Delbem (2013) is extended by looking at the performance of different variants of the LTGA. The impact of linkage-learning when compared with random-models is evaluated in different scenarios, using the results of the well-known CBGA as reference. The operators used in the CBGA were also implemented on LTGA (mutation and diversity preservation) and the results compared again. A different hierarchical clustering algorithm is also used for the linkage-tree learning (*complete linkage*) instead of Unweighted Pair Grouping Method with Arithmetic-mean (UPGMA). From these settings, the main aspects responsible for the quality of learning in the LTGA are covered, in such a way that the results enable conclusions about why linkage-learning is not crucial for the LTGA when solving MKPs.

5.2.1 Benchmark Instances

The instances provided by Chu and Beasley (1998) have been widely used in the literature and will be also the basis for our experiments. The set comprises instances of size $N \in \{100, 250, 500\}$, for each size there are instances of dimension $d \in \{5, 10, 30\}$ and tightness ratio $\varrho \in \{0.25, 0.5, 0.75\}$ (where $\varrho = c_i / \sum_{j=1}^N w_{ij}$). For each tuple (N, d, ϱ) there are 10 instances, totaling 270 different instances.

The resource consumption w_{ij} were generated as integer numbers drawn uniformly at random in the interval $(0, 1000)$. The profits p_j are correlated to the weights as $p_j = \lfloor 500U(0, 1) \rfloor + \sum_{i=1}^d w_{ij}/d$, where $U(0, 1)$ is a real number generated uniformly at random in the interval $(0, 1)$. A small ϱ implies a harder problem, however, most of these instances were already solved to optimality by exact methods, except for some of the largest ones ($N = 500, d = 10, 30$) (Boussier et al., 2010). All these instances are available on the ORLIB website¹.

5.2.2 Experiment Settings

CBGA results are used as a reference to evaluate the performance of many different LTGA variants. This section describes each of these variants, briefly explaining why they were chosen and define the parameters used for the algorithms in the experiments²:

1. CBGA proposed by Chu and Beasley (1998) with improvements from Gottlieb (2000a);
2. LTGA, original algorithm proposed by Thierens and Bosman (2011);

¹<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/mknapinfo.html>

²<http://lcrserver.icmc.usp.br/attachments/download/278/ltga.zip>

3. LTGA with CBGA mutation. The intrinsic nature of MKP, favors a fast convergence of decision variables outside the core. This fact contributes for the loss of diversity of core variables. To overcome this drawback, a simple two standard mutation is applied every time there is an exchange of genetic material;
4. *Random* LTGA (rLTGA), this algorithm uses a linkage-tree built from a random matrix of dissimilarities. It is a reference for the benefits of linkage-learning;
5. rLTGA with CBGA mutation;
6. *Sorted* LTGA (sLTGA), it sorts the population in order to improve the worst solutions first at each generation. This version evaluates if the order of improvements matter to the whole search. The hypothesis is that by improving the worst solutions first would also improve the results when crossing-over the best solutions;
7. SLTGA with CBGA mutation;
8. *Sorted Random* LTGA (srLTGA);
9. srLTGA with CBGA mutation;
10. *Complete* LTGA (cLTGA), it uses *complete-linkage* instead of UPGMA. Due to the fast convergence of variables outside the core, the linkage-tree built by using d_{avg} tends to create skewed trees, in the sense that converged variables are clustered first and new clustering steps just append a new node to the previous subtree. In an attempt to avoid this effect, we also evaluate the results with the d_{max} ;
11. cLTGA with CBGA mutation;
12. *Complete-Random* LTGA (crLTGA);
13. crLTGA with CBGA mutation;
14. *Sorted Complete* LTGA (scLTGA), this algorithm uses the d_{max} matrix update;
15. cLTGA with CBGA mutation;

These algorithms were run in two scenarios: (1) with the algorithms previously described and (2) with the diversity preservation mechanism from the CBGA implemented in all LTGA variants. In both scenarios, each algorithm was run with 10 different randomly generated seeds. For each seed 10 runs were performed, in a total of 100 runs for each pair algorithm-instance. The population size was fixed in $N = 100$ and the algorithms stopped when the population has converged to the same solution or when a maximum of 500,000 objective function evaluations had been performed.

Both stopping criteria have a reason to be chosen. First, the LTGA converges very fast and we want to observe the quality of the solutions found by the original algorithm and its ability to avoid premature convergence. Second, when using diversity preservation the algorithms will not converge, thus, an upper bound on the number of fitness evaluations is needed, we have chosen 500,000 (this is half the value used Chu and Beasley (1998), but still is far larger than the value required by the LTGA without diversity preservation).

5.3 Results and Discussion

This section describes and compare the results of both scenarios previously defined. We first show the average performance of each algorithm in all 270 Chu and Beasley instances. From such comparison we are able to identify the best algorithms and the impact of each element (mutation, diversity preservation and clustering method) on the efficacy of the LTGA. The averages were computed considering the whole set of instances, instead of separating instances according to their ρ . This decision enables us to see the big picture of average performances of the algorithms and, since the same instances were used for all algorithms, the averages provide a relevant estimate for the performance measures used.

5.3.1 Average Gap from the Optimum

Figure 5.1 shows the average gaps reached by each algorithm for $d = 10, 30$, the first and third rows represent the results for the original algorithms, while the second and forth rows represent the results using the CBGA diversity preservation in the LTGA variants. Abscissa in Figure 5.1 represents the algorithms in the same order described in Section 5.2.2.

Algorithms using mutation are represented by odd identifiers. As can be seen, mutation has always improved the results, decreasing the gap to the best known solution. Moreover, mutation seems to be more important when solving small instances. For these instances the LTGA converges prematurely, i.e. large gaps. Therefore, we can suppose that mutation slowed-down the convergence rate enabling more time for better solutions to be found. When using diversity preservation, the algorithms do not suffer from premature convergence anymore, but even in this case mutation is still more effective on small instances (i.e. smaller gaps). To explain this fact we suppose two hypothesis: (1) the interaction between mutation and the repair operator enable a better local exploitation, justifying the small gaps for instances with $N = 100$, (2) the influence of mutation decreases as N grows to $N = 250, 500$, indicating that standard mutation may be weak to solve large instances.

Considering the second scenario (second and forth rows in Figure 5.1), it is easy to see that the use of diversity preservation improved the results for all instances, note the differences in scale. In this situation, LTGA using mutation has performed very similar to the CBGA. The point here is that by avoiding the insertion of duplicate solutions in the population, the LTGA has

Figure 5.1: Average percentage gap from the best known solution for $d = 10, 30$ and $N = 100, 250, 500$. The averages were computed from the results for all instances. Figures (a),(b),(c),(g),(h),(i) show the gaps reached Without Diversity Preservation (WoDP). Figures (d),(e),(f),(j),(k),(l) show the gaps reached With Diversity Preservation (WDP).

time to explore the search space, which does not happen in the original algorithm. The original LTGA converges too fast, as it finds an above average solution, the population in a short time converges to that solution, and the search ends trapped in a low-quality local optimum. We could argue that a diversity preservation mechanism should be adopted by the LTGA in order to better solve a larger class of optimization problems.

Regarding the clustering procedure, the use of *average linkage* or *complete linkage* did not impact the results. Furthermore, linkage-trees and random linkage-trees have led to very similar results, corroborating with the conclusions drawn by Martins and Delbem (2013). It is also worth mentioning the dispersion of the distribution of the points in Figure 5.1, which is a result of the different instances considered.

5.3.2 Number of Function Evaluations

The average number of fitness evaluations required by each algorithm to reach their final solutions (number of evaluations at first-hitting the best solution found) is evaluated only for the LTGA using diversity preservation, since it was shown to be essential for the quality of solutions found. As seen in Figure 5.2, the number of evaluations from CBGA and LTGA variants were very similar, with a slightly advantage for the CBGA.

Figure 5.2: Average number of function evaluations when the algorithms first hit their best solutions, for $d = 5, 10, 30$ and $N = 100, 250, 500$. The averages were computed from the results for all instances and with diversity preservation. Figures (a),(b),(c) show the number of evaluations for $d = 5$, Figures (d),(e),(f) for $d = 10$ and Figures (g),(h),(i) for $d = 30$.

For the small instances (with $N = 100$) and low dimensionality ($d = 5, 10$) the results are skewed around the median and a large number of points is seen as outliers. Excluding these two cases, the remaining figures do not show many outliers and the points are more evenly distributed around the median. From these results we can see that CBGA and LTGA have spent a similar number of function evaluations to reach their final solutions. However, the CBGA has found better solutions, revealing better capacity to solve the MKP.

5.3.3 Average Running-time

This section compares the running-time of CBGA and LTGA. Since the advantage of the CBGA is relatively small in some cases, running-time comes as a last performance measure.

Figure 5.3: Average total time to perform 500,000 fitness evaluations for $d = 5, 10, 30$ and $N = 100, 250, 500$. The averages were computed from the results for all instances and with diversity preservation. Figures (a),(b),(c) show the number of evaluations for $d = 5$, Figures (d),(e),(f) for $d = 10$ and Figures (g),(h),(i) for $d = 30$.

The LTGA without diversity preservation always converged prematurely, being faster than the CBGA but in a bad way. Figure 5.3 shows the results with diversity preservation, where

both algorithms are in similar conditions, with the same amount of function evaluations as limit. Their total running-times were very similar, with the LTGA being the fastest in some cases. This is an important point, since it shows that the LTGA can perform as well as simple GAs concerning running-time even in problems for which linkage-learning not seems to be essential. Although this is a good result for the LTGA, it is not clear why the CBGA has taken more time to perform the 500,000 function evaluations. However, it is worth mentioning that the differences in running-time only happen for $N = 100, 250$ and are of 2 seconds approximately, these differences practically vanish in larger and harder instances, $N = 500$ and $d = 30$.

5.3.4 On the Usefulness of Linkage-trees

The main attractive for the use of machine learning techniques in GAs is the possibility of efficiency improvements due to the learning of structural properties from a black-box optimization problem. However, the results in previous sections show that linkage-learning did not provide any improvements in the case of MKPs. We could assume the fact the MKPs is not nearly-decomposable as justification, however, this would not help to understand the problem. This section relates the *knapsack-core* concept with the results of linkage-tree learning in order to better understand why the LTGA is not able to exploit the structure of the MKPs.

The true knapsack-core is unknown but assume the following general notation: there is a index set $\mathcal{I} = \{1, \dots, N\}$ referencing every decision variable, from which the set $\mathcal{C} \subset \mathcal{I}$ represent a set of decision variables indexes containing the knapsack-core and $\mathcal{I} \setminus \mathcal{C}$ a index set containing non-core decision variables.

The decision variables in the *core* ($x_j, \forall j \in \mathcal{C}$) have medium efficiency values e_j , therefore, they are expected to take more time to converge when compared to variables outside the *core* ($x_j, \forall j \in \mathcal{I} \setminus \mathcal{C}$). This happens since the insertion of a item with large efficiency value and remotion of an item with small efficiency value will probably increase the fitness value, which commonly happens in the first generations, making variables in such situation converge first. Besides, the repair operator implemented by the CBGA enforces the convergence of non-core variables. In order to show how this impacts on the linkage-tree built, assume that all variables $x_j \in \mathcal{I} \setminus \mathcal{C}$ have converged to 0 or 1 and decision variables $x_i \in \mathcal{C}$ have not converged. In this situation, the dissimilarity matrix computed from *NVI* is reduced to:

$$d(X_p, X_q) = \begin{cases} 1, & (x_p \in \mathcal{I} \setminus \mathcal{C} \wedge x_q \in \mathcal{C}) \vee (x_p \in \mathcal{C} \wedge x_q \in \mathcal{I} \setminus \mathcal{C}), \\ 0, & x_p, x_q \in \mathcal{I} \setminus \mathcal{C}, \\ NVI(X_p, X_q), & \text{otherwise.} \end{cases}$$

In other words, the dissimilarity is 0 between converged variables and 1 between converged and non-converged variables. The dissimilarity between non-converged variables is within the $(0, 1)$ range. To understand why these values appear, we need to observe the *NVI* computation in

details. The *variation of information* is normalized by the joint entropy:

$$NVI(X_p, X_q) = 1 - \frac{I(X_p, X_q)}{H(X_p, X_q)},$$

therefore:

1. if $I(X_p, X_q) = 0$ and $H(X_p, X_q) > 0$, $NVI(X_p, X_q) = 1$ is maximum, and
2. if $I(X_p, X_q) = 0$ and $H(X_p, X_q) = 0$, $NVI(X_p, X_q) = 0$ is minimum³.

The first situation occurs with any pair of variables (x_p, x_q) , with $x_p \in \mathcal{I} \setminus \mathcal{C}$ and $x_q \in \mathcal{C}$ (or vice versa). In this case, $p(X_p = x) = 1$, i.e. x_p has converged to the value x . As x_q has not converged yet, $p(X_q = x)$ equals to some value $0 < q < 1$.

Theorem 5.1. *The normalized variation of information between two variables $x_p \in \mathcal{I} \setminus \mathcal{C}$ and $x_q \in \mathcal{C}$ is maximum and equal to $NVI(X_p, X_q) = 1$.*

Proof. Since x_p has converged, the joint entropy $H(X_p, X_q)$ is fully characterized by the values of x_q . For example, consider $p(X_q = 1) = 1$. In this case, the joint probabilities are defined as follows:

$$\begin{aligned} p(X_p = 0, X_q = 0) &= 0 \\ p(X_p = 0, X_q = 1) &= 0 \\ p(X_p = 1, X_q = 0) &= p(X_q = 0) \\ p(X_p = 1, X_q = 1) &= p(X_q = 1) \end{aligned}$$

Therefore, the joint entropy reduces to $H(X_p, X_q) = H(X_q)$. From the $I(X_p, X_q)$ definition and the fact $H(X_p) = 0$, follows that $I(X_p, X_q) = H(X_q) - H(X_p) - H(X_q) = 0$. As consequence $I(X_p, X_q) = 0$, $H(X_p, X_q) > 0$ and $NVI(X_p, X_q) = 1$. $\square$

The second situation occurs with any pair of variables $x_p, x_q \in \mathcal{I} \setminus \mathcal{C}$. In this case, both $p(X_p = x) = 1$ and $p(X_q = y) = 1$ in the current population. This situation is very simple, as follows.

Theorem 5.2. *The normalized variation of information between two variables $x_p, x_q \in \mathcal{I} \setminus \mathcal{C}$ is minimum and equals to $NVI(X_p, X_q) = 0$.*

Proof. Since $p(X_p = x) = 1$ and $p(X_q = y) = 1$, the joint entropy $H(X_p, X_q) = 0$, as consequence $I(X_p, X_q) = 0$ and $NVI(X_p, X_q) = 0$. $\square$

³This would lead to a division by zero, but it is assumed as 1.

From these results, we can infer the structure of the linkage-tree built by UPGMA. As the decision variables with high and low e_j tend to converge faster than the others, it is highly probable they compose $\mathcal{I} \setminus \mathcal{C}$. Therefore, from Lemma 5.2, such decision variables will be clustered first. Lets us assume that the first cluster C_1 is composed of $\{x_p, x_q\} \subseteq \mathcal{C}$, now UPGMA will update the dissimilarities between C_1 and any other variable x_k :

$$d_{\text{avg}}(C_1, \{x_k\}) = \frac{1}{|C_1| + 1}d(X_p, X_k) + \frac{1}{|C_1| + 1}d(X_q, X_k),$$

and two different cases are possible, $x_k \in \mathcal{I} \setminus \mathcal{C}$ and $d(x_p, x_k) = 0$ or $x_k \in \mathcal{C}$ and $d(x_p, x_q) = 1$, leading to:

$$d_{\text{avg}}(C_1, \{x_k\}) = \begin{cases} 0, & x_k \in \mathcal{I} \setminus \mathcal{C}, \\ 2/3, & x_k \in \mathcal{C} \end{cases}$$

These values show that after the first joining, variables $x_j \in \mathcal{I} \setminus \mathcal{C}$ will still be closer according to d_{avg} . Therefore, the next joining will also cluster converged variables, in fact this situation repeats until all variables $x_j \in \mathcal{I} \setminus \mathcal{C}$ have been clustered. Therefore, the linkage-trees will present two well-separated giant clusters, each of them containing the variables $x_j \in \mathcal{I} \setminus \mathcal{C}$ and the variables $x_j \in \mathcal{C}$, respectively. Figure 5.4 illustrates such tree, obtained in the first generation of the LTGA. The subtree on left represents the cluster composed of variables $x_j \in \mathcal{I} \setminus \mathcal{C}$ and all distances in this giant cluster are the same and equal to zero. The subtree on the right represent the cluster composed of variables $x_j \in \mathcal{C}$, as can be seen some hierarchical structure appears in this part but with large values of NVI .⁴

Figure 5.4: An example of linkage-tree built for an MKP instance with $N = 100$ and $d = 5$ at the first generation of the LTGA, after repair of the solutions. Two giant clusters are evident, on the left, converged variables, on the right, non-converged variables.

By the definition of *core*, is evident that the *knapsack core* is a subset of the rightmost subtree. Therefore, we could consider that the linkage-tree “learned”, in some sense, the *core*. However, such “learning” also occurs implicitly on the CBGA, therefore, this is not a point in favor of the LTGA. Both algorithms are able to find the same information, the CBGA implicitly and the LTGA explicitly. But even with the information provided by the linkage-tree, the LTGA was not

⁴Trees generated with the `hclust` function in R (<http://cran.r-project.org>) and NVI matrices.

able to outperform its random variants. The explanation for this fact is also shown in Figure 5.4, although the rightmost subtree shows a hierarchical structure, the confidence in those clusters is not enough, as it is shown by the NVI on the ordinate. In an ideal scenario, these values should be much smaller in order to be useful during crossover.

Figure 5.5 shows an example of how the linkage-trees built change along the generations. An important characteristic observable in these figures is that the NVI of variables in the rightmost subtree does not decrease in a useful way. In other words, the NVI is in almost all cases near its maximum value or near its minimum value, i.e. the dependence among variables is weak and this fact only changes when the variables start to converge. Under this conditions, the LTGA works the whole time with non-significant subsets, a fact that explains the similar results obtained by the use of random linkage-trees.

Figure 5.5: Examples of linkage-trees built for an MKP instance with $N = 100$ and $d = 5$ at the 2nd, 4th, 6th, 8th, 10th and 12th generations of the LTGA. Variables $x_j \in \mathcal{I} \setminus \mathcal{C}$ converged first and were grouped first. Decision variables $x_j \in \mathcal{C}$ have showed weak statistical dependencies, fact represented by high values of NVI .

These results make evident interesting aspects about the LTGA. The algorithm performs linkage-learning and uses the corresponding model independently of the accuracy or confidence

of the model built. In fact, the LTGA always produces a linkage-tree even if there is no apparent justification to cluster variables. Although this does not seem to be a problem for the LTGA, it may be a source of wrong conclusions about the effectiveness of EDAs, for example, by equivocally assuming that the effectiveness of the method is due to linkage-learning when, in fact, it might be due to crossover.

5.4 Conclusions

EDAs have proved themselves as effective methods to solve nearly-decomposable problems. However, the assumption of near-decomposability cannot be taken as an ubiquitous characteristic of optimization problems, therefore, there must be limitations for the effectiveness of such algorithms. This chapter extended the studies about the usefulness of linkage-learning to solve MKPs, by performing a comparison between the CBGA (Chu and Beasley, 1998) and the LTGA (Thierens and Bosman, 2011).

Various LTGA variants were implemented in order to evaluate the influence of mutation, diversity preservation and linkage-learning in the LTGA results. The CBGA was implemented based on improvements suggested by Gottlieb (2001) and the performance comparison was made using the 270 MKP instances provided by Chu and Beasley (1998). We showed results for 15 LTGA settings (with/without mutation, using different linkage-measures, using random linkage-trees, etc), in which the quality of solutions found, the number of fitness evaluations and running time were compared.

Concerning the quality of solutions, the CBGA has outperformed all LTGA variants. The original LTGA suffered from premature convergence and found solutions with considerably large gaps from the target. When diversity preservation was enabled on the LTGA, its results significantly improved, since the algorithm had time to explore/exploit the search space and consequently more time to find better solutions. Enabling mutation, LTGA obtained better results, proving that diversity is really a concern when using intense repair-operators. However, even with these adjustments, the CBGA still found the best solutions.

By evaluating the role of linkage-learning in the performance of the fifteen LTGA variants, we note that all of them have performed very similarly. The settings using random-linkage models have also reached the same quality for solutions, similar number of fitness evaluations and expected running-time. This fact corroborates with previous studies which had tried to measure the nearly-decomposability of the MKPs and also concluded for the hardness of identifying useful substructures (Martins and Delbem, 2013).

In a broader perspective, the performance comparisons made clear the weakness of linkage-learning to help solve the MKP. To understand why this is the case, the linkage-trees were analyzed using information-theoretic measures. The results have shown that such linkage-trees are primarily composed of two giant clusters, separating converged and non-converged variables. Furthermore, it is worth noting that the weak dependencies measured among non-converged

variables implies that the LTGA only identifies low-confidence substructures. This fact explains why the results with and without linkage-learning were so similar.

It is evident that there must be many optimization problems for which linkage-learning is not useful, as shown for the MKP. On the other hand, from the experiments with the LTGA we have witnessed its ability to keep a reasonable search power even without an accurate model, indeed it was able to find considerably high-quality solutions. Differently from many EDAs, the LTGA has a reproduction operator which performs very similarly to a local-search in situations in which linkage-trees are not of much help. We believe, such operators are important for any general propose EDA, since they can circumvent linkage-learning limitations and guarantee the usefulness of such methods in a large number of situations.

Chapter 6

Reproductive Bias and Linkage-learning in Multiobjective Optimization

Previous chapters have analyzed how multimodality influences the accuracy of linkage-learning, hence decreasing the efficiency of Estimation of Distribution Algorithms (EDAs). Such limitations of linkage-learning in single-objective optimization should be a concern in the multi-objective context, in which multimodality is an inherent characteristic. Some studies have reported benefits in using Multi-objective Estimation of Distribution Algorithms (MOEDAs). However, aside the studies of Sastry et al. (2005) and Martí et al. (2011b), most of the literature on MOEDAs have focused on straight adaptations of single-objective EDAs, as consequence, the benefits of MOEDAs are usually justified by arguments validated in single-objective problems.

This chapter addresses such controversies and analyzes the impact of linkage-guided crossover operators in multi-objective optimization. First, the desirable characteristic of reproduction operators is defined from their *uniformity-preservation* in the phenotype space. Next, traditional crossover operators are shown to be biased in the phenotype space and linkage-guided crossover is shown to approach uniformity preservation. From the perspective of *reproductive bias*, different Multi-objective Evolutionary Algorithm (MOEA) methodologies are analyzed and their ability to find a diverse set of non-dominated solutions is justified. At last, the effectiveness of *mating restrictions*, decomposition-based approaches and MOEDAs are discussed from a unique perspective that relates them all.

6.1 Introduction

A multi-objective optimization problem is defined by a feasible set $\mathcal{G}$, an objective set $\mathcal{P}$ and a mapping $\varphi : \mathcal{G} \mapsto \mathcal{P}$. The feasible set $\mathcal{G}$ is a subset $\mathcal{G} \subseteq \mathbb{R}^N$ (the decision space), whereas the objective set is the codomain of φ , $\mathcal{P} \subseteq \mathbb{R}^M$ (the objective space, with $M \geq 2$). Additionally,

$f : \mathcal{P} \mapsto \mathcal{F}$ maps objective vectors in $\mathcal{P}$ to an ordered set $\mathcal{F}$. Without loss of generality, the solution of an optimization problem consists in finding feasible solutions $\mathbf{x} \in \mathcal{G}$ which maximizes $f(\varphi(\mathbf{x}))$. Due to the lack of a total ordering in the objective space $\mathcal{P}$, the Pareto-optimal front is composed of incomparable objective vectors (Miettinen, 1999; Zitzler et al., 2003). Therefore, the main goal in solving Multi-objective Optimization Problems (MOPs) is to find a set of well-diversified (diversity) Pareto-optimal solutions (optimality/convergence) from which a target solution can be chosen *a posteriori* in the decision-making process.

MOEAs were proposed as an alternative method, in which *diversity* and *convergence* in relation to the Pareto-optimal front are pursued concomitantly (Bosman and Thierens, 2003). Differently from single-objective Evolutionary Algorithms (EAs), *fitness assignment* in MOEAs must take in account the quality and diversity of objective vectors within the current population. For such a purpose, many different approaches have been proposed along the years, such as, Pareto-dominance relations (Deb and Jain, 2014; Deb et al., 2002; Jain and Deb, 2014), hypervolume measures (Auger et al., 2012; Beume et al., 2007; Bringmann and Friedrich, 2013) and decomposition (Li and Zhang, 2009; Zhang and Li, 2007; Zhang et al., 2009).

Since the early studies on MOEAs, *fitness assignment* has been one of the main research focuses (Zhou et al., 2011). Dominance rankings was the first well-succeeded approach implemented by many MOEAs (Section 2.4). However, when solving MOPs with many objectives (usually $M > 3$), Pareto-dominance faces serious difficulties in providing the desirable selection pressure and does not produce good results (Ishibuchi et al., 2008b). The hypervolume measure was proposed as an alternative for such cases (Auger et al., 2012; Bringmann and Friedrich, 2013) and many algorithms have been developed relying on it (Bader and Zitzler, 2010; Beume et al., 2007). However, its computational complexity still is an impediment for problems with many objectives ($M \gg 2$) (Beume et al., 2009).

Regarding *reproduction operators*, the literature is not so abundant. Many methods have been proposed to improve convergence and diversity of non-dominated solutions, however, little has been investigated about the specific weaknesses of traditional operators. The studies of Ishibuchi and Shibata (2003a,b), for example, proved the benefits of *mating restrictions* in many different scenarios (Ishibuchi et al., 2010; Ishibuchi and Narukawa, 2005; Ishibuchi et al., 2007, 2008a, 2012, 2014), however, they did not clearly elucidate why such methodology works. More recent studies of Sato et al. (2013) and Ishibuchi et al. (2014) went beyond in such a direction by showing the benefits of generating offspring close to the parents, these studies give us directions to explain the effectiveness of mating restrictions, crossover and linkage-learning for MOPs.

This chapter supports a hypothesis that unifies aspects of different reproduction operators and helps the understanding of the fundamental requirements for reproduction operators in multi-objective optimization. In MOEAs with mating restrictions, recombination is restricted to occur between neighboring parents in the objective space (Ishibuchi et al., 2008a); in MOEDAs (Zhou et al., 2011) that use multiple models, the objective space is usually partitioned and solutions are generated in each subregion; in decomposition-based MOEAs, recombination occurs only

between neighboring parents (neighborhoods defined between aggregated functions) in the objective space (Ishibuchi et al., 2013). Such similarities suggests the choice of close parents during reproduction helps to improve traditional crossover (uniform, 1-point and 2-point crossover). This thesis supports that *uniformity-preservation* (see Definition 2.7) in the phenotype space is the characteristic approximated by these approaches, which underlies effective reproduction operators in MOEAs. The lack of this characteristic leads to *biased* operators, which spoil the benefits of diversity preservation mechanisms and decreases their efficacy in producing well-spread Pareto-fronts.

6.2 Methodology

This section analyzes the characteristics of simple bi-objective problems composed of additively separable functions. Our objective is to understand which kind of reproduction operators would increase the chances of finding Pareto-optimal fronts. The conclusions drawn for the artificial functions are then generalized to describe desirable characteristics of general crossover operators for MOPs. In this context, linkage models are shown to be useful regarding the maintenance of *diversity* and *convergence* towards the Pareto-optimal front.

6.2.1 Bi-objective Additively Separable Functions

Additively separable functions have been extensively used as reference to explain the behavior and benefits of linkage-learning in EDAs. In this section, bi-objective problems composed of two conflicting *mk*-trap functions are analyzed. The first objective function is composed of $\text{trap}(u, k_1)$ functions of size k_1 (Equation (6.1)), whereas the second objective function is composed of $\text{invtrap}(u, k_2)$ functions of size k_2 (Equation (6.2)).

$$\text{trap}(u, k_1) = \begin{cases} k_1 & \text{if } u = k_1, \\ k_1 - 1 - u & \text{if } u < k_1. \end{cases} \quad (6.1)$$

$$\text{invtrap}(u, k_2) = \begin{cases} k_2 & \text{if } u = 0, \\ u - 1 & \text{if } u > 0. \end{cases} \quad (6.2)$$

In this simple problem, the objectives agree on the variables composing the sub-functions, but disagree on the values of the variables. Solution $\mathbf{x}^*$ is the optimal solution for f_{trap} and $\mathbf{y}^*$ is the optimal solution for f_{invtrap} . These solutions are illustrated below with $k = 5$ ($k_1 = k_2 = k$).

$$\mathbf{x}^* = (11111 \ 11111 \ 11111 \ 11111 \ 11111 \ 11111)$$

$$\mathbf{y}^* = (00000 \ 00000 \ 00000 \ 00000 \ 00000 \ 00000)$$

The Pareto-optimal set $\mathcal{P}^*$ in this case is constituted by combinations of Building Blocks (BBs) from each sub-function. Since there are m BBs for each objective, there are 2^m possible solutions in $\mathcal{P}^*$. However, most of these solutions represent the same vector in the objective space and only $m + 1$ different points exist in the Pareto-optimal front (Martins et al., 2011). If $m = N/k$ is the number of BBs and Θ represents a BB from the first objective whereas Φ represents a BB from the second objective, the Pareto-optimal set can be built as follows

$$\begin{aligned}
 \mathbf{x}_0 &= (\Theta_1, \Theta_2, \dots, \Theta_{m-1}, \Theta_m) \\
 \mathbf{x}_1 &= (\Theta_1, \Theta_2, \dots, \Theta_{m-1}, \Phi_m) \\
 \mathbf{x}_2 &= (\Theta_1, \Theta_2, \dots, \Phi_{m-1}, \Phi_m) \\
 &\vdots \\
 \mathbf{x}_{m-1} &= (\Theta_1, \Phi_2, \dots, \Phi_{m-1}, \Phi_m) \\
 \mathbf{x}_m &= (\Phi_1, \Phi_2, \dots, \Phi_{m-1}, \Phi_m)
 \end{aligned} \tag{6.3}$$

Such a simple bi-objective problem was used by Sastry et al. (2005) to investigate the scalability of early MOEDAs. According to the authors the large number of Pareto-optimal solutions hampers the maintenance of the diversity of objective vectors. Despite the massive multimodality of Pareto-optimal solutions, they are composed of combinations of the optimal substructures for each objective and only a few different objective vectors compose the Pareto-optimal front.

Theorem 6.1. *In the above bi-objective additive separable functions, Pareto-optimal solutions $\mathbf{x} \in \mathcal{G}$ are composed of combinations of optimal BBs from each objective function.*

Proof. Each objective function f_i ($i = \{1, 2\}$) is composed of m subproblems: $f_i(\mathbf{x}) = h_{i1}(\mathbf{x}) + \dots + h_{im}(\mathbf{x})$, where $h_{ij}(\mathbf{x})$ is the contribution of the j -th subproblem ($j \in [m]$) to $f_i(\mathbf{x})$. Since the function f_i is separable and $\mathbf{x}^*$ is an optimal solution, then $h_{ij}(\mathbf{x}^*) \geq h_{ij}(\mathbf{x})$, $\forall j$.

In a proof by contradiction, consider the optimal solution $\mathbf{x}^*$ and substitute the solution of its j -th subproblem for a non-optimal solution $\mathbf{x}$, i.e. $h_{1j}(\mathbf{x}) < h_{1j}(\mathbf{x}^*)$. A new solution $\mathbf{u}$ is obtained with objective vector:

$$\mathbf{z}_{\mathbf{u}} = (f_1(\mathbf{x}^*) - h_{1j}(\mathbf{x}^*) + h_{1j}(\mathbf{x}), f_2(\mathbf{x}^*) - h_{2j}(\mathbf{x}^*) + h_{2j}(\mathbf{x})).$$

For $\mathbf{u}$ to be Pareto-optimal, there must not be any $\mathbf{y} \geq_P \mathbf{u}$. However, as the j -th subproblem from $\mathbf{x}$ is not optimal in any objective, a solution $\mathbf{y}$ can be created similarly to $\mathbf{x}$, except the j -th subproblem comes from an optimal solution $\mathbf{y}^*$ from the other objective. Since $h_{2j}(\mathbf{y}^*) > h_{2j}(\mathbf{x})$ and $h_{1j}(\mathbf{y}^*) < h_{1j}(\mathbf{x})$ only if $\mathbf{x}$ is optimal, then $\mathbf{y} \geq_P \mathbf{x}$, which contradicts our assumption. $\square$

Concerning the diversity of objective vectors, the Pareto-optimal set of is composed of solutions with a different number of BBs from each objective function. For example, the neighbors of solution $\mathbf{x}^*$ in the objective space share approximately the same number of BBs

from each objective. Figure 6.1 illustrates three Pareto-optimal solutions and their respective objective vectors in the Pareto-optimal front, with $m = 6$ and $k = 5$. Global-optima solutions

Figure 6.1: Pareto front of a bi-objective problem composed of $m = 6$ additively separable deceptive functions of size $k = 5$.

for each objective, $\mathbf{x}^*$ and $\mathbf{y}^*$, are the extreme solutions in the Pareto-front. On the other hand, solutions in the middle of the Pareto-optimal front are composed of half BBs from each objective. Based on such a perspective, reproduction operators should generate solutions composed of different numbers of BBs from each objective for obtaining a proper diversity of solutions along the Pareto front. Propositions 6.2-6.3 summarize some characteristics of bi-objective additively separable functions.

Conjecture 6.2. *The positioning of a vector $\mathbf{z}_x \in \mathcal{P}$ depends on the number of BBs from each objective that composes solution $\mathbf{x} \in \mathcal{G}$.*

Conjecture 6.3. *A BB-wise crossover (i.e. without disruption) between Pareto-optimal solutions can generate only Pareto-optimal solutions.*

mk -trap and mk -invtrap are unimodal functions with well-known global-optima solutions (only ones and only zeros, respectively). On the other hand, their combination into a bi-objective problem results in a highly multimodal function, where all combinations of optimal BBs lead to 2^m Pareto-optimal solutions. In this sense, the bi-objective problem is equivalent to a highly multimodal function and the same linkage-learning limitations described in Chapter 3 apply. Since so simple functions produce such a highly multimodal bi-objective problem, in more practical optimization problems higher multimodality is expected.

6.2.2 Bi-objective Non-separable Functions

The scenario described in the previous section is an idealistic example to which linkage-learning can be successfully applied. However, many other difficulties arise in more realistic MOPs. Functions are much more likely to be non-separable and a non-disruptive crossover is unlikely to occur. This section addresses the difficulties imposed by non-separability and generalizes our approach by relating it to traditional crossover operators.

All solutions generated from the application of the crossover described in Equations 6.3 are Pareto-optimal. However, this situation is not usual in real problems, as in general, solutions cannot be generated without disruption of some BBs. Therefore, Theorem 6.1 does not apply, as

illustrated next. Consider the bi-objective deceptive problem described in Section 6.2.1, but now with different subproblem sizes, $k_1 = 5$ for f_{trap} and $k_2 = 3$ for $f_{invtrap}$ ($N = 15$). Each of these objective functions has global-optima solutions $\mathbf{x}^*$, $\mathbf{y}^*$ and objective vectors $\mathbf{z}_1, \mathbf{z}_2$, illustrated in the Figure 6.2.

$$\begin{aligned}\mathbf{x}^* &= (11111 \ 11111 \ 11111), & \mathbf{z}_1 &= (15, 10) \\ \mathbf{y}^* &= (000 \ 000 \ 000 \ 000 \ 000), & \mathbf{z}_2 &= (12, 15).\end{aligned}$$

Figure 6.2: Objective space for a bi-objective problem, $N = 15, k_1 = 5, k_2 = 3$. Bullets (•) represent solutions generated by crossover between the extreme points in the front (*). Other objective vectors are represented by (o).

Exchanging one BB of size $k_1 = 5$ between $\mathbf{x}^*$ and $\mathbf{y}^*$, $\mathbf{x}_1$ and $\mathbf{y}_1$ are obtained. Since the subproblems are not the same size, the second BB is disrupted in the offspring solution $\mathbf{y}_1$. Although $\mathbf{x}_1$ seems to have been less disrupted than $\mathbf{y}_1$, $\mathbf{x}_1$ is Pareto-optimal while $\mathbf{y}_1$ is not.

$$\begin{aligned}\mathbf{x}_1 &= (00000 \ 11111 \ 11111), & \mathbf{z}_{11} &= (14, 9) \\ \mathbf{y}_1 &= (111 \ 110 \ 000000000), & \mathbf{z}_{12}^* &= (13, 12).\end{aligned}$$

By considering a BB of size $k_2 = 3$, leads to solutions $\mathbf{x}_2$ and $\mathbf{y}_2$, in which the first BB of size $k_1 = 5$ is disrupted in $\mathbf{x}_2$:

$$\begin{aligned}\mathbf{x}_2 &= (00011 \ 11111 \ 11111), & \mathbf{z}_{21} &= (12, 11) \\ \mathbf{y}_2 &= (111 \ 000 \ 000 \ 000 \ 000), & \mathbf{z}_{22} &= (9, 14).\end{aligned}$$

By exchanging two BBs from each objective ($k_1 = 5, k_2 = 3$), $\mathbf{x}_3, \mathbf{y}_3$, are produced.

$$\begin{aligned}\mathbf{x}_3 &= (000000000 \ 011 \ 111), & \mathbf{z}_{31} &= (13, 12) \\ \mathbf{y}_3 &= (111111111 \ 110 \ 000), & \mathbf{z}_{32} &= (14, 9)\end{aligned}$$

The exchange of a larger number of BBs leads to solutions represented by the objective vectors shown in Figure 6.2. Most of the objective vectors generated are dominated by extreme solutions. In fact, there is no way to guarantee only non-dominated solutions. Such a situation exemplifies what commonly occurs in practical problems.

6.2.3 Goals of Multiobjective Crossover

The first step for the generalization of the ideas described in Section 6.2.1-6.2.2 is to validate Proposition 6.2 without the use of explicit BBs. So far, lBBs have been considered as units that may generate highly-fit offspring when exchanged between parents. In this sense, BBs can be seen as the minimal units of information conceived from the parents to the offspring or, in other words, the units of information that relate the decision and objective spaces.

Without relying on the BB identification, Proposition 6.2 can be generalized by substituting BBs for the expected number of bits transmitted from each parent to the offspring. Although a BB of size k should contain more information about an objective function than k randomly chosen bits, this relaxation enables generalization. Consider a solution $\mathbf{x} \in \mathcal{G}$ generated by crossover between two parents $\mathbf{x}_1, \mathbf{x}_2$ and the respective objective vectors $\mathbf{z}, \mathbf{z}_1, \mathbf{z}_2 \in \mathcal{P}$.

Conjecture 6.4. *The expected position of $\mathbf{z}$ depends on how much information (in terms of bits) $\mathbf{x}$ preserves from its parents to solve each objective, i.e. $\mathbf{z}$ tends to be in the neighborhood of $\mathbf{z}_1$ if more information is preserved from $\mathbf{x}_1$ than from $\mathbf{x}_2$ and vice-versa.*

In other words, if $\mathbf{x}$ becomes closer to $\mathbf{x}^*$ in the decision space, $\mathbf{z}$ and $\mathbf{z}^*$ are more likely to be close in the objective space. Since this might not always be true, this is not an exact statement, however, it is an useful relaxation towards the understanding of crossover for MOPs.

Proposition 6.4 holds for the Pareto-optimal set (6.3), i.e. a solution in the middle of the front is equidistant to $\mathbf{x}^*$ and $\mathbf{y}^*$, in the decision space. Now, assume an optimal solution $\mathbf{x}^* \in \mathcal{G}$ for a general objective function f_1 and an optimal solution $\mathbf{y}^* \in \mathcal{G}$ for another objective function f_2 . If a small subset of decision variables is randomly chosen and their values in $\mathbf{y}^*$ set in $\mathbf{x}^*$, producing a new solution $\mathbf{x}$, the new solution objective vector $\mathbf{z}_x$ is likely to be closer to $\mathbf{z}_{x^*}$ than to $\mathbf{z}_{y^*}$. In some cases, if the subset exchanged conveys plenty of information about one objective, $\mathbf{z}_x$ might not be so close to $\mathbf{z}_{x^*}$. However, this is an unlikely situation in traditional crossover operators where bits are selected randomly.

The above discussion provides an indirect explanation about why restricted mating (in the objective space) has been useful to improve the diversity in MOEAs. If recombination is restricted to neighboring parents in the objective space, the offspring is more likely to be close of its parents. Since all non-dominated solutions have similar fitness, restricted mating favors a “more” uniform exploration of the whole extension of the Pareto-fronts. Relying on

Definition 2.7, this characteristic can be generalized as a requirement for effective crossover operators in MOPs.

Definition 6.1 (Uniformity-preserving operators for MOPs). A variation operator V is uniformity-preserving in the phenotype space if, given a population P with of objective vectors $\mathcal{F} \sim \text{Unif}(\mathcal{P}^{|P|})$, it produces a population $P' = V(P)$ with objective vectors $\mathcal{F}' \sim \text{Unif}(\mathcal{P}^{|P'|})$.

In other words, an uniformity-preserving operator in the phenotype space does not introduce any bias in the objective vectors generated. Conversely, a variation operator is biased if it is not uniformity-preserving. The *reproductive bias* is measured from the distribution of the objective vectors produced by the operator and an unbiased variation operator is defined as follows.

Definition 6.2 (Unbiased multiobjective variation). Consider an M -objective problem and a set of weight vectors $w \in \mathbf{W}$ uniformly distributed in the objective space. A variation operator $V : P \mapsto P'$ is unbiased in the objective space if the offspring it generates induce objective vectors uniformly distributed among the aggregations defined by $\mathbf{W}$.

In other words, an unbiased operator must produce a population P' whose objective vectors are uniformly distributed among the possible aggregations $\mathbf{W}$. Figure 6.3 illustrates the difference between an unbiased crossover and a biased one towards the center of the Pareto-optimal front. The shaded regions delimit the hypothetical region of convergence of both operators, where the unbiased version is expected to achieve much better diversity. MOEAs that explicitly handle aggregation functions, implement unbiased crossover operators, restricted mating or clustering in the objective space also approximate unbiasedness.

Figure 6.3: Points representing the offspring generated by unbiased (a) and biased (b) crossovers. The unbiased version exploits the regions of all possible aggregations uniformly and enables the exploration of a more well-spread set of objective vectors (shaded region).

Section 6.2.4 investigates the bias of reproduction operators like n -point, p -uniform and linkage-guided crossover.¹

¹Do not confuse the unbiased multiobjective crossover with the notion of unbiased variation operators by Lehre and Witt (2012), which is defined on the decision space.

6.2.4 Reproductive Bias of Crossover

Consider the Non-dominated Sorting Genetic Algorithm II (NSGA-II) with p -uniform crossover ($p = 0.5$), standard mutation ($p = 1/N$) and no restricted mating or niching in the decision space. The p -uniform crossover generates a child $\mathbf{y}$ from two parents $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{G}$. Each bit y_i , $i \in [N]$ comes from parent $\mathbf{x}_1$ with probability p , or parent $\mathbf{x}_2$ with probability $1 - p$. After crossover, the number of bits from $\mathbf{x}_1$ composing $\mathbf{y}$ is a random variable $\mathcal{B}_p$ with the expected value:

$$\mathbb{E}(\mathcal{B}_p) = \sum_{i=1}^N p = N \cdot p \quad (6.4)$$

If $p = 0.5$, $\mathbf{y}$ is, on average, distant $N/2$ bits from each parent in the decision space. Now, suppose $\mathbf{x}_1$ is a nearly-optimal solution according to an objective and $\mathbf{x}_2$ is a nearly-optimal solution for another objective. Since $\mathbf{y}$ will have half of the genetic material of each parent, it is expected to have only half of their fitness. In this case, $\mathbf{z}_y$ is biased towards the center of the objective space. Although NSGA-II's crowding may attribute extra fitness to solutions $\mathbf{x}_1, \mathbf{x}_2$ to favor extreme regions, the region in the middle of the current front will be more intensely exploited and extreme solutions are likely to be lost. Along the generations, the population will then converge to the middle region of the objective space causing a decrease of diversity.

As also observed by Sato et al. (2013) and Ishibuchi et al. (2014), by increasing (or decreasing) the probability p of uniform crossover, the expected position of $\mathbf{z}_y$ will change accordingly. To illustrate the influence of p on the expected positioning of $\mathbf{z}_y$, suppose choosing parents $\mathbf{x}_1, \mathbf{x}_2$, such that $\mathbf{x}_1$ is better than $\mathbf{x}_2$ according to the first objective. Then, for each $0 \leq p \leq 1$, the

Figure 6.4: Positioning of objective vectors obtained by different values for $0 \leq p \leq 1$. The expected distance between parents and offspring induces a bias in the population along the generations.

NSGA-II is ran for a certain number of generations. Observe, low or high p values privilege the search in the extreme regions of the objective space and decrease the search in the middle region.² Such a correlation between p and the positioning of the objective vectors generated is a

²If the parents are ordered, the resulting offspring will be symmetrical around the middle region, i.e. $p \equiv 1 - p$.

great advantage of the use of uniform crossover, as it enables a choice between a large spreading (low/high values of p) or a search focused in the middle region ($p \approx 0.5$).

If, otherwise, an n -point crossover is used, the situation is quite different. For example, consider an n -point crossover with an odd n . As the parents are cut, an even number of subsections is created, therefore, solution $\mathbf{y}$ will be composed of the same number of subsections from each parent. Although the size of these subsections can vary in each recombination, the expected number of bits from each parent is the same as that of $n = 1$. Below is the expected number of bits from each parent with $n = 1$, there are $N - 1$ possible crossover points:

$$(N - 1) \left\{ \begin{array}{ll} \underbrace{x_1, x_2, x_3, x_4, x_5, \dots, x_{N-1}, x_N}_{(s=1)} \\ \underbrace{x_1, x_2, x_3, x_4, x_5, \dots, x_{N-1}, x_N}_{(s=2)} \\ \dots \\ \underbrace{x_1, x_2, x_3, x_4, x_5, \dots, x_{N-1}, x_N}_{(s=N-1)} \end{array} \right.$$

Each possible crossover point occurs with equal probability $1/(N - 1)$ and transmits s bits to the offspring. Therefore, the expected number of transmitted bits is given by

$$\mathbb{E}(\mathcal{B}_{n=1}) = \sum_{s=1}^{N-1} s \cdot (N - 1)^{-1} = \frac{N}{2} \quad (6.5)$$

With an even number n of crossover points, parents will be cut into an odd $(n + 1)$ number of subsections. For $n = 2$, there are $\binom{N-1}{2}$ possible different ways of choosing the two crossover points, each of them with equal probability $1/\binom{N-1}{2}$. There are $(N - s - 1)$ different configurations that transmit $s = 1$ bit from one parent to the offspring, therefore

$$(N - s - 1) \left\{ \begin{array}{ll} \underbrace{x_1, x_2, x_3, x_4, x_5, \dots, x_{N-1}, x_N}_{(s=1)} \\ \underbrace{x_1, x_2, x_3, x_4, x_5, \dots, x_{N-1}, x_N}_{(s=1)} \\ \dots \\ \underbrace{x_1, x_2, x_3, x_4, x_5, \dots, x_{N-1}, x_N}_{(s=1)} \end{array} \right.$$

Following the same idea for $1 \leq s \leq (N - 1)$, there are $(N - s - 1)$ ways of transmitting s bits, each of them occurring with the same probability. Therefore, the expected number of bits transmitted from each parent is equal to

$$\mathbb{E}(\mathcal{B}_{n=2}) = \sum_{s=1}^{N-1} s(N - s - 1) \cdot \binom{N-1}{2}^{-1} = \frac{N}{3} \quad (6.6)$$

$$= \frac{2N}{3} \quad (6.7)$$

In words, a solution $\mathbf{y}$ will be composed (on average) of $N/3$ bits from one parent and $2N/3$ from the other. As n assumes larger even values, the subsections will decrease in size, hence $\mathbb{E}(\mathcal{B}_n) \approx N/2$. Figure 6.5 shows the number of bits from each parent in both p -uniform and n -point crossover operators obtained experimentally. In summary, $\mathbb{E}(\mathcal{B}_n) \approx N/2$ for all odd n . For even values of n , $\mathbb{E}(\mathcal{B}_n)$ decreases to $\mathbb{E}(\mathcal{B}_n) \approx N/2$ as n grows. Therefore, n -point crossover has a strong bias towards the middle region which is not possible to be circumvented by changing n .

Figure 6.5: Number of bits from each parent in crossover. For any odd number of points, half the bits are expected to come from each parent. For even number of points, the value decreases exponentially and tends to $N/2$. In uniform crossover, $\mathcal{B}_n$ varies linearly according to p .

On the other hand, the p -uniform crossover is much more flexible and, by changing p , the focus on diversity/spreading or convergence is modified. A small value of p , say $p = 0.1$, will generate solutions composed of 90% of bits from one of the parents. If such a parent is close to an extreme region, in half the cases $\mathbf{y}$ is also expected to be close to that extreme region. Otherwise, if $p \approx 0.5$, $\mathbf{y}$ will be, on average, at halfway to the parents. If only one of the parents is in an extreme region or in distinct extreme regions, $\mathbf{y}$ will more likely be pushed towards the middle region. If both parents are in the same region, the value of p is not so relevant. Besides, by using a small/large value for p the chances of $\mathbf{y}$ being close to one of its parents in the objective space increases, which favors uniformity-preservation.

Such a characteristic bias of n -point crossover has a negative effect on the multiobjective evolutionary optimization because it makes more difficult to keep the diversity of objective vectors. To the best of my knowledge, no study has investigated why $(n = 1, 2)$ -point and $(p = 0.5)$ -uniform crossovers show such an unwanted bias. Ishibuchi and Shibata (2003a,b) have extensively studied the impact of crossover on diversity and convergence, but in most cases conclusions were made without considering the inherent bias of crossover (Ishibuchi et al., 2010; Ishibuchi and Narukawa, 2005; Ishibuchi and Shibata, 2003a,b; Ishibuchi et al., 2007, 2008a, 2012). The analysis of crossover bias explains why mating restrictions help to increase diversity of solutions and evidences the importance of unbiased crossover operators.

6.2.5 Handling Bias with Restricted Mating

Decomposition-based approaches and MOEDAs that use multiple models in the objective space can all be characterized as approaches implementing restricted mating. Such methods have consistently enabled improvements in the diversity of Pareto fronts found by MOEAs, the concept of crossover bias can be used to justify their benefits.

In the usual scenarios in which MOEAs are evaluated, the n -point or p -uniform crossover operators are most commonly applied with $n = 1, 2$ and $p = 0.5$. It was already shown that in such configurations there is a strong bias towards the middle region of the Pareto front. Since restricted mating limits reproduction to neighboring solutions, it works as a way of balancing the exploration of these different possible biases. In other words, each subpopulation defined by neighboring solutions is associated with the implicitly defined aggregation vector w that induces a single objective problem in the same region. Since there are multiple implicitly defined subpopulations and each of them produces their own bias, multiple directions in the objective space are explored simultaneously. Figure 6.6 illustrates the idea of multiple biases.

Figure 6.6: Illustrative example of the impact of restricted mating on the creation of multiple bias in the objective space and improvement of the spreading of objective vectors.

Without restricted mating (Figure 6.6(a)), solutions can reproduce independently of their distance in the objective space, therefore, the whole population is expected to collapse towards its middle region. With restricted mating (Figure 6.6(b)), only solutions from the same subregion can reproduce, which induces a bias for each subregion.

In summary, restricted matings decrease the bias of crossover and enables a “more” uniform exploration of the objective space. However, unbiased crossovers produces solutions uniformly spread along the current non-dominated front and fewer solutions populate each subregion, therefore, convergence towards the Pareto-optimal front tends to be slower.

6.2.6 Handling Bias with Linkage-learning

As previously mentioned in Chapter 3, no unique linkage model enables a non-disruptive crossover in complex optimization problems. Therefore, different linkage models eventually produce offspring of similar characteristics. Let us assume a linkage model $\mathcal{T}$ based on disjoint subsets of variables and a parameter $\Psi \geq 2$ that defines the number of subsets to be identified by linkage-learning, with $\Psi = 2$ being the least disruptive option.

To understand how the use of linkages impacts on the crossover bias (and the expected number of bits transferred, $\mathbb{E}(\mathcal{B}_\Psi)$), assume $\Psi = 2$ and a simple linkage model composed of two disjoint subsets $\mathcal{T} = \{\tau_1, \tau_2\}$, such that $|\tau_1| + |\tau_2| = N$. If these two subsets are approximately the same size, $\mathbb{E}(\mathcal{B}_\Psi) \approx N/2$ and a strong bias towards middle regions is expected. On the other hand, if τ_1 is small in comparison to τ_2 , a bias towards extreme regions is expected. When a large $\Psi \leq N$ is used, the subsets decrease in size and $\mathbb{E}(\mathcal{B}_\Psi) \approx N/2$. In summary, a small Ψ is similar to a small p in uniform crossover, whereas a large Ψ is similar to $p \approx 0.5$. This explanation does not depend on what is being modeled by linkage-learning, instead it only relies on $\mathbb{E}(\mathcal{B}_\Psi)$.

To effectively understand the role of linkage-learning in MOPs, the information being extracted from the population must be investigated. In single-objective optimization, linkage-learning is usually described as a method that identifies substructures of the problem being solved. However, Chapters 3-4 showed linkage-learning is not so robust concerning multimodality and Section 6.2.1 showed massive multimodality emerges even in simple bi-objective problems. Therefore, it is unlikely for linkage-learning to identify the substructures composing MOPs. Since linkage-learning identifies subsets of variables relying on statistics measured from the population, it groups variables that systematically assume few patterns of values in the current population. It may be hypothesized that linkage-learning models the correlation among the objectives. In the case of a high correlation $0 \leq \rho \leq 1$, many variables agree on their values, i.e. if a solution is improved according to an objective function, its quality according to other objective functions is likely to improve. In the case of negative correlations $-1 \leq \rho < 0$, if a solution is improved according to an objective function, its quality according to the others is likely to decrease. If this interpretation is correct, with $\Psi = 2$ one of the subsets $\{\tau_1, \tau_2\}$ is expected to be much larger than the other for highly positively/negatively correlated objectives and of similar size when $\rho \approx 0$.

6.3 Results and Discussion

Three hypotheses have been raised in this chapter. First, it was argued that the average positioning of an offspring in the objective space depends on how much information is conveyed from each parent. Second, it was showed theoretically that the n -point crossover bias is barely affected by the number of crossover points n and p -uniform crossover bias depends on p . Third, it

was hypothesized that linkage-learning models the correlation between objectives instead of substructures of each objective. The three hypotheses were evaluated experimentally, p -uniform, n -point and linkage-guided crossovers were implemented in NSGA-II with standard mutation, the results are discussed in this section.

Section 6.3.1 describes the ρMNK -landscapes model from which instances were generated with different degrees of multimodality and correlation. Section 6.3.2 evaluates the two first hypotheses through the comparison of the biases of n -point and p -uniform crossovers with different values of n and p . Section 6.3.3 reports on the validation of Proposition 6.4 through the implementation of an adaptive version of p -uniform crossover, in which p depends on the crowding distance of the parents. In Section 6.3.4 the subsets produced during linkage-learning are analyzed to relate ρ and the resulting linkage models.

6.3.1 The ρMNK -landscapes Problem

Multi-objective MNK -landscapes were proposed by Aguirre and Tanaka (2004, 2005), who studied the performance of one-bit climbers. The impact of problem size N , degree of non-linearity K and number of objectives M on the instances were also analyzed by Aguirre and Tanaka (2007). Extending these studies, Verel et al. (2011a,b, 2013) proposed and analyzed the fitness landscapes of MNK -landscapes with different degrees of correlation, ρ , among objective functions (ρMNK -landscapes). Verel et al. (2013) also provided an instance generator, from which instances with varying correlation (ρ), epistasis (K), dimensionality (N) and number of objectives (M) can be generated.³

$$\max f_{\rho MNK}^j(\mathbf{x}), \forall j \in [M] \quad (6.8)$$

Bi-objective ($M = 2$) instances of size $N = 500$ and epistasis degrees $K = 2, 4, 8, 12$ were generated. As discussed by Verel et al. (2013), instances with negative correlations produce wider Pareto-fronts appropriate for the evaluation of diversity preservation mechanisms, whereas instances with zero or higher correlation produce Pareto-fronts of shorter extension and more distant from the initial population. Since diversity preservation is the main concern, results are shown for instances with $\rho = -0.9, 0$ correlation. For each instance, 30 runs were performed, with a limit of 500,000 fitness evaluations (i.e. 250,000 solutions for $M = 2$).⁴

6.3.2 Traditional Crossover bias

This section addresses a comparison of traditional uniform crossover, with $p = 0.5$, and other values of $0 \leq p \leq 0.5$ to illustrate the different biases obtained by varying p . Figure 6.7 shows the EAFs for instances with $\rho = -0.9$.

³<http://mocobench.sourceforge.net/>

⁴The results were compared by the Empirical Attainment Function (EAF) (Fonseca et al., 2011) computed with

Figure 6.7: Comparison between uniform crossover with $p = 0, 0.1, 0.2, 0.3, 0.4, 0.5$ and $p = 0.5$ for instances with $\rho = -0.9$. Dark region represents the better performance of every parameterization.

With $p = 0$, no crossover was applied and NSGA-II faced difficulties in exploring the middle region of the Pareto-front, however, it showed a much better performance in extreme regions. With $p = 0.1$, on the other hand, the Pareto-fronts found were mostly equivalent to the results of $p = 0.5$ in the middle region, but with advantages in the extreme regions. With $p = 0.2$, the results in extreme regions were still better, without disadvantages in the middle region. With $p = 0.3, 0.4$, the results were very similar to $p = 0.5$.

Figure 6.8: Comparison between uniform crossover with $p = 0, 0.1, 0.2, 0.3, 0.4, 0.5$ and $p = 0.5$ for instances with $\rho = 0$. Dark region represents the better performance of every parameterization.

Figure 6.8 shows the results for instances with $\rho = 0$. These instances have a less extensive Pareto-optimal front and the importance of diversity preservation decreases. Therefore, only for $p = 0, 0.1$ some benefits were observed in the extreme regions. In highly correlated instances, convergence is more important than diversity and $p = 0.5$ is the best choice.

6.3.3 Crowding-biased Uniform Crossover

In order to handle crossover bias and at the same time synchronize the goals of crowding and reproduction, parameter p in uniform crossover must be adapted according to the characteristics of the parents undergoing crossover. In NSGA-II, the crowding distance $d(\mathbf{x})$ associated with every solution $\mathbf{x} \in P(t)$ at generation $t > 0$ describes how isolated the objective vector $\mathbf{z}_\mathbf{x}$ is in the objective space. By defining p according to the crowding distance of its parents $\mathbf{x}_1$ and $\mathbf{x}_2$ (respectively, $d(\mathbf{x}_1)$ and $d(\mathbf{x}_2)$), offspring can be generated closer to the most isolated parent.

NSGA-II performs the non-dominated sorting in a population that combines parents $P(t)$ and offspring $P_{\text{offspring}}$. The solutions in the first fronts are inserted in the next population $P(t+1)$ and the crowding distance is computed only for solutions in fronts that do not fit the population as a whole. In our case, the crowding distance is used during crossover, therefore, it is computed for all fronts.

Crowding distance $d(\mathbf{x}) \in [0, \infty)$, $\forall \mathbf{x} \in P(t)$, with infinity being attributed to the extreme solutions in each front. On the other hand, the uniform crossover parameter is $p \in [0, 0.5]$. In order to implement a uniform crossover biased towards solutions in less crowded regions, assume a reference parent $\mathbf{x} : d(\mathbf{x}) > d(\mathbf{y})$ and then scale the crowding distance such that $d(\mathbf{x}) \in [0, 0.5]$. Algorithm 6.1 describes how a mapping can be defined and used to implement a crowding-biased uniform crossover.

Algorithm 6.1 Crowding-biased uniform crossover

Require: Two solutions $\mathbf{x}, \mathbf{y} \in P(t)$ ordered such that their crowding distances $d(\mathbf{x}) > d(\mathbf{y})$ and the lower and upper limits for $\min_p \leq p \leq \max_p$, e.g. $0.5 \leq p \leq 0.9$.

Ensure: A solution $\mathbf{u}$ closer to its most isolated parent.

```

 $p \leftarrow 0.5$ 
 $\Delta_d \leftarrow \max_{\mathbf{v} \in P(t)} d(\mathbf{v}) - \min_{\mathbf{v} \in P(t)} d(\mathbf{v})$   $\triangleright$  Compute the range of  $d$  in  $P(t)$ , in generation  $t$ .
 $\Delta_p \leftarrow \max_p - \min_p$   $\triangleright$  Compute the range of  $p$ 
if  $d(\mathbf{x}) < \infty$  then
     $p \leftarrow \min_p + \frac{(d(\mathbf{x}) - \min_d) \cdot \Delta_p}{\Delta_d}$   $\triangleright$  Scale  $d(\mathbf{x})$  between  $\min_p$  and  $\max_p$ 
else
     $p \leftarrow \max_p$   $\triangleright$  If  $d(\mathbf{x}) = \infty$  then  $p$  is maximum
 $\mathbf{u} \leftarrow \mathbf{x}$   $\triangleright$  Copy  $\mathbf{x}$  into  $\mathbf{u}$ 
for  $i \in \{1, \dots, N\}$  do
    if  $\text{random}(0, 1) > p$  then  $\triangleright$  Substitute  $(1 - p) \cdot N$  bits of  $\mathbf{u}$  for bits of  $\mathbf{y}$ 
         $u_i \leftarrow y_i$ 

```

return $\mathbf{w}$

In words, taking into account the crowding distances of solutions in the current population $P(t)$, for each pair of parents $\mathbf{x}$ and $\mathbf{y}$, a p is defined to bias the offspring towards the most isolated parent. If $\mathbf{x}$ is an extreme solution, with $d(\mathbf{x}) = \infty$, the resulting offspring will be composed, on average, of $\max_p \cdot N$ bits from $\mathbf{x}$. Otherwise, if $d(\mathbf{x}) = \min_d$, $\mathbf{x}$ is in a crowded region (or the crowding distance has not been computed for $\mathbf{x}$), therefore, $p = \min_p$ is applied.

Figure 6.9 shows the comparison between $p = 0, 0.1, 0.2, 0.3, 0.4, 0.5$ uniform crossover and the crowding-biased uniform crossover for instances with $\rho = -0.9$. Without crossover and with $p = 0$, NSGA-II performed slightly better in the extreme regions, but much worse in the middle region. Uniform crossover with $p = 0.1, 0.2, 0.3, 0.4, 0.5$ performed slightly better in the

Figure 6.9: Comparison between the crowding-biased uniform crossover (NSGA2-self) and traditional uniform crossover with $p = 0, 0.1, 0.2, 0.3, 0.4$ (NSGA2-p=*) in instances with $\rho = -0.9$. Darker regions represent the better performance of each algorithm.

middle region than its crowding-biased counterpart. However, as shown in the dark regions, the improvements in spreading achieved by the crowding-biased crossover were far more evident. In

summary, if a well-spread Pareto-front is desirable, a better spreading can be achieved by biasing the uniform crossover towards less populated regions.

Figure 6.10 shows the results for instances with $\rho = 0$. As expected, by increasing the correlation between objectives, the benefits of diversity preservation decreases. However, in most cases, extreme regions were better explored by the crowding-biased uniform crossover than p -uniform crossover without loss of convergence in the middle region.

Figure 6.10: Comparison between the crowding-biased uniform crossover (NSGA2-self) and traditional uniform crossover with $p = 0, 0.1, 0.2, 0.3, 0.4$ (NSGA2-p=*) in instances with $\rho = 0$. Darker regions represent the better performance of each algorithm.

6.3.4 Linkage-guided Crossover bias

In Section 6.2.6, it was hypothesized that $\Psi = 2$ would lead to linkage models $\mathcal{T} = \{\tau_1, \tau_2\}$ in which the sizes of τ_1 and τ_2 would be correlated to ρ . Besides, it was suggested that the difference in size between τ_1 and τ_2 would affect crossover bias $\mathbb{E}(\mathcal{B}_\Psi)$. This section shows results for both hypotheses, first the sizes of τ_1 are measured during the experiments and, second, the results of the linkage-guided crossover are evaluated. Figure 6.11 shows how the sizes of τ_1 were distributed along the experiments.

Figure 6.11: With $\Psi = 2$ and linkage models $\mathcal{T} = \{\tau_1, \tau_2\}$, these figures show the size distribution of τ_1 along the generations in instances with $\rho = -0.9, -0.5, 0, 0.5, 0.9$ and $K = 2, 4, 8, 12$.

Since $|\tau_1| + |\tau_2| = N$, the distribution of τ_2 can be inferred by symmetry. In instances with negatively correlated objectives, $\rho = -0.9, -0.5$, τ_1 usually contains a small number of decision variables, probably representing the agreement between the objectives. On the other hand, in instances with positively correlated objectives, $\rho = 0.5, 0.9$, τ_1 groups most of the decision variables. As K increases, the variance in the distributions also increases, which might be a result of massive multimodality and less accurate linkage-learning (as explained in Chapter 3). In summary, there is a clear correlation between ρ and $|\tau_1|$, i.e. as ρ grows, $|\tau_1|$ also grows.

Figure 6.12: Comparison between the linkage-guided crossover (NSGA2-II-n= Ψ) and traditional uniform crossover with $p = 0.5$ (NSGA2-p=0.5) in instances with $\rho = -0.9$. Darker regions represent the better performance of each algorithm.

From such a result, it is concluded that linkage-learning models, to some extent, the correlation between the objectives. Therefore, linkage-guided reproduction should be more effective in exploiting extreme regions than traditional crossover operators because it would preserve more information during crossover than randomized variation operators.

Figure 6.12 shows a comparison of the EAFs generated by the linkage-guided crossover (Algorithm 3.1) with $\Psi = 2, 9, 17, 33, 65, 257$ and uniform crossover with $p = 0.5$ in instances with $\rho = -0.9$. As expected, spreading considerably decreased as Ψ increased. In the extreme

Figure 6.13: Comparison between the linkage-guided crossover (NSGA2-ll-n= Ψ) and traditional uniform crossover with $p = 0.5$ (NSGA2-p=0.5) in instances with $\rho = 0$. Darker regions represent the better performance of each algorithm.

case, with $\Psi = 257$, there was no practical benefit of using the linkage-learning instead of uniform crossover. These results have confirmed the hypothesis that as Ψ increases, the expected number of bits from each parent approaches $\mathbb{E}(\mathcal{B}_\Psi) \approx N/2$ and the use of linkage-guided crossover produces the same bias of uniform crossover with $p = 0.5$.

Figure 6.13 shows a comparison of the EAFs generated by linkage-guided crossover with $\Psi = 2, 9, 17, 33, 65, 257$ and uniform crossover with $p = 0.5$ in instances with $\rho = 0$. Such instances have more convex and shorter Pareto-optimal fronts, therefore, diversity preservation is expected to exert little influence on their solution. However, the linkage-guided crossover still provided better exploitation of extreme regions with small values of Ψ and as Ψ increased such benefits vanished.

Although the previous results show the benefits of using linkage-learning, it is not clear how linkages helped to keep a better diversity. Figure 6.11 suggests linkage-learning identify a consensus between the objectives and decrease crossover bias. In order to understand the behavior of linkage-guided crossover, the positioning of the offspring in the objective space (in relation to their parents) was analyzed. Whenever a solution $\mathbf{y}$ is produced from parents $\mathbf{x}_1$ and $\mathbf{x}_2$, the parent whose objective vector is the closest to $\mathbf{z}_y$ is identified (let us assume it is $\mathbf{x}_1$). By assuming $\mathbf{z}_{\mathbf{x}_1}$ as the origin of a coordinate system, the position of $\mathbf{z}_y$ relative to $\mathbf{z}_{\mathbf{x}_1}$ provides a vector linking a parent to the child in the objective space, called *reproduction vector*. Figure 6.14 illustrates the reproduction vectors generated for instances with $\rho = -0.9$ and $K = 2, 4, 8, 12$.

Figure 6.14: Reproduction vectors generated by linkage-guided and uniform crossovers, for instances with $\rho = -0.9$. The color of a point represents its distance to the closest parent, darker red implies more distant parents.

The reproduction vectors are represented by points in the graphic, whereas the distance between the parents is represented by the red color intensity, i.e. darker red indicates more distant parents. The difference between the operators is evident, linkage-guide crossover, with $\Psi = 2$, always produces offspring close to one of the parents, while uniform crossover, with $p = 0.5$, produces offspring close to the parents only if the parents were already close.

Figure 6.15 illustrates the reproduction vectors produced by linkage-guided and uniform crossovers for instances with $\rho = 0$ and $K = 2, 4, 8, 12$. The results are similar to those of the previous case and linkage-guide crossover (with $\Psi = 2$) always produced offspring closer to the parents. However, if p is decreased from 0.5 to 0.1 in uniform crossover, the offspring approaches one of the parents and different regions of the Pareto-front are better explored.

Figure 6.15: Reproduction vectors generated by linkage-guided and uniform crossover, for instances with $\rho = 0$. The color of a point represents its distance to the closest parent, darker red implies more distant parents.

In both cases, $\rho = -0.9, 0$, linkage-guided crossover is more robust to K than p -uniform crossover. In the p -uniform crossover, as K increases, the number of reproduction vectors in the opposite direction (3^{th} quadrant) of the Pareto-optimal front considerably increases, the same do not occur when linkage-learning is used.

These results show if crossover generates offspring close to one of the parents, the whole Pareto-front is more likely to be explored uniformly, i.e., it decreases crossover bias. This hypothesis was proved by the results with p -uniform crossover and confirmed in the case of linkage-guided crossover with $\Psi = 2$. In both situations, a better spread of Pareto-front was achieved at the end of the search, however, linkage-learning was more effective. Figure 6.16 shows the disruptiveness of crossover increases as Ψ grows, which favors convergence speed.

Figure 6.16: Reproduction vectors generated by linkage-guided with $\Psi = 9, 17, 33$, for instances with $\rho = -0.9$. The color of a point represents its distance to the closest parent; darker red implies more distant parents.

6.3.5 Measuring Crossover Bias

Definition 6.2, described the ideal characteristics of a crossover operator for MOEAs and the results provided in the previous sections endorsed the importance of unbiased variation operators. This section summarizes the results and provides a straight way to compare the bias of the various crossover operators previously evaluated.

Given a set of weight vectors $\mathbf{W}$, a crossover is said *unbiased in the objective space* if the offspring it produces is uniformly distributed according to $\mathbf{W}$, i.e., on average, the same number of offspring is generated in each of the regions covered by each aggregated single-objective function. At each generation $t > 0$, the number of solutions generated in each region must be accounted for the measuring of the bias of specific crossover operators. Since in practice $\mathbf{W}$ must be of bounded size $|\mathbf{W}| \leq \lambda$, approximations must be made so that the region in which an objective vector is associated can be identified.

Figure 6.17 illustrates the idea of how to measure the number of objective vectors in each subregion defined by $w \in \mathbf{W}$, where points in the same shaded region are associated with the same w . At each generation t , a nadir vector $z^{(nadir,t)}$ is estimated from $P(t)$ as $z^{(nadir,t)} = (\min_{z \in P(t)} z_1, \dots, \min_{z \in P(t)} z_M)$ and used as a reference point.

Figure 6.17: Illustrative example of weight vectors and the subregions they define. The white points are objective vectors whose ideal weight vectors w^* coincide with some $w \in \mathbf{W}$.

To every objective vector z there is an ideal weight vector w^* associated, which produces an aggregated single-objective function in the same positioning of z (white points in Figure 6.17 represent z vectors whose ideal weight vector w^* coincides with some $w \in \mathbf{W}$).

$$w_i^* = \frac{z_i - z_i^{(nadir,t)}}{\sum_{i=1}^M (z_i - z_i^{(nadir,t)})}, \quad \forall i = 1, \dots, M. \quad (6.9)$$

Such ideal weight vectors are then compared with the predefined $\mathbf{W}$. For our purposes it is enough to define $\mathbf{W} = \{w_i : w_i = (i/10, 1 - i/10), \forall i = 0, \dots, 10\}$. In this case, with $M = 2$,

each $w_i \in \mathbf{W}$ is uniquely identifiable from its first component, therefore, the closest w_i to any w^* can be computed as $i = \lceil w_1 \times 10 \rceil + 1$, where $\lceil \cdot \rceil$ is the nearest integer value. In other words, an objective vector z , with ideal weight vector w^* is within the region defined by the weight vector $w_{\lceil w_1^* \times 10 \rceil + 1}$. The number of objective vectors generated in each subregion characterize the bias of any specific crossover operator. Ideally, in unbiased operators, such a number should be uniformly distributed among the $\forall w \in \mathbf{W}$.

Figure 6.18 shows the number of times an offspring was generated in each subregion defined by $w \in \mathbf{W}$. Since every w can be uniquely identified by its first component, the graphic relates w_1 to the average number of solutions generated on w . Crossover bias for n -point, p -uniform crossovers endorsed the theoretical values derived in Section 6.2.4, in which n does not influence the bias but p does. On the other hand, Linkage-guided crossover bias was assessed for the first time and a strong bias towards the middle $w \in \mathbf{W}$ was observed only if Ψ is large.

Figure 6.18: Crossover bias averaged from 30 runs, each comprised of 2,500 generations and population size $\mu = 100$. The first row shows the results for n -point crossover with $n = 1, 2, 3, 6, 12$, the second row shows results for p -uniform crossover with $p = 0.01, 0.1, 0.3, 0.4, 0.5$ and the third results for linkage-guided crossover with $\Psi = 2, 9, 17, 65, 257$.

The least disruptive linkage-guided crossover, with Ψ , produced a small bias very similar to that produced by p -uniform crossover with $p = 0.01$. In both cases, Pareto-fronts with high spreading were found, which proves the importance of unbiasedness of variation operators in the objective space (see Definition 6.2).

6.4 Conclusions

This chapter addressed the analysis of the impact of linkage-learning on multi-objective optimization from the perspective of reproductive bias. By using simple bi-objective additively separable functions as a study case, the results of crossover were analyzed and the desirable characteristics of crossover operators suited for MOEAs were defined. Such characteristics were then evaluated in traditional reproduction operators and in a linkage-guided crossover.

The biases of n -point and p -uniform crossover were analyzed in terms of the offspring they produce. Independently of the parameter n , n -point crossover produces offspring approximately at half distance of its parents. In the case of p uniform crossover, the offspring tends to be closer to one of the parents, according to the value of p . Regarding the expected number of bits transmitted from parent to offspring, n -point crossover is mostly equivalent to uniform crossover with $p = 0.5$, but more limited in the sense that varying n do not produces much different results.

Such inherent bias has implications in the objective space, offspring is more intensely generated in the middle region of the Pareto-front and the spreading progressively decreases along the generations. It has been shown that a simple way to circumvent such a drawback is to use smaller values for $0 < p < 0.5$, which decreases the bias intensity and helps extreme regions to be better explored. This occurs because a small p implies the generation of offspring closer to the parents and helps to better explore all the regions of the Pareto-front.

It was also noted that one of the difficulties in diversity preservation is the incompatibility between the crowding mechanism and crossover results. While the first tries to guarantee a well-distributed front, the second ends up producing more offspring in the middle region. In order to circumvent such an incompatibility, a crowding-biased uniform crossover was proposed, which generates offspring closer to the more isolated parent. Experiments with instances of the ρMNK -landscapes model showed the efficacy of this simple approach in increasing diversity and spreading of the Pareto-fronts produced.

In the context of linkage-learning, great benefits in terms of diversity preservation were identified. The results suggest that linkage-learning models the consensus among the objectives, therefore, the linkages identified depend on the correlation between the objective functions. Furthermore, linkage-guided crossover tends to generate offspring more closer to one of the parents than any other crossover evaluated. Such ability seems to be a key point in favor of linkage-learning.

In summary, the key point of a crossover well-suited for multi-objective optimization is the uniformity-preservation in the objective space. Such characteristic enables the exploration of all parts of the Pareto-front uniformly, avoiding specific biases. This idea has already been explored in the literature in the context of restricted mating, but it was also showed that it can be achieved by using small p values in the uniform crossover or by adapting p to generate offspring closer to more isolated parents (crowding-biased uniform crossover) or by using linkage-guided crossover with few linkage sets.

Chapter 7

Conclusions

This thesis addressed the analysis of linkage-learning in the context of single- and bi-objective problems, regarding the optimization of pseudo-Boolean functions. The main goal was to contribute to the understanding of the limitations of linkage-learning, and Estimation of Distribution Algorithms (EDAs), in the context of single- and bi-objective evolutionary optimization.

Chapter 3 described theoretical results, in terms of additively separable functions, which demonstrated the influence of multimodality on the accuracy of linkage-learning. It was shown that the increase of multimodality makes the observation of statistical dependencies in the population difficult, which is an impediment for obtaining useful linkage-models. It was argued that *allelic-pairwise independence* might occur as a result of extreme multimodality.

Chapter 4 also investigates the impact of multimodality on linkage-learning accuracy, but from a more practical perspective. If the accuracy of linkage models is restricted by multimodality, there must be a range of problems for which linkage-learning is useful. This hypothesis was investigated considering non-separable instances of the NK -landscapes model with different degrees of multimodality (defined by K). Regarding *random linkages*, the range in which linkage-learning provided the best benefits encompasses the $1 \leq K \leq 6$ interval. On the other hand, there is a large region in which linkage-learning produced reasonable benefits. For example, for large instances ($N = 1000$), linkage-learning was reasonably useful in the large $1 \leq K \leq 13$ interval. However, for small instances ($N = 100$), linkage-learning was useful only in the small region delimited by the $1 \leq K \leq 5$ interval. Regarding *nearest-neighbor linkages*, regions were well delimited, independently of the problem size (N), and in all cases linkage-learning was of great utility in the $1 \leq K \leq 9$ interval. These results demonstrate that *random linkages* induces harder instances than *nearest-neighbor linkages*.

Chapter 5 analyzed the well-known Multidimensional Knapsack Problem (MKP) in order to exemplify the difficulties of applying linkage-learning in a practical problem. Results, in terms of effectiveness, were discussed and the linkage models built analyzed according to specific properties of the MKP. By evaluating the role of linkage-learning in the performance of the Linkage Tree Genetic Algorithm (LTGA), it was noted that random-linkage models provided similar results in terms of the number of fitness evaluations, solution quality and expected running-

time. In a broader perspective, the comparisons made clear the weakness of linkage-learning in helping to solve the MKP. The linkage-trees were analyzed using information-theoretic measures and results showed they were primarily composed of two giant clusters, separating converged and non-converged variables. Besides, only weak dependencies were measured among non-converged variables, which implies that the LTGA only identified low-confidence linkages and explains the similar results obtained by linkage-learning and random linkages.

Chapter 6 addressed the analysis of the impact of linkage-guided crossover operators in bi-objective optimization. First, the desirable property of reproduction operators was defined as the *uniformity-preservation* in the phenotype space. According to this property, traditional crossover operators were shown to be biased in the phenotype space and linkage-guided crossover was shown to approach uniformity preservation. The results suggested that linkage-learning models the consensus among the objectives, therefore, the linkages identified depend on the correlation between the objective functions. Furthermore, linkage-guided crossover tends to generate offspring closer to one of the parents than any other crossover evaluated. Such an ability seems to be a key point in favor of linkage-learning.

In summary, multimodality was shown to impose serious limitations for linkage-learning based on bivariate statistics, in *single-objective optimization*. Since NP-hard optimization problems are expected to be highly multimodal, new EDAs must, somehow, conciliate model-building and high-order operators in order to circumvent their limitations. In *bi-objective optimization*, the population was shown to contain statistical information about the objective correlations, which were useful to achieve a better exploration of the objective space. However, such characteristic may not scale well with the number of objectives and must be investigated. The use of *reinforcement learning* techniques is an alternative which provides ways to hybridize EDAs and local-search methods in more robust algorithms.

Bibliography

- AGUIRRE, H. E. and TANAKA, K. (2004), Effects of elitism and population climbing on multiobjective MNK-landscapes, in: *IEEE Congress on Evolutionary Computation (CEC'04)*, vol. 1, IEEE, pp. 449–456, DOI: [10.1109/CEC.2004.1330891](https://doi.org/10.1109/CEC.2004.1330891).
- (2005), Selection, Drift, Recombination, and Mutation in Multiobjective Evolutionary Algorithms on Scalable MNK-Landscapes, in: *Evolutionary Multi-Criterion Optimization (EMO'05)*, vol. 3410, LNCS, Springer, pp. 355–369, DOI: [10.1007/978-3-540-31880-4_25](https://doi.org/10.1007/978-3-540-31880-4_25).
- (2007), Working principles, behavior, and performance of MOEAs on MNK-landscapes, *European Journal of Operational Research*, **181** (3): 1670–1690, DOI: [10.1016/j.ejor.2006.08.004](https://doi.org/10.1016/j.ejor.2006.08.004).
- AUGER, A. et al. (2012), Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications, *Theoretical Computer Science*, **425** (0): 75–103, DOI: [10.1016/j.tcs.2011.03.012](https://doi.org/10.1016/j.tcs.2011.03.012).
- AZAD, M. A. K., ROCHA, A. M. A., and FERNANDES, E. M. (2014), Improved binary artificial fish swarm algorithm for the 0–1 multidimensional knapsack problems, *Swarm and Evolutionary Computation*, **14** (0): 66–75, DOI: [10.1016/j.swevo.2013.09.002](https://doi.org/10.1016/j.swevo.2013.09.002).
- BACK, T. (1994), Selective pressure in evolutionary algorithms: a characterization of selection mechanisms, in: *IEEE Congress on Evolutionary Computation (CEC'94)*, vol. 1, IEEE, pp. 57–62, DOI: [10.1109/ICEC.1994.350042](https://doi.org/10.1109/ICEC.1994.350042).
- BADER, J. and ZITZLER, E. (2010), HypE: An algorithm for fast hypervolume-based many-objective optimization, *Evolutionary Computation*, **19** (1): 45–76, DOI: [10.1162/EVCO_a_00009](https://doi.org/10.1162/EVCO_a_00009).
- BADER, J. M. (2009), Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods, PhD thesis, Swiss Federal Institute of Technology Zurich.
- BALAS, E. and ZEMEL, E. (1980), An algorithm for large zero-one knapsack problems, *Operations Research*, **28** (5): 1130–1154, DOI: [10.1287/opre.28.5.1130](https://doi.org/10.1287/opre.28.5.1130).
- BALUJA, S. (1994), Population Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, tech. rep., Carnegie-Mellon University of Pittsburgh.

- BERTSIMAS, D. and DEMIR, R. (2002), An Approximate Dynamic Programming Approach to Multidimensional Knapsack Problems, *Management Science*, **48** (4): 550–565, DOI: [10.1287/mnsc.48.4.550.208](https://doi.org/10.1287/mnsc.48.4.550.208).
- BEUME, N., NAUJOKS, B., and EMMERICH, M. (2007), SMS-EMOA: Multiobjective selection based on dominated hypervolume, *European Journal of Operational Research*, **181** (3): 1653–1669, DOI: [10.1016/j.ejor.2006.08.008](https://doi.org/10.1016/j.ejor.2006.08.008).
- BEUME, N. et al. (2009), On the Complexity of Computing the Hypervolume Indicator, *IEEE Transactions on Evolutionary Computation*, **13** (5): 1075–1082, DOI: [10.1109/TEVC.2009.2015575](https://doi.org/10.1109/TEVC.2009.2015575).
- BLICKLE, T. and THIELE, L. (1996), A Comparison of Selection Schemes Used in Evolutionary Algorithms, *Evolutionary Computation*, **4** (4): 361–394, DOI: [10.1162/evco.1996.4.4.361](https://doi.org/10.1162/evco.1996.4.4.361).
- BOOKER, L. B. (1982), Intelligent Behavior As an Adaptation to the Task Environment, AAI8214966, PhD thesis, University of Michigan.
- BOSMAN, P. and THIERENS, D. (2003), The balance between proximity and diversity in multi-objective evolutionary algorithms, *IEEE Transactions on Evolutionary Computation*, **7** (2): 174–188, DOI: [10.1109/TEVC.2003.810761](https://doi.org/10.1109/TEVC.2003.810761).
- (2011), The Roles of Local Search, Model Building and Optimal Mixing in Evolutionary Algorithms from a BBO Perspective, in: *Genetic and Evolutionary Computation Conference (GECCO'11)*, ACM, pp. 663–670, DOI: [10.1145/2001858.2002065](https://doi.org/10.1145/2001858.2002065).
- (2012a), Linkage Neighbors, Optimal Mixing and Forced Improvements in Genetic Algorithms, in: *Genetic and Evolutionary Computation Conference (GECCO'12)*, ACM, pp. 585–592, DOI: [10.1145/2330163.2330247](https://doi.org/10.1145/2330163.2330247).
- (2012b), On Measures to Build Linkage Trees in LTGA, in: *Parallel Problem Solving from Nature (PPSN'12)*, vol. 7491, LNCS, Springer, pp. 276–285, DOI: [10.1007/978-3-642-32937-1_28](https://doi.org/10.1007/978-3-642-32937-1_28).
- BOSMAN, P., THIERENS, D., LOZANO, J., et al. (2006), Multi-objective Optimization with the Naive *MIDEA*, in: *Towards a New Evolutionary Computation*, vol. 192, Studies in Fuzziness and Soft Computing, Springer, pp. 123–157, DOI: [10.1007/3-540-32494-1_6](https://doi.org/10.1007/3-540-32494-1_6).
- BOUSSIER, S. et al. (2010), A multi-level search strategy for the 0–1 Multidimensional Knapsack Problem, *Discrete Applied Mathematics*, **158** (2): 97–109, DOI: [10.1016/j.dam.2009.08.007](https://doi.org/10.1016/j.dam.2009.08.007).
- BOYER, V., ELKIHIL, M., and BAZ, D. E. (2009), Heuristics for the 0–1 multidimensional knapsack problem, *European Journal of Operational Research*, **199** (3): 658–664, DOI: [10.1016/j.ejor.2007.06.068](https://doi.org/10.1016/j.ejor.2007.06.068).
- BRINGMANN, K. and FRIEDRICH, T. (2012), Approximating the least hypervolume contributor: NP-hard in general, but fast in practice, *Theoretical Computer Science*, **425** (0): 104–116, DOI: [10.1016/j.tcs.2010.09.026](https://doi.org/10.1016/j.tcs.2010.09.026).

- (2013), Approximation quality of the hypervolume indicator, *Artificial Intelligence*, **195** (0): 265–290, DOI: [10.1016/j.artint.2012.09.005](https://doi.org/10.1016/j.artint.2012.09.005).
- CHEN, S.-C. and YU, T.-L. (2009), Difficulty of Linkage Learning in Estimation of Distribution Algorithms, in: *Genetic and Evolutionary Computation Conference (GECCO'09)*, ACM, pp. 397–404, DOI: [10.1145/1569901.1569957](https://doi.org/10.1145/1569901.1569957).
- CHIH, M. et al. (2014), Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem, *Applied Mathematical Modelling*, **38** (4): 1338–1350, DOI: [10.1016/j.apm.2013.08.009](https://doi.org/10.1016/j.apm.2013.08.009).
- CHU, P. and BEASLEY, J. (1998), A Genetic Algorithm for the Multidimensional Knapsack Problem, *Journal of Heuristics*, **4** (1): 63–86, DOI: [10.1023/A:1009642405419](https://doi.org/10.1023/A:1009642405419).
- CIOPPA, A. D., STEFANO, C. D., and MARCELLI, A. (2004), On the role of population size and niche radius in fitness sharing, *IEEE Transactions on Evolutionary Computation*, **8** (6): 580–592, DOI: [10.1109/TEVC.2004.837341](https://doi.org/10.1109/TEVC.2004.837341).
- COFFIN, D. J. and SMITH, R. E. (2007), The limitations of distribution sampling for linkage learning, in: *Conference on Evolutionary Computation (CEC'07)*, IEEE, pp. 364–369, DOI: [10.1109/CEC.2007.4424494](https://doi.org/10.1109/CEC.2007.4424494).
- CROCE, F. D. and GROSSO, A. (2012), Improved core problem based heuristics for the 0/1 multi-dimensional knapsack problem, *Computers & Operations Research*, **39** (1): 27–31, DOI: [10.1016/j.cor.2011.03.013](https://doi.org/10.1016/j.cor.2011.03.013).
- CROCOMO, M. K., MARTINS, J. P., and DELBEM, A. C. B. (2012), Decomposition of Black-box Optimization Problems by Community Detection in Bayesian Networks, *International Journal of Natural Computing Research*, **3** (4): 1–19, DOI: [10.4018/jncr.2012100101](https://doi.org/10.4018/jncr.2012100101).
- DE BONET, J., ISBELL, C., and VIOLA, P. (1997), MIMIC: Finding optima by estimating probability densities, *Advances in neural information processing systems*, 424–430.
- DE JONG, K. (1975), Analysis of the behavior of a class of genetic adaptive systems, PhD thesis, The University of Michigan.
- DE JONG, K. A. and SPEARS, W. M. (1992), A formal analysis of the role of multi-point crossover in genetic algorithms, *Annals of Mathematics and Artificial Intelligence*, **5** (1): 1–26.
- DEB, K. and JAIN, H. (2014), An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints, *IEEE Transactions on Evolutionary Computation*, **18** (4): 577–601, DOI: [10.1109/TEVC.2013.2281535](https://doi.org/10.1109/TEVC.2013.2281535).
- DEB, K. et al. (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, **6** (2): 182–197.
- DOBSON, G. (1982), Worst-Case Analysis of Greedy Heuristics for Integer Programming with Nonnegative Data, *Mathematics of Operations Research*, **7** (4): 515–531, DOI: [10.1287/moor.7.4.515](https://doi.org/10.1287/moor.7.4.515).

- DROSTE, S., JANSEN, T., and WEGENER, I. (2006), Upper and Lower Bounds for Randomized Search Heuristics in Black-Box Optimization, *Theory of Computing Systems*, **39** (4): 525–544, DOI: [10.1007/s00224-004-1177-z](https://doi.org/10.1007/s00224-004-1177-z).
- DUCHEYNE, E. I., DE BAETS, B., and DE WULF, R. (2005), Probabilistic Models for Linkage Learning in Forest Management, in: *Knowledge Incorporation in Evolutionary Computation*, vol. 167, Studies in Fuzziness and Soft Computing, Springer, pp. 177–194, DOI: [10.1007/978-3-540-44511-1_9](https://doi.org/10.1007/978-3-540-44511-1_9).
- ECHEGOYEN, C. et al. (2011), On the limits of effectiveness in estimation of distribution algorithms, in: *IEEE Congress on Evolutionary Computation (CEC'11)*, IEEE, pp. 1573–1580, DOI: [10.1109/CEC.2011.5949803](https://doi.org/10.1109/CEC.2011.5949803).
- EMMERICH, M. and FONSECA, C. (2011), Computing Hypervolume Contributions in Low Dimensions: Asymptotically Optimal Algorithm and Complexity Results, in: *Evolutionary Multi-Criterion Optimization*, vol. 6576, LNCS, Springer, pp. 121–135, DOI: [10.1007/978-3-642-19893-9_9](https://doi.org/10.1007/978-3-642-19893-9_9).
- ESHELMAN, L. (1991), On crossover as an evolutionarily viable strategy, in: *Genetic and Evolutionary Computation Conference (GECCO'91)*, ACM, pp. 61–68.
- FLESZAR, K. and HINDI, K. S. (2009), Fast, effective heuristics for the 0–1 multi-dimensional knapsack problem, *Computers & Operations Research*, **36** (5): 1602–1607, DOI: [10.1016/j.cor.2008.03.003](https://doi.org/10.1016/j.cor.2008.03.003).
- FOGEL, D. B. (1999), *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*, 2nd, Wiley-IEEE Press.
- FOGEL, L. J., OWENS, A. J., and WALSH, M. J. (1966), Artificial intelligence through simulated evolution.
- FONSECA, C. M. and FLEMING, P. J. (1995), An Overview of Evolutionary Algorithms in Multiobjective Optimization, *Evolutionary Computation*, **3** (1): 1–16, DOI: [10.1162/evco.1995.3.1.1](https://doi.org/10.1162/evco.1995.3.1.1).
- FONSECA, C. M. et al. (2011), On the Computation of the Empirical Attainment Function, in: *Evolutionary Multi-Criterion Optimization*, vol. 6576, LNCS, Springer, pp. 106–120, DOI: [10.1007/978-3-642-19893-9_8](https://doi.org/10.1007/978-3-642-19893-9_8).
- FONSECA, C., FLEMING, P., et al. (1993), Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in: *International Conference on Genetic Algorithms (ICGA'93)*, vol. 1, Morgan Kaufmann, pp. 416–423.
- FOX, G. E. and SCUDDER, G. D. (1985), A heuristic with tie breaking for certain 0–1 integer programming models, *Naval Research Logistics Quarterly*, **32** (4): 613–623, DOI: [10.1002/nav.3800320408](https://doi.org/10.1002/nav.3800320408).
- FREVILLE, A. (2004), The multidimensional 0–1 knapsack problem: An overview, *European Journal of Operational Research*, **155** (1): 1–21, DOI: [10.1016/S0377-2217\(03\)00274-1](https://doi.org/10.1016/S0377-2217(03)00274-1).

- FREVILLE, A. and PLATEAU, G. (1994), An efficient preprocessing procedure for the multi-dimensional 0–1 knapsack problem, *Discrete Applied Mathematics*, **49** (1): 189–212, DOI: [10.1016/0166-218X\(94\)90209-7](https://doi.org/10.1016/0166-218X(94)90209-7).
- GAVISH, B. and PIRKUL, H. (1985), Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality, *Mathematical Programming*, **31** (1): 78–105, DOI: [10.1007/BF02591863](https://doi.org/10.1007/BF02591863).
- GIAGKIOZIS, I., PURSHOUSE, R. C., and FLEMING, P. J. (2013), Generalized Decomposition, in: *Evolutionary Multi-Criterion Optimization (EMO'13)*, vol. 7811, LNCS, Springer, pp. 428–442, DOI: [10.1007/978-3-642-37140-0_33](https://doi.org/10.1007/978-3-642-37140-0_33).
- GIAGKIOZIS, I., PURSHOUSE, R. C., and FLEMING, P. J. (2014), Generalized decomposition and cross entropy methods for many-objective optimization, *Information Sciences*, **282** (0): 363–387, DOI: [10.1016/j.ins.2014.05.045](https://doi.org/10.1016/j.ins.2014.05.045).
- GILMORE, P. C. and GOMORY, R. E. (1966), The Theory and Computation of Knapsack Functions, *Operations Research*, **14** (6): 1045–1074, DOI: [10.1287/opre.14.6.1045](https://doi.org/10.1287/opre.14.6.1045).
- GLOVER, F. and KOCHENBERGER, G. (1996), Critical Event Tabu Search for Multidimensional Knapsack Problems, in: *Meta-Heuristics*, Springer, pp. 407–427, DOI: [10.1007/978-1-4613-1361-8_25](https://doi.org/10.1007/978-1-4613-1361-8_25).
- GOLDBERG, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st, Addison-Wesley Professional, pp. 432.
- (2002), The design of innovation, vol. 7, *Genetic Algorithms and Evolutionary Computation*, Springer, pp. 248.
- GOLDBERG, D. E. and RICHARDSON, J. (1987), Genetic algorithms with sharing for multimodal function optimization, in: *International Conference on Genetic Algorithms (ICGA'87)*, L. Erlbaum Associates Inc., pp. 41–49.
- GOLDBERG, D. E., SASTRY, K., and LLORÀ, X. (2007), Toward routine billion-variable optimization using genetic algorithms, *Complexity*, **12** (3): 27–29, DOI: [10.1002/cplx.20168](https://doi.org/10.1002/cplx.20168).
- GONZÁLEZ, C., LOZANO, J., and LARRAÑAGA, P. (2002), Mathematical modelling of UMDac algorithm with tournament selection. Behaviour on linear and quadratic functions, *International Journal of Approximate Reasoning*, **31** (3): 313–340, DOI: [10.1016/S0888-613X\(02\)00092-0](https://doi.org/10.1016/S0888-613X(02)00092-0).
- GOTTLIEB, J. (2000a), On the Effectivity of Evolutionary Algorithms for the Multidimensional Knapsack Problem, in: *Artificial Evolution (AE'00)*, vol. 1829, LNCS, Springer Berlin Heidelberg, pp. 23–37, DOI: [10.1007/10721187_2](https://doi.org/10.1007/10721187_2).
- (2000b), Permutation-based Evolutionary Algorithms for Multidimensional Knapsack Problems, in: *Symposium on Applied Computing (SAC'00)*, ACM, pp. 408–414, DOI: [10.1145/335603.335866](https://doi.org/10.1145/335603.335866).
- (2001), On the Feasibility Problem of Penalty-Based Evolutionary Algorithms for Knapsack Problems, in: *Applications of Evolutionary Computing*, vol. 2037, LNCS, Springer, pp. 50–59, DOI: [10.1007/3-540-45365-2_6](https://doi.org/10.1007/3-540-45365-2_6).

- GRONAU, I. and MORAN, S. (2007), Optimal implementations of UPGMA and other common clustering algorithms, *Information Processing Letters*, **104** (6): 205–210, DOI: [10.1016/j.ipl.2007.07.002](https://doi.org/10.1016/j.ipl.2007.07.002).
- HA, B. et al. (2014), Application of Modified Bayesian Optimization Algorithm to the design of reflectarray antenna, in: *International Conference on Numerical Electromagnetic Modeling and Optimization for RF, Microwave, and Terahertz Applications (NEMO), 2014*, pp. 1–4, DOI: [10.1109/NEMO.2014.6995720](https://doi.org/10.1109/NEMO.2014.6995720).
- HAMMER, P. L. and PELED, U. N. (1972), On the Maximization of a Pseudo-Boolean Function, *Journal of the ACM*, **19** (2): 265–282, DOI: [10.1145/321694.321700](https://doi.org/10.1145/321694.321700).
- HANAFLI, S. and WILBAUT, C. (2011), Improved convergent heuristics for the 0-1 multidimensional knapsack problem, *Annals of Operations Research*, **183** (1): 125–142, DOI: [10.1007/s10479-009-0546-z](https://doi.org/10.1007/s10479-009-0546-z).
- HARIK, G. R. (1995), Finding Multimodal Solutions Using Restricted Tournament Selection, in: *International Conference on Genetic Algorithms (ICGA'95)*, Morgan Kaufmann, pp. 24–31.
- HARIK, G. R. and GOLDBERG, D. E. (2000), Linkage learning through probabilistic expression, *Computer Methods in Applied Mechanics and Engineering*, **186** (2–4): 295–310, DOI: [10.1016/S0045-7825\(99\)00388-6](https://doi.org/10.1016/S0045-7825(99)00388-6).
- HARIK, G. R., LOBO, F. G., and GOLDBERG, D. E. (1999), The compact genetic algorithm, *IEEE Transactions on Evolutionary Computation*, **3** (4): 287–297, DOI: [10.1109/4235.797971](https://doi.org/10.1109/4235.797971).
- HARIK, G. R., LOBO, F. G., and SASTRY, K. (2006), Linkage Learning via Probabilistic Modeling in the Extended Compact Genetic Algorithm (ECGA), in: *Scalable Optimization via Probabilistic Modeling*, Studies in Computational Intelligence, Springer, pp. 39–61, DOI: [10.1007/978-3-540-34954-9_3](https://doi.org/10.1007/978-3-540-34954-9_3).
- HARIK, G. (1997), Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms, PhD thesis, The University of Michigan.
- HAUSCHILD, M. et al. (2009), Analyzing Probabilistic Models in Hierarchical BOA, *IEEE Transactions on Evolutionary Computation*, **13** (6): 1199–1217, DOI: [10.1109/TEVC.2008.2004423](https://doi.org/10.1109/TEVC.2008.2004423).
- HAUSCHILD, M. et al. (2007), Analyzing probabilistic models in hierarchical BOA on traps and spin glasses, in: *Genetic and Evolutionary Computation Conference (GECCO'07)*, ACM, pp. 523–530, DOI: [10.1145/1276958.1277070](https://doi.org/10.1145/1276958.1277070).
- HAUSCHILD, M. W. and PELIKAN, M. (2010), Network Crossover Performance on NK Landscapes and Deceptive Problems, in: *Genetic and Evolutionary Computation Conference (GECCO'10)*, ACM, pp. 713–720, DOI: [10.1145/1830483.1830612](https://doi.org/10.1145/1830483.1830612).
- HAUSCHILD, M. W. et al. (2008), Using previous models to bias structural learning in the hierarchical BOA, in: *Genetic and Evolutionary Computation Conference (GECCO'08)*, ACM, pp. 415–422, DOI: [10.1145/1389095.1389172](https://doi.org/10.1145/1389095.1389172).

- HILL, R. R., CHO, Y. K., and MOORE, J. T. (2012), Problem reduction heuristic for the 0–1 multidimensional knapsack problem, *Computers & Operations Research*, **39** (1): 19–26, DOI: [10.1016/j.cor.2010.06.009](https://doi.org/10.1016/j.cor.2010.06.009).
- HINGEE, K. and HUTTER, M. (2008), Equivalence of probabilistic tournament and polynomial ranking selection, in: *Congress on Evolutionary Computation (CEC'08)*, CEC '08, IEEE, pp. 564–571, DOI: [10.1109/CEC.2008.4630852](https://doi.org/10.1109/CEC.2008.4630852).
- HOLLAND, J. H. (1975), *Adaptation in Natural and Artificial Systems, An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, Complex Adaptive Systems*, A Bradford Book, pp. 183.
- HONG, Y., REN, Q., and ZENG, J. (2005), Adaptive population size for univariate marginal distribution algorithm, in: *Congress on Evolutionary Computation (CEC'05)*, vol. 2, IEEE, pp. 1396–1402 Vol. 2, DOI: [10.1109/CEC.2005.1554853](https://doi.org/10.1109/CEC.2005.1554853).
- HONG, Y. et al. (2008), Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm, *Pattern Recognition*, **41** (9): 2742–2756, DOI: [10.1016/j.patcog.2008.03.007](https://doi.org/10.1016/j.patcog.2008.03.007).
- HUBAND, S. et al. (2003), An evolution strategy with probabilistic mutation for multi-objective optimisation, in: *Congress on Evolutionary Computation (CEC'03)*, vol. 4, IEEE, pp. 2284–2291 Vol.4, DOI: [10.1109/CEC.2003.1299373](https://doi.org/10.1109/CEC.2003.1299373).
- IGEL CHRISTIAN, HANSEN NIKOLAUS, and ROTH STEFAN (2007), Covariance Matrix Adaptation for Multi-objective Optimization, *Evolutionary Computation*, **15** (1): 1–28, DOI: [10.1162/evco.2007.15.1.1](https://doi.org/10.1162/evco.2007.15.1.1).
- ISHIBUCHI, H. et al. (2009), Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations, in: *International Conference on Systems, Man and Cybernetics (SMC'09)*, IEEE, pp. 1758–1763, DOI: [10.1109/ICSMC.2009.5346628](https://doi.org/10.1109/ICSMC.2009.5346628).
- ISHIBUCHI, H., TSUKAMOTO, N., and NOJIMA, Y. (2010), Diversity Improvement by Non-Geometric Binary Crossover in Evolutionary Multiobjective Optimization, *IEEE Transactions on Evolutionary Computation*, **14** (6): 985–998, DOI: [10.1109/TEVC.2010.2043365](https://doi.org/10.1109/TEVC.2010.2043365).
- ISHIBUCHI, H. and NARUKAWA, K. (2005), Recombination of Similar Parents in EMO Algorithms, in: *Evolutionary Multi-Criterion Optimization (EMO'05)*, vol. 3410, LNCS, Springer, pp. 265–279, DOI: [10.1007/978-3-540-31880-4_19](https://doi.org/10.1007/978-3-540-31880-4_19).
- ISHIBUCHI, H. and SHIBATA, Y. (2003a), A Similarity-Based Mating Scheme for Evolutionary Multiobjective Optimization, in: *Genetic and Evolutionary Computation Conference (GECCO'03)*, vol. 2723, Springer, pp. 1065–1076, DOI: [10.1007/3-540-45105-6_116](https://doi.org/10.1007/3-540-45105-6_116).
- (2003b), An Empirical Study on the Effect of Mating Restriction on the Search Ability of EMO Algorithms, in: *Evolutionary Multi-Criterion Optimization (EMO'03)*, vol. 2632, LNCS, Springer, pp. 433–447, DOI: [10.1007/3-540-36970-8_31](https://doi.org/10.1007/3-540-36970-8_31).

- ISHIBUCHI, H. et al. (2007), Effects of the Use of Non-geometric Binary Crossover on Evolutionary Multiobjective Optimization, in: *Genetic and Evolutionary Computation Conference (GECCO'07)*, ACM, pp. 829–836, DOI: [10.1145/1276958.1277122](https://doi.org/10.1145/1276958.1277122).
- ISHIBUCHI, H. et al. (2008a), An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization, *European Journal of Operational Research*, **188** (1): 57–75, DOI: [10.1016/j.ejor.2007.04.007](https://doi.org/10.1016/j.ejor.2007.04.007).
- ISHIBUCHI, H., TSUKAMOTO, N., and NOJIMA, Y. (2008b), Evolutionary many-objective optimization: A short review, in: *Congress on Evolutionary Computation (CEC'08)*, IEEE, pp. 2419–2426, DOI: [10.1109/CEC.2008.4631121](https://doi.org/10.1109/CEC.2008.4631121).
- ISHIBUCHI, H., AKEDO, N., and NOJIMA, Y. (2012), Recombination of Similar Parents in SMS-EMOA on Many-Objective 0/1 Knapsack Problems, in: *Parallel Problem Solving from Nature (PPSN'12)*, vol. 7492, LNCS, Springer, pp. 132–142, DOI: [10.1007/978-3-642-32964-7_14](https://doi.org/10.1007/978-3-642-32964-7_14).
- (2013), Relation between Neighborhood Size and MOEA/D Performance on Many-Objective Problems, in: *Evolutionary Multi-Criterion Optimization (EMO'13)*, vol. 7811, LNCS, Springer, pp. 459–474, DOI: [10.1007/978-3-642-37140-0_35](https://doi.org/10.1007/978-3-642-37140-0_35).
- ISHIBUCHI, H. et al. (2014), Distance-Based Analysis of Crossover Operators for Many-Objective Knapsack Problems, in: *Parallel Problem Solving from Nature (PPSN'14)*, vol. 8672, LNCS, Springer, pp. 600–610, DOI: [10.1007/978-3-319-10762-2_59](https://doi.org/10.1007/978-3-319-10762-2_59).
- JAIN, H. and DEB, K. (2014), An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach, *IEEE Transactions on Evolutionary Computation*, **18** (4): 602–622, DOI: [10.1109/TEVC.2013.2281534](https://doi.org/10.1109/TEVC.2013.2281534).
- JANSEN, T. and WEGENER, I. (2005), Real royal road functions—where crossover provably is essential, *Discrete Applied Mathematics*, **149** (1–3): 111–125, DOI: [10.1016/j.dam.2004.02.019](https://doi.org/10.1016/j.dam.2004.02.019).
- JASZKIEWICZ, A. (2002a), Genetic local search for multi-objective combinatorial optimization, *European Journal of Operational Research*, **137** (1): 50–71, DOI: [10.1016/S0377-2217\(01\)00104-7](https://doi.org/10.1016/S0377-2217(01)00104-7).
- (2002b), On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment, *IEEE Transactions on Evolutionary Computation*, **6** (4): 402–412, DOI: [10.1109/TEVC.2002.802873](https://doi.org/10.1109/TEVC.2002.802873).
- JENSEN, J. L. W. V. (1906), Sur les fonctions convexes et les inégalités entre les valeurs moyennes, *Acta Mathematica*, **30** (1): 175–193, DOI: [10.1007/BF02418571](https://doi.org/10.1007/BF02418571).
- KAN, A. R., STOUGIE, L., and VERCELLIS, C. (1993), A class of generalized greedy algorithms for the multi-knapsack problem, *Discrete Applied Mathematics*, **42** (2–3): 279–290, DOI: [10.1016/0166-218X\(93\)90051-0](https://doi.org/10.1016/0166-218X(93)90051-0).
- KAUFFMAN, S. A. (1993), *The origins of order*, Oxford University Press New York.

- KELBERT, M. and SUHOV, Y. M. (2013), Information theory and coding by example, 1st, Cambridge University Press, pp. 526.
- KELLERER, H., PFERSCHY, U., and PISINGER, D. (2004), Knapsack problems, Springer Verlag.
- KHAN, N., GOLDBERG, D. E., and PELIKAN, M. (2002), Multi-Objective Bayesian Optimization Algorithm, tech. rep., University of Illinois.
- KNOWLES, J. and CORNE, D. (2003), Properties of an adaptive archiving algorithm for storing nondominated vectors, *IEEE Transactions on Evolutionary Computation*, **7** (2): 100–116, DOI: [10.1109/TEVC.2003.810755](https://doi.org/10.1109/TEVC.2003.810755).
- KNOWLES, J. D. (2002), Local-search and hybrid evolutionary algorithms for Pareto optimization. PhD thesis, University of Reading.
- KONG, M., TIAN, P., and KAO, Y. (2008), A new ant colony optimization algorithm for the multidimensional Knapsack problem, *Computers & Operations Research*, **35** (8): 2672–2683, DOI: [10.1016/j.cor.2006.12.029](https://doi.org/10.1016/j.cor.2006.12.029).
- KONSTANTINIDIS, A. and YANG, K. (2011), Multi-objective energy-efficient dense deployment in Wireless Sensor Networks using a hybrid problem-specific MOEA/D, *Applied Soft Computing*, **11** (6): 4117–4134, DOI: [10.1016/j.asoc.2011.02.031](https://doi.org/10.1016/j.asoc.2011.02.031).
- KOZA, J. R. (1992), Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press.
- KRAMER, G. and GALLAGHER, J. (2003), Improvements to the *CGA enabling online intrinsic evolution in compact EH devices, in: *Proceedings of NASA/DoD Conference on Evolvable Hardware*, pp. 225–231, DOI: [10.1109/EH.2003.1217670](https://doi.org/10.1109/EH.2003.1217670).
- KRAMER, G., GALLAGHER, J., and RAYMER, M. (2004), On the relative efficacies of *cGA variants for intrinsic evolvable hardware; population, mutation, and random immigrants, in: *Proceedings of NASA/DoD Conference on Evolvable Hardware*, pp. 225–230, DOI: [10.1109/EH.2004.1310834](https://doi.org/10.1109/EH.2004.1310834).
- KULIK, A. and SHACHNAI, H. (2010), There is no EPTAS for two-dimensional knapsack, *Information Processing Letters*, **110** (16): 707–710, DOI: [10.1016/j.ipl.2010.05.031](https://doi.org/10.1016/j.ipl.2010.05.031).
- KÖTZING, T., SUDHOLT, D., and THEILE, M. (2011), How Crossover Helps in Pseudo-boolean Optimization, in: *Conference on Genetic and Evolutionary Computation (GECCO'11)*, ACM, pp. 989–996, DOI: [10.1145/2001576.2001711](https://doi.org/10.1145/2001576.2001711).
- LARRAANAGA, P. and LOZANO, J. A. (2001), Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers.
- LARRAÑAGA, P. et al. (2012), A review on probabilistic graphical models in evolutionary computation, *Journal of Heuristics*, **18** (5): 795–819, DOI: [10.1007/s10732-012-9208-4](https://doi.org/10.1007/s10732-012-9208-4).
- LAUMANN, M. and OCENASEK, J. (2002), Bayesian Optimization Algorithms for Multi-objective Optimization, in: *Parallel Problem Solving from Nature (PPSN'02)*, vol. 2439, LNCS, Springer, pp. 298–307, DOI: [10.1007/3-540-45712-7_29](https://doi.org/10.1007/3-540-45712-7_29).

- LEHRE, P. and WITT, C. (2012), Black-Box Search by Unbiased Variation, *Algorithmica*, **64** (4): 623–642, DOI: [10.1007/s00453-012-9616-8](https://doi.org/10.1007/s00453-012-9616-8).
- LEHRE, P. and YAO, X. (2012), On the Impact of Mutation-Selection Balance on the Runtime of Evolutionary Algorithms, *IEEE Transactions on Evolutionary Computation*, **16** (2): 225–241, DOI: [10.1109/TEVC.2011.2112665](https://doi.org/10.1109/TEVC.2011.2112665).
- LI, H. and ZHANG, Q. (2009), Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II, *IEEE Transactions on Evolutionary Computation*, **13** (2): 284–302, DOI: [10.1109/TEVC.2008.925798](https://doi.org/10.1109/TEVC.2008.925798).
- LI, J. and SONG, Y. (2013), Community detection in complex networks using extended compact genetic algorithm, *Soft Computing*, **17** (6): 925–937, DOI: [10.1007/s00500-012-0942-1](https://doi.org/10.1007/s00500-012-0942-1).
- LI, J. et al. (2014), Team of Bayesian optimization algorithms to solve task assignment problems in heterogeneous computing systems, in: *International Conference on Systems, Man and Cybernetics (SMC'14)*, IEEE, pp. 127–132, DOI: [10.1109/SMC.2014.6973895](https://doi.org/10.1109/SMC.2014.6973895).
- LI, J. and AICKELIN, U. (2006), BOA for Nurse Scheduling, in: *Scalable Optimization via Probabilistic Modeling*, vol. 33, Studies in Computational Intelligence, Springer, pp. 315–332, DOI: [10.1007/978-3-540-34954-9_14](https://doi.org/10.1007/978-3-540-34954-9_14).
- LI, V. and CURRY, G. (2005), Solving multidimensional knapsack problems with generalized upper bound constraints using critical event tabu search, *Computers & Operations Research*, **32** (4): 825–848, DOI: [10.1016/j.cor.2003.08.021](https://doi.org/10.1016/j.cor.2003.08.021).
- LI, V. C. (2005), Tight oscillations tabu search for multidimensional knapsack problems with generalized upper bound constraints, *Computers & Operations Research*, **32** (11): 2843–2852, DOI: [10.1016/j.cor.2004.04.020](https://doi.org/10.1016/j.cor.2004.04.020).
- LI, V. C., LIANG, Y.-C., and CHANG, H.-F. (2012), Solving the multidimensional knapsack problems with generalized upper bound constraints by the adaptive memory projection method, *Computers & Operations Research*, **39** (9): 2111–2121, DOI: [10.1016/j.cor.2011.10.016](https://doi.org/10.1016/j.cor.2011.10.016).
- LIMA, C. et al. (2007), Influence of selection and replacement strategies on linkage learning in BOA, in: *Congress on Evolutionary Computation (CEC'07)*, IEEE, pp. 1083–1090, DOI: [10.1109/CEC.2007.4424590](https://doi.org/10.1109/CEC.2007.4424590).
- LIMA, C. et al., Model accuracy in the Bayesian optimization algorithm, *Soft Computing*, **15** (7): 1351–1371, DOI: [10.1007/s00500-010-0675-y](https://doi.org/10.1007/s00500-010-0675-y).
- LIU, H.-L., GU, F., and ZHANG, Q. (2014), Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems, *IEEE Transactions on Evolutionary Computation*, **18** (3): 450–455, DOI: [10.1109/TEVC.2013.2281533](https://doi.org/10.1109/TEVC.2013.2281533).
- LOZANO, J. A. (2006), Towards a new evolutionary computation: Advances on estimation of distribution algorithms, vol. 192, Springer.

- MA, X. et al. (2014), MOEA/D with Baldwinian learning inspired by the regularity property of continuous multiobjective problem, *Neurocomputing*, **145** (0): 336–352, DOI: [10.1016/j.neucom.2014.05.025](https://doi.org/10.1016/j.neucom.2014.05.025).
- MAHFOUD, S. W. (1995), Niching methods for genetic algorithms, PhD thesis, University of Illinois at Urbana-Champaign.
- MARSTEN, R. E. and MORIN, T. L. (1977), Optimal solutions found for Senju and Toyoda 0/1 integer programming problems, *Management Science*, **23** (12): 1364–1365, DOI: [10.1287/mnsc.23.12.1364](https://doi.org/10.1287/mnsc.23.12.1364).
- MARSTEN, R. and MORIN, T. (1978), A hybrid approach to discrete mathematical programming, *Mathematical Programming*, **14** (1): 21–40, DOI: [10.1007/BF01588949](https://doi.org/10.1007/BF01588949).
- MARTELLO, S., PISINGER, D., and TOTH, P. (2000), New trends in exact algorithms for the 0–1 knapsack problem, *European Journal of Operational Research*, **123** (2): 325–332, DOI: [10.1016/S0377-2217\(99\)00260-X](https://doi.org/10.1016/S0377-2217(99)00260-X).
- MARTINS, J. P. and DELBEM, A. C. B. (2013), The influence of linkage-learning in the linkage-tree GA when solving multidimensional knapsack problems, in: *Genetic and Evolutionary Computation Conference (GECCO'13)*, ACM, pp. 821–828, DOI: [10.1145/2463372.2463476](https://doi.org/10.1145/2463372.2463476).
- (2014), Multimodality and the Linkage-learning Difficulty of Additively Separable Functions, in: *Genetic and Evolutionary Computation Conference (GECCO'14)*, ACM, pp. 365–372, DOI: [10.1145/2576768.2598281](https://doi.org/10.1145/2576768.2598281).
- (2015a), Assessment of Linkage learning Limits in Highly Multimodal NK-landscapes, *Swarm and Evolutionary Computation (IN REVIEW)*,
- (2015b), Handling Crossover Bias to Improve Diversity in Multiobjective Evolutionary Optimization, in: *Genetic and Evolutionary Computation Conference (GECCO'15)*, (IN REVIEW), ACM.
- (2015c), Linkage learning and Crossover bias in Bi-objective pMNK-landscapes, *Swarm and Evolutionary Computation*, (TO SUBMIT).
- MARTINS, J. P. et al. (2011), Multi-objective Phylogenetic Algorithm: Solving Multi-objective Decomposable Deceptive Problems, in: *Evolutionary Multi-Criterion Optimization (EMO'11)*, vol. 6576, LNCS, Springer, pp. 285–297, DOI: [10.1007/978-3-642-19893-9_20](https://doi.org/10.1007/978-3-642-19893-9_20).
- MARTINS, J. P. et al. (2013), A Comparison of Linkage-learning-based Genetic Algorithms in Multidimensional knapsack Problems, in: *Conference on Evolutionary Computation (CEC'13)*, IEEE, pp. 502–509, DOI: [10.1109/CEC.2013.6557610](https://doi.org/10.1109/CEC.2013.6557610).
- MARTINS, J. P., FONSECA, C. M., and DELBEM, A. C. B. (2014), On the performance of linkage-tree genetic algorithms for the multidimensional knapsack problem, *Neurocomputing*, **146** (0): 17–29, DOI: [10.1016/j.neucom.2014.04.069](https://doi.org/10.1016/j.neucom.2014.04.069).
- MARTÍ, L. et al. (2008a), Introducing MONEDA: Scalable Multiobjective Optimization with a Neural Estimation of Distribution Algorithm, in: *Conference on Genetic and Evolutionary Computation (GECCO'08)*, ACM, pp. 689–696.

- MARTÍ, L. et al. (2008b), Model-building algorithms for multiobjective EDAs: Directions for improvement, in: *Congress on Evolutionary Computation (CEC'08)*, IEEE, pp. 2843–2850, DOI: [10.1109/CEC.2008.4631179](https://doi.org/10.1109/CEC.2008.4631179).
- MARTÍ, L. et al. (2009), On the Model–Building Issue of Multi–Objective Estimation of Distribution Algorithms, in: *Hybrid Artificial Intelligence Systems*, vol. 5572, LNCS, Springer, pp. 293–300, DOI: [10.1007/978-3-642-02319-4_35](https://doi.org/10.1007/978-3-642-02319-4_35).
- MARTÍ, L. et al. (2011a), MB-GNG: Addressing drawbacks in multi-objective optimization estimation of distribution algorithms, *Operations Research Letters*, **39** (2): 150–154, DOI: [10.1016/j.orl.2011.01.002](https://doi.org/10.1016/j.orl.2011.01.002).
- MARTÍ, L. et al. (2011b), Scalable multi-objective optimization, PhD thesis, Universidad Carlos III, Departamento de Informática.
- MIETTINEN, K. (1999), Nonlinear multiobjective optimization, International series in operations research & management science, Kluwer Academic Publishers.
- MIETTINEN, K. and MÄKELÄ, M. M. (2002), On scalarizing functions in multiobjective optimization, *OR Spectrum*, **24** (2): 193–213, DOI: [10.1007/s00291-001-0092-9](https://doi.org/10.1007/s00291-001-0092-9).
- MILLER, B. and SHAW, M. (1996), Genetic algorithms with dynamic niche sharing for multimodal function optimization, in: *Conference on Evolutionary Computation (CEC'96)*, IEEE, pp. 786–791, DOI: [10.1109/ICEC.1996.542701](https://doi.org/10.1109/ICEC.1996.542701).
- MILLER, B. L. and GOLDBERG, D. E. (1996), Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise, *Evolutionary Computation*, **4** (2): 113–131, DOI: [10.1162/evco.1996.4.2.113](https://doi.org/10.1162/evco.1996.4.2.113).
- MITCHELL, M. (1998), An introduction to genetic algorithms, Complex Adaptive Systems, A Bradford Book, pp. 221.
- MITCHELL, M. and HOLLAND, J. H. (1993), When Will a Genetic Algorithm Outperform Hill Climbing?, in: *International Conference on Genetic Algorithms (ICGA'93)*, Morgan Kaufmann, pp. 647–656.
- MORAGLIO, A. and POLI, R. (2004), Topological Interpretation of Crossover, in: *Genetic and Evolutionary Computation Conference (GECCO'04)*, vol. 3102, Springer, pp. 1377–1388, DOI: [10.1007/978-3-540-24854-5_131](https://doi.org/10.1007/978-3-540-24854-5_131).
- MOSCATO, P. (1989), On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, *Caltech concurrent computation program, C3P Report*, **826**: 1989.
- MÜHLENBEIN, H. and PAASS, G. (1996), From recombination of genes to the estimation of distributions I. Binary parameters, in: *Parallel Problem Solving from Nature (PPSN'96)*, LNCS, Springer, pp. 178–187, DOI: [10.1007/3-540-61723-X_982](https://doi.org/10.1007/3-540-61723-X_982).
- MÜHLENBEIN, H. and SCHLIERKAMP-VOOSEN, D. (1993), Predictive models for the breeder genetic algorithm i. continuous parameter optimization, *Evolutionary Computation*, **1** (1): 25–49.

- MÜHLENBEIN, H. (1997), The Equation for Response to Selection and Its Use for Prediction, *Evolutionary Computation*, **5** (3): 303–346.
- OCENASEK, J. et al. (2004), A Mixed Bayesian Optimization Algorithm with Variance Adaptation, in: *Parallel Problem Solving from Nature (PPSN'04)*, vol. 3242, LNCS, Springer, pp. 352–361, DOI: [10.1007/978-3-540-30217-9_36](https://doi.org/10.1007/978-3-540-30217-9_36).
- PAIXÃO, T. et al. (2014), Unified Model of Evolution, tech. rep., pp. 1–26.
- PELIKAN, M., SASTRY, K., and GOLDBERG, D. (2005), Multiobjective hBOA, clustering, and scalability, in: *Genetic and Evolutionary Computation Conference*, GECCO '05, ACM.
- PELIKAN, M., SASTRY, K., and CANTU-PAZ, E. (2006a), Scalable optimization via probabilistic modeling: From algorithms to applications, Springer Verlag.
- PELIKAN, M. (2005), Hierarchical Bayesian Optimization Algorithm, in: *Hierarchical Bayesian Optimization Algorithm*, vol. 170, Studies in Fuzziness and Soft Computing, [10.1007/978-3-540-32373-0_6](https://doi.org/10.1007/978-3-540-32373-0_6), Springer Berlin / Heidelberg, pp. 105–129, DOI: [10.1007/978-3-540-32373-0_6](https://doi.org/10.1007/978-3-540-32373-0_6).
- (2008), Analysis of Estimation of Distribution Algorithms and Genetic Algorithms on NK Landscapes, in: *Genetic and Evolutionary Computation Conference*, GECCO '08, ACM, pp. 1033–1040, DOI: [10.1145/1389095.1389287](https://doi.org/10.1145/1389095.1389287).
- (2010), NK landscapes, problem difficulty, and hybrid evolutionary algorithms, in: *Genetic and Evolutionary Computation Conference*, GECCO '10, ACM, pp. 665–672, DOI: [10.1145/1830483.1830606](https://doi.org/10.1145/1830483.1830606).
- (2011), Analysis of Epistasis Correlation on NK Landscapes with Nearest-neighbor Interactions, in: *Genetic and Evolutionary Computation Conference*, GECCO '11, ACM, pp. 1013–1020, DOI: [10.1145/2001576.2001714](https://doi.org/10.1145/2001576.2001714).
- PELIKAN, M. and GOLDBERG, D. E. (2003a), A hierarchy machine: Learning to optimize from nature and humans, *Complexity*, **8** (5): 36–45.
- (2003b), Hierarchical BOA solves Ising spin glasses and MAXSAT, in: *Genetic and Evolutionary Computation Conference*, vol. 2724, GECCO '03, Springer, pp. 1271–1282.
- PELIKAN, M. and HARTMANN, A. (2006), Searching for Ground States of Ising Spin Glasses with Hierarchical BOA and Cluster Exact Approximation, in: *Scalable Optimization via Probabilistic Modeling*, vol. 33, Studies in Computational Intelligence, Springer Berlin Heidelberg, pp. 333–349, DOI: [10.1007/978-3-540-34954-9_15](https://doi.org/10.1007/978-3-540-34954-9_15).
- PELIKAN, M. and MUEHLENBEIN, H. (1999), The Bivariate Marginal Distribution Algorithm, in: *Advances in Soft Computing*, Springer Verlag, pp. 521.
- PELIKAN, M., GOLDBERG, D. E., and CANTÚ-PAZ, E. (1999), BOA: The Bayesian Optimization Algorithm, in: *Genetic and Evolutionary Computation Conference*, vol. I, GECCO '99, Morgan Kaufmann Publishers, San Francisco, CA, pp. 525–532.
- PELIKAN, M., SASTRY, K., and GOLDBERG, D. (2006b), Multiobjective Estimation of Distribution Algorithms, vol. 33, Studies in Computational Intelligence, [10.1007/978-3-540-34954-](https://doi.org/10.1007/978-3-540-34954-9)

- 9_10, Springer Berlin / Heidelberg, pp. 223–248, DOI: [10.1007/978-3-540-34954-9_10](https://doi.org/10.1007/978-3-540-34954-9_10).
- PELIKAN, M., HELMUT, K. G., and KOBE, S. (2008), Finding Ground States of Sherrington-Kirkpatrick Spin Glasses with Hierarchical Boa and Genetic Algorithms, in: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO '08*, ACM, pp. 447–454, DOI: [10.1145/1389095.1389176](https://doi.org/10.1145/1389095.1389176).
- PELIKAN, M., HAUSCHILD, M. W., and THIERENS, D. (2011), Pairwise and problem-specific distance metrics in the linkage tree genetic algorithm, in: *Genetic and Evolutionary Computation Conference, GECCO '11*, ACM, pp. 1005–1012, DOI: [10.1145/2001576.2001713](https://doi.org/10.1145/2001576.2001713).
- PETROWSKI, A. (1996), A clearing procedure as a niching method for genetic algorithms, in: *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 798–803, DOI: [10.1109/ICEC.1996.542703](https://doi.org/10.1109/ICEC.1996.542703).
- PUCHINGER, J., RAIDL, G. R., and PFERSCHY, U. (2010), The multidimensional knapsack problem: Structure and algorithms, *INFORMS Journal on Computing*, **22** (2): 250–265.
- RAIDL, G. (1998), An improved genetic algorithm for the multiconstrained 0-1 knapsack problem, in: *IEEE Congress on Evolutionary Computation, CEC '98*, pp. 207–211, DOI: [10.1109/ICEC.1998.699502](https://doi.org/10.1109/ICEC.1998.699502).
- RECHENBERG, I. (1965), Cybernetic solution path of an experimental problem.
- REEVES, C. R. and ROWE, J. E. (2002), *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*, Kluwer Academic Publishers.
- REZA, R. and ARASH, H. (2006), A Step Forward in Studying the Compact Genetic Algorithm, *Evolutionary Computation*, **14** (3): 277–289, DOI: [10.1162/evco.2006.14.3.277](https://doi.org/10.1162/evco.2006.14.3.277).
- ROTHLAUF, F. (2006), *Representations for Genetic and Evolutionary Algorithms*, 2nd, Springer Berlin Heidelberg, pp. 325.
- SANTARELLI, S. et al. (2006), Military antenna design using simple and competent genetic algorithms, *Mathematical and Computer Modelling*, **43** (9–10): 990–1022, DOI: [10.1016/j.mcm.2005.05.024](https://doi.org/10.1016/j.mcm.2005.05.024).
- SARENI, B. and KRAHENBUHL, L. (1998), Fitness sharing and niching methods revisited, *IEEE Transactions on Evolutionary Computation*, **2** (3): 97–106, DOI: [10.1109/4235.735432](https://doi.org/10.1109/4235.735432).
- SASTRY, K., GOLDBERG, D., and PELIKAN, M. (2005), Limits of scalability of multiobjective estimation of distribution algorithms, in: *IEEE Congress on Evolutionary Computation*, vol. 3, CEC '05, pp. 2217–2224, DOI: [10.1109/CEC.2005.1554970](https://doi.org/10.1109/CEC.2005.1554970).
- SASTRY, K. and GOLDBERG, D. (2004), Let's Get Ready to Rumble: Crossover Versus Mutation Head to Head, in: *Genetic and Evolutionary Computation Conference*, vol. 3103, GECCO '04, [10.1007/978-3-540-24855-2_12](https://doi.org/10.1007/978-3-540-24855-2_12), Springer Berlin / Heidelberg, pp. 126–137.
- SASTRY, K., GOLDBERG, D. E., and JOHNSON, D. D. (2007), Scalability of a Hybrid Extended Compact Genetic Algorithm for Ground State Optimization of Clusters, *Materials and Manufacturing Processes*, **22** (5): 570–576, DOI: [10.1080/10426910701319654](https://doi.org/10.1080/10426910701319654).

- SATO, H., AGUIRRE, H., and TANAKA, K. (2013), Variable space diversity, crossover and mutation in MOEA solving many-objective knapsack problems, *Annals of Mathematics and Artificial Intelligence*, **68** (4): 197–224, DOI: [10.1007/s10472-012-9293-y](https://doi.org/10.1007/s10472-012-9293-y).
- SEBAG, M. and DUCOULOMBIER, A. (1998), Extending population-based incremental learning to continuous search spaces, in: *Parallel Problem Solving from Nature — PPSN V*, vol. 1498, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 418–427, DOI: [10.1007/BFb0056884](https://doi.org/10.1007/BFb0056884).
- SENJU, S. and TOYODA, Y. (1968), An Approach to Linear Programming with 0–1 Variables, *Management Science*, **15** (4): B–196–B–207, DOI: [10.1287/mnsc.15.4.B196](https://doi.org/10.1287/mnsc.15.4.B196).
- SHAH, R. and REED, P. (2011), Comparative analysis of multiobjective evolutionary algorithms for random and correlated instances of multiobjective d-dimensional knapsack problems, *European Journal of Operational Research*, **211** (3): 466–479, DOI: [10.1016/j.ejor.2011.01.030](https://doi.org/10.1016/j.ejor.2011.01.030).
- SHIH, W. (1979), A branch and bound method for the multiconstraint zero-one knapsack problem, *Journal of the Operational Research Society*, 369–378.
- SHIR, O. (2012), Niching in Evolutionary Algorithms, in: *Handbook of Natural Computing*, Springer Berlin Heidelberg, pp. 1035–1069, DOI: [10.1007/978-3-540-92910-9_32](https://doi.org/10.1007/978-3-540-92910-9_32).
- SIMON, H. (1996), The sciences of the artificial, the MIT Press.
- SOKOLOV, A. and WHITLEY, D. (2005), Unbiased tournament selection, in: *Genetic and Evolutionary Computation Conference, GECCO '05*, ACM, pp. 1131–1138, DOI: [10.1145/1068009.1068198](https://doi.org/10.1145/1068009.1068198).
- SRINIVAS, N. and DEB, K. (1994), Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation*, **2** (3): 221–248, DOI: [10.1162/evco.1994.2.3.221](https://doi.org/10.1162/evco.1994.2.3.221).
- SUDHOLT, D. (2005), Crossover is Provably Essential for the Ising Model on Trees, in: *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, GECCO '05*, ACM, pp. 1161–1167, DOI: [10.1145/1068009.1068202](https://doi.org/10.1145/1068009.1068202).
- (2012), Crossover Speeds Up Building-block Assembly, in: *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, GECCO '12*, ACM, pp. 689–702, DOI: [10.1145/2330163.2330260](https://doi.org/10.1145/2330163.2330260).
- TAN, Y.-y. et al. (2013), MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives, *Computers & Operations Research*, **40** (6): 1648–1660, DOI: [10.1016/j.cor.2012.01.001](https://doi.org/10.1016/j.cor.2012.01.001).
- TAVARES, J., PEREIRA, F., and COSTA, E. (2006), The Role of Representation on the Multidimensional Knapsack Problem by means of Fitness Landscape Analysis, in: *IEEE Congress on Evolutionary Computation, CEC '06*, pp. 2307–2314, DOI: [10.1109/CEC.2006.1688593](https://doi.org/10.1109/CEC.2006.1688593).

- TAVARES, J., PEREIRA, F., and COSTA, E. (2008), Multidimensional Knapsack Problem: A Fitness Landscape Analysis, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **38** (3): 604–616, DOI: [10.1109/TSMCB.2008.915539](https://doi.org/10.1109/TSMCB.2008.915539).
- THIERENS, D. (2010), The Linkage Tree Genetic Algorithm, in: *Parallel Problem Solving from Nature (PPSN XI)*, vol. 6238, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 264–273, DOI: [10.1007/978-3-642-15844-5_27](https://doi.org/10.1007/978-3-642-15844-5_27).
- THIERENS, D. and BOSMAN, P. (2011), Optimal mixing evolutionary algorithms, in: *Genetic and Evolutionary Computation Conference, GECCO '11*, ACM, pp. 617–624, DOI: [10.1145/2001576.2001661](https://doi.org/10.1145/2001576.2001661).
- (2012a), Evolvability Analysis of the Linkage Tree Genetic Algorithm, in: *Parallel Problem Solving from Nature - PPSN XII*, vol. 7491, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 286–295, DOI: [10.1007/978-3-642-32937-1_29](https://doi.org/10.1007/978-3-642-32937-1_29).
- (2012b), Learning the Neighborhood with the Linkage Tree Genetic Algorithm, in: *Learning and Intelligent Optimization*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 491–496, DOI: [10.1007/978-3-642-34413-8_50](https://doi.org/10.1007/978-3-642-34413-8_50).
- (2012c), Predetermined Versus Learned Linkage Models, in: *Genetic and Evolutionary Computation Conference, GECCO '12*, ACM, pp. 289–296, DOI: [10.1145/2330163.2330205](https://doi.org/10.1145/2330163.2330205).
- (2013), Hierarchical Problem Solving with the Linkage Tree Genetic Algorithm, in: *Genetic and Evolutionary Computation Conference, GECCO '13*, ACM, pp. 877–884, DOI: [10.1145/2463372.2463477](https://doi.org/10.1145/2463372.2463477).
- TOMINAGA, Y. et al. (2013), Binary-Based Topology Optimization of Magnetostatic Shielding by a Hybrid Evolutionary Algorithm Combining Genetic Algorithm and Extended Compact Genetic Algorithm, *IEEE Transactions on Magnetics*, **49** (5): 2093–2096, DOI: [10.1109/TMAG.2013.2240282](https://doi.org/10.1109/TMAG.2013.2240282).
- VASQUEZ, M. and HAO, J.-K. (2001), A hybrid approach for the 0–1 multidimensional knapsack problem, in: *In Proceedings of the International Joint Conference on Artificial Intelligence 2001*, pp. 328–333.
- VASQUEZ, M. and VIMONT, Y. (2005), Improved results on the 0–1 multidimensional knapsack problem, *European Journal of Operational Research*, **165** (1): 70–81, DOI: [10.1016/j.ejor.2004.01.024](https://doi.org/10.1016/j.ejor.2004.01.024).
- VAZ, D., PAQUETE, L., and PONTE, A. (2013), A note on the $\tilde{f}_t$ -indicator subset selection, *Theoretical Computer Science*, **499** (0): 113–116, DOI: [10.1016/j.tcs.2013.05.013](https://doi.org/10.1016/j.tcs.2013.05.013).
- VELDHUIZEN, D and LAMONT, G (2000), Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art, *Evolutionary Computation*, **8** (2): 125–147, DOI: [10.1162/106365600568158](https://doi.org/10.1162/106365600568158).
- VEREL, S. et al. (2011a), Analyzing the Effect of Objective Correlation on the Efficient Set of MNK-Landscapes, in: *Learning and Intelligent Optimization*, vol. 6683, Lecture Notes in

- Computer Science, Springer Berlin Heidelberg, pp. 116–130, DOI: [10.1007/978-3-642-25566-3_9](https://doi.org/10.1007/978-3-642-25566-3_9).
- (2011b), Pareto Local Optima of Multiobjective NK-Landscapes with Correlated Objectives, in: *Evolutionary Computation in Combinatorial Optimization*, vol. 6622, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 226–237, DOI: [10.1007/978-3-642-20364-0_20](https://doi.org/10.1007/978-3-642-20364-0_20).
- (2013), On the structure of multiobjective combinatorial search space: MNK-landscapes with correlated objectives, *European Journal of Operational Research*, **227** (2): 331–342, DOI: [10.1016/j.ejor.2012.12.019](https://doi.org/10.1016/j.ejor.2012.12.019).
- WANG, L. et al. (2012a), A novel modified binary differential evolution algorithm and its applications, *Neurocomputing*, **98** (0): 55–75, DOI: [10.1016/j.neucom.2011.11.033](https://doi.org/10.1016/j.neucom.2011.11.033).
- WANG, L., WANG, S.-y., and XU, Y. (2012b), An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem, *Expert Systems with Applications*, **39** (5): 5593–5599, DOI: [10.1016/j.eswa.2011.11.058](https://doi.org/10.1016/j.eswa.2011.11.058).
- WATSON, R. A. and JANSEN, T. (2007), A Building-block Royal Road Where Crossover is Provably Essential, in: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, ACM, pp. 1452–1459, DOI: [10.1145/1276958.1277224](https://doi.org/10.1145/1276958.1277224).
- WEINGARTNER, H. M. and NESS, D. N. (1967), Methods for the Solution of the Multidimensional 0/1 Knapsack Problem, *Operations Research*, **15** (1): 83–103, DOI: [10.1287/opre.15.1.83](https://doi.org/10.1287/opre.15.1.83).
- WRIGHT, S. (1932), The Roles of mutation, inbreeding, crossbreeding, and selection in evolution, in: *Proceedings of the 6th International Conference on Genetics*, vol. 1, pp. 356–366.
- XIE, H., ZHANG, M., and ANDREAE, P. (2007), Another investigation on tournament selection: modelling and visualisation, in: *Genetic and Evolutionary Computation Conference*, GECCO '07 1, ACM, pp. 1468–1475, DOI: [10.1145/1276958.1277226](https://doi.org/10.1145/1276958.1277226).
- YANG, S. (2005), Memory-enhanced univariate marginal distribution algorithms for dynamic optimization problems, in: *The 2005 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2560–2567 Vol. 3, DOI: [10.1109/CEC.2005.1555015](https://doi.org/10.1109/CEC.2005.1555015).
- YANG, S. and YAO, X. (2005), Experimental study on population-based incremental learning algorithms for dynamic optimization problems, *Soft Computing*, **9** (11): 815–834, DOI: [10.1007/s00500-004-0422-3](https://doi.org/10.1007/s00500-004-0422-3).
- (2008), Population-Based Incremental Learning With Associative Memory for Dynamic Environments, *IEEE Transactions on Evolutionary Computation*, **12** (5): 542–561, DOI: [10.1109/TEVC.2007.913070](https://doi.org/10.1109/TEVC.2007.913070).
- YIN, X. and GERMA, N. (1993), A Fast Genetic Algorithm with Sharing Scheme Using Cluster Analysis Methods in Multimodal Function Optimization, in: *Artificial Neural Nets and Genetic Algorithms*, Springer Vienna, pp. 450–457, DOI: [10.1007/978-3-7091-7533-0_65](https://doi.org/10.1007/978-3-7091-7533-0_65).

- YU, T.-L. et al. (2007), Population Sizing for Entropy-based Model Building in Discrete Estimation of Distribution Algorithms, in: *Genetic and Evolutionary Computation Conference, GECCO'2007*, GECCO '07, ACM, ACM, pp. 601–608, DOI: [10.1145/1276958.1277080](https://doi.org/10.1145/1276958.1277080).
- ZHANG, Q. and LI, H. (2007), MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, *IEEE Transactions on Evolutionary Computation*, **11** (6): 712–731, DOI: [10.1109/TEVC.2007.892759](https://doi.org/10.1109/TEVC.2007.892759).
- ZHANG, Q., LIU, W., and LI, H. (2009), The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances, in: *IEEE Congress on Evolutionary Computation, CEC '09*, IEEE Press, pp. 203–208.
- ZHANG, Q. et al. (2010), Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model, *IEEE Transactions on Evolutionary Computation*, **14** (3): 456–474, DOI: [10.1109/TEVC.2009.2033671](https://doi.org/10.1109/TEVC.2009.2033671).
- ZHOU, A. et al. (2011), Multiobjective evolutionary algorithms: A survey of the state of the art, *Swarm and Evolutionary Computation*, **1** (1): 32–49, DOI: [10.1016/j.swevo.2011.03.001](https://doi.org/10.1016/j.swevo.2011.03.001).
- ZITZLER, E. et al. (2003), Performance assessment of multiobjective optimizers: an analysis and review, *IEEE Transactions on Evolutionary Computation*, **7** (2): 117–132, DOI: [10.1109/TEVC.2003.810758](https://doi.org/10.1109/TEVC.2003.810758).
- ZITZLER, E., THIELE, L., and BADER, J. (2010), On Set-Based Multiobjective Optimization, *IEEE Transactions on Evolutionary Computation*, **14** (1): 58–79, DOI: [10.1109/TEVC.2009.2016569](https://doi.org/10.1109/TEVC.2009.2016569).
- ZITZLER, E. and KÜNZLI, S. (2004), Indicator-Based Selection in Multiobjective Search, in: *Parallel Problem Solving from Nature - PPSN VIII*, vol. 3242, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 832–842, DOI: [10.1007/978-3-540-30217-9_84](https://doi.org/10.1007/978-3-540-30217-9_84).
- ZITZLER, E. and THIELE, L. (1999), Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, **3** (4): 257–271, DOI: [10.1109/4235.797969](https://doi.org/10.1109/4235.797969).
- ZITZLER, E., LAUMANN, M., and THIELE, L. (2002), SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization, in: *Evolutionary Methods for Design, Optimisation, and Control*, CIMNE, Barcelona, Spain, pp. 95–100.
- ZYDALLIS, J. B. (2003), Explicit building-block multiobjective genetic algorithms: theory, analysis, and development, AAI3077559, PhD thesis, Air Force Institute of Technology, School of Engineering and Management, pp. 391.